Medical Imaging Research Laboratory
www.creatis.insa-lyon.fr

Université Claude Bernard Lyon 1

UNIVERSITÉ DE LYON

(rigid/affine) Image registration

N. Duchateau – adapted from D. Sarrut + S. Rit

Master MISS / FSR-MED - Lyon, FR - 06/01/2022

Course overview

Theory:

1) Rigid/affine + basics = 3h — N. Duchateau
2) Non-rigid + advanced concepts = 3h + 1.5h — D. Sarrut

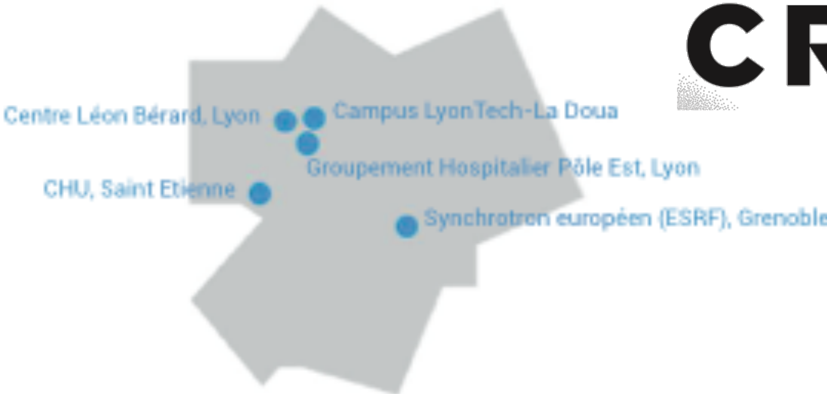
Labs:

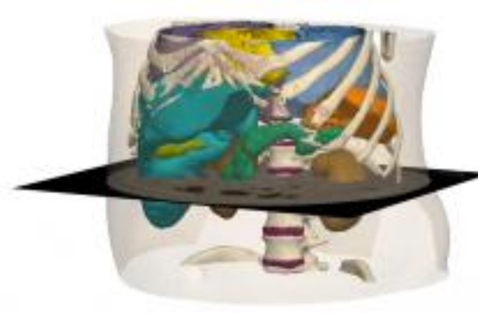
1) Elastix / rigid-affine-nonrigid = 3h — D. Sarrut
2) Motion estimation = 3h — P. Clarysse
3) Quality check / temporal tracking = 3h — N. Duchateau

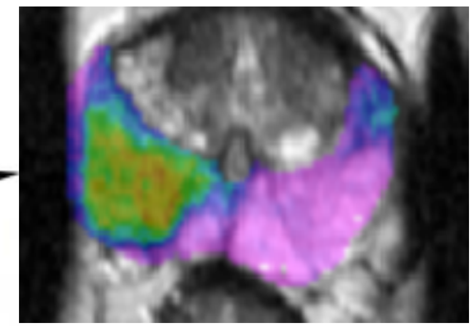
Evaluation: **Labs (x3) + Exam (1h30)**

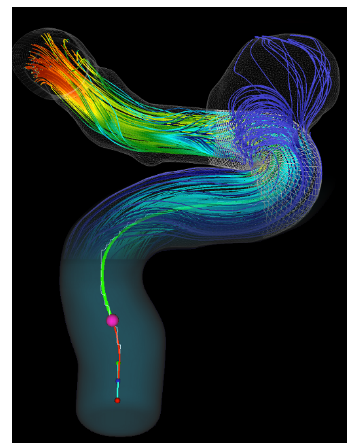
About (ND)

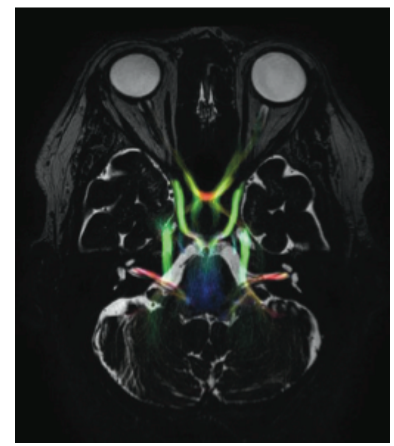
CREATIS











<u>4 équipes:</u>

- Analyse d'images et modélisation
- Imagerie ultrasonore
- Imagerie tomographique + radiothérapie
- Imagerie RMN + optique

+ plate-forme **d'imagerie multi-modale (PILOT)**

+ plate-forme de **calcul scientifique** + VIP (Virtual Imaging Platform)

Introduction

Introduction

Definition = "the process of aligning **[images/meshes]** so that corresponding ***features*** can easily be related"

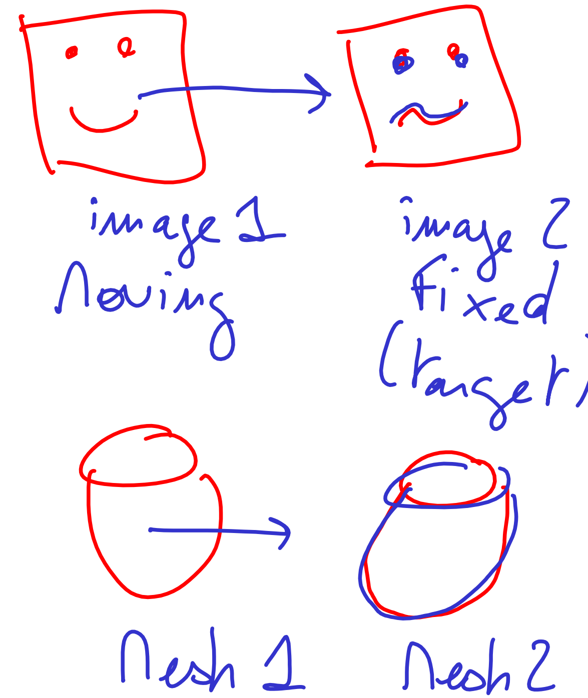
= registration = recalage

= alignment

= geometrical correspondence

= matching

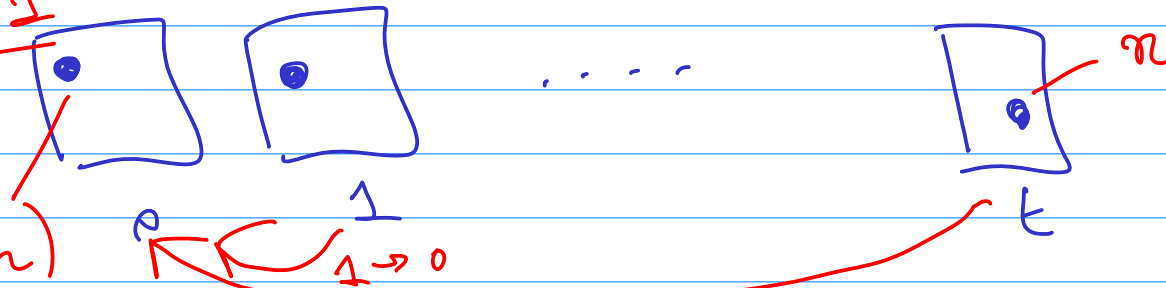
(= motion estimation ?)



Sequence = estimer mouvement

TP#3

Stratégie #1

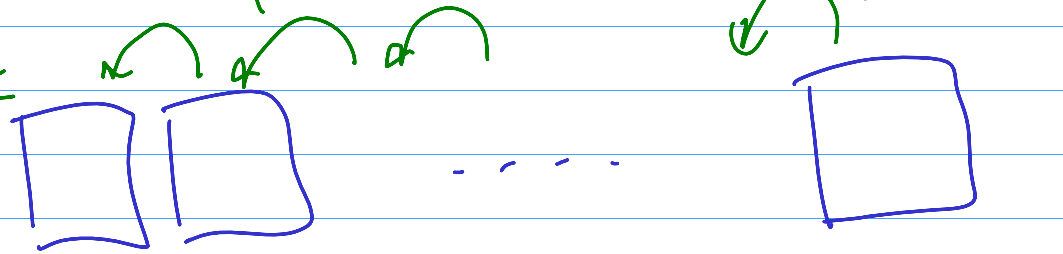


$\phi_{t \to 0}(x)$

estimation de la transformation $t \to 0$

Stratégie #2

$1 \to 0$ $2 \to 1$ $t \to t-1$



Composition $\phi_{1 \to 0} \circ \phi_{2 \to 1} \circ \dots$

Introduction

Definition = “the process of aligning **[images/meshes]** so that corresponding ***features*** can easily be related”

= registration

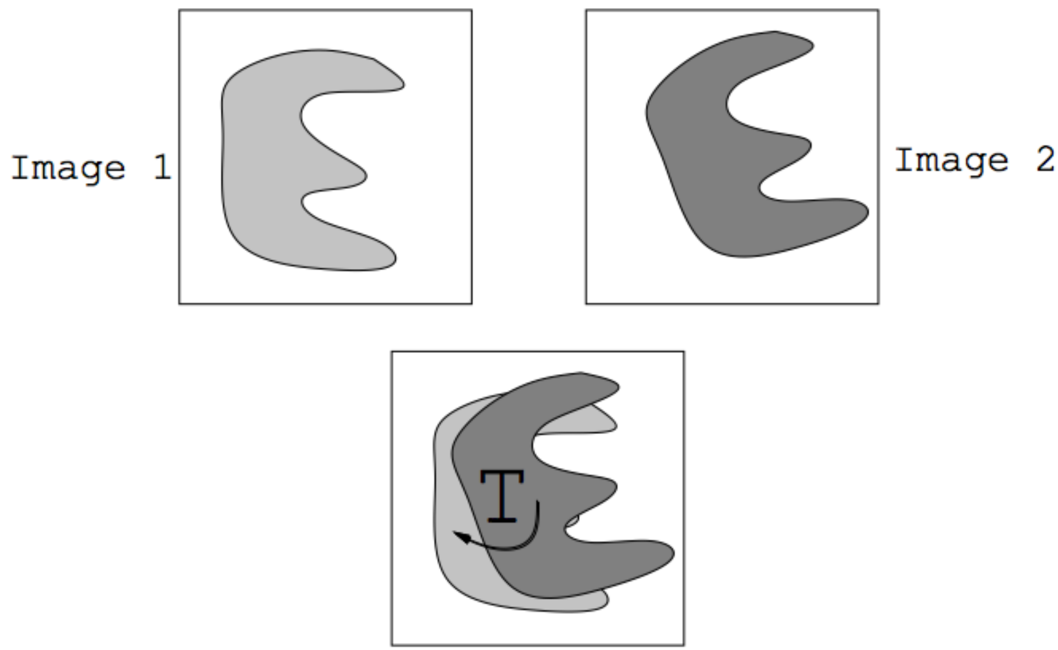
= alignment

= geometrical correspondence

= matching

(= motion estimation ?)

= find **spatial transformation *T*** that matches two [images/meshes]



Introduction

Definition = “the process of aligning **[images/meshes]** so that corresponding ***features*** can easily be related”

= registration

= alignment

= geometrical correspondence

= matching

(= motion estimation ?)

Important?

Many applications:

- Medical imaging
- Computer vision
- ...

**

≠ anatomies
entre patients

intra - patient

1 mois
6 mois

≠ modalités

Introduction

Definition = “the process of aligning **[images/meshes]** so that corresponding ***features*** can easily be related”

= registration

= alignment

= geometrical correspondence

= matching

(= motion estimation ?)

Important?

Many applications:

- Medical imaging →
- Computer vision →
- …

Constraints?

- Accurate?
- Fast?
- …

propriétés maths ? …

Introduction

Definition = "the process of aligning **[images/meshes]** so that corresponding ***features*** can easily be related"

= registration

= alignment

= geometrical correspondence

= matching

(= motion estimation ?)

Important?

Many applications:
- Medical imaging
- Computer vision
- …

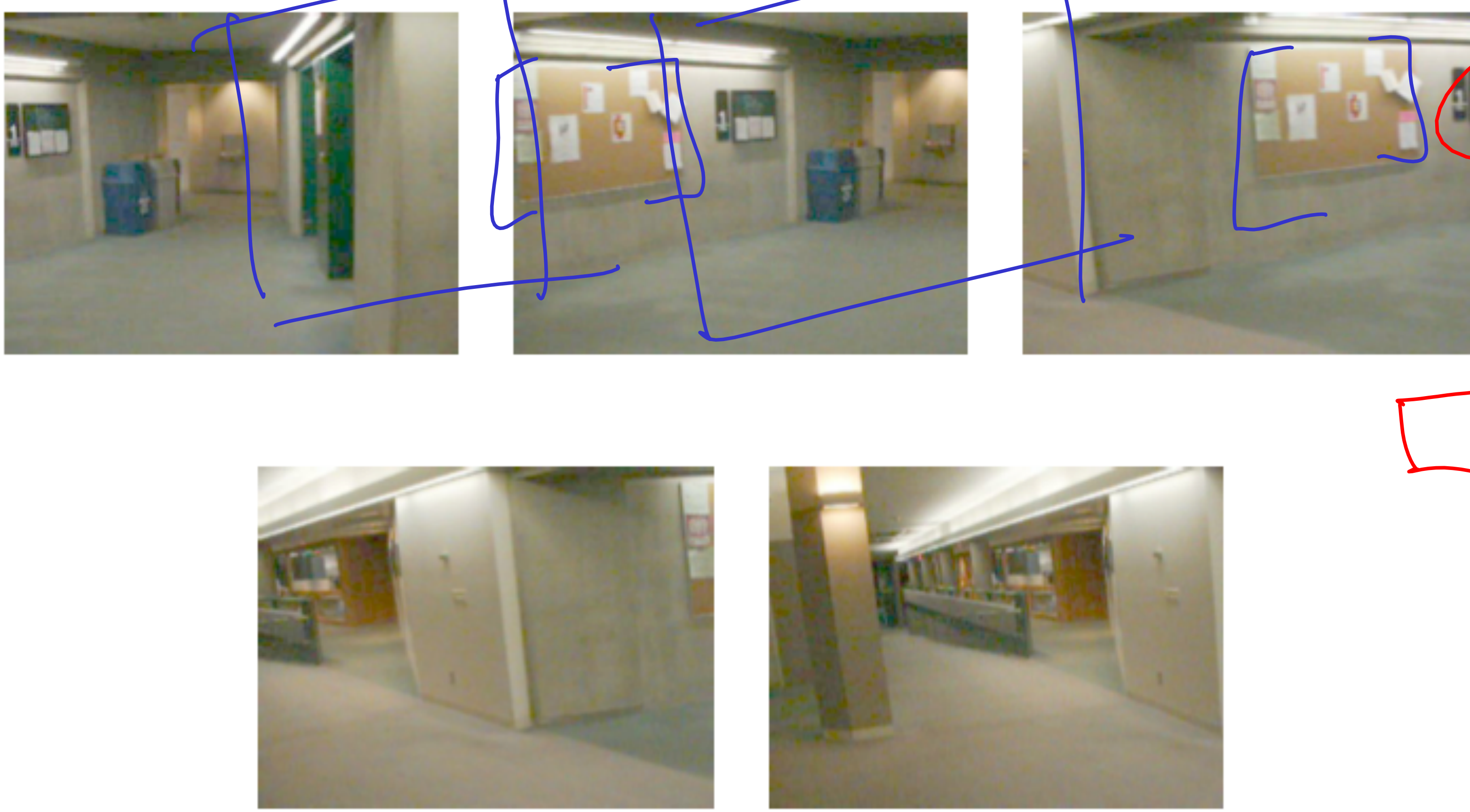
Constraints?
- Accurate?
- Fast?
- …

Basic elements used in many applications:
- Interpolation
- Similarity measure
- Optimization
- …

Examples

Panoramic photography

T simples :
rotation
translation
affine



Examples

Panoramic photography

Examples

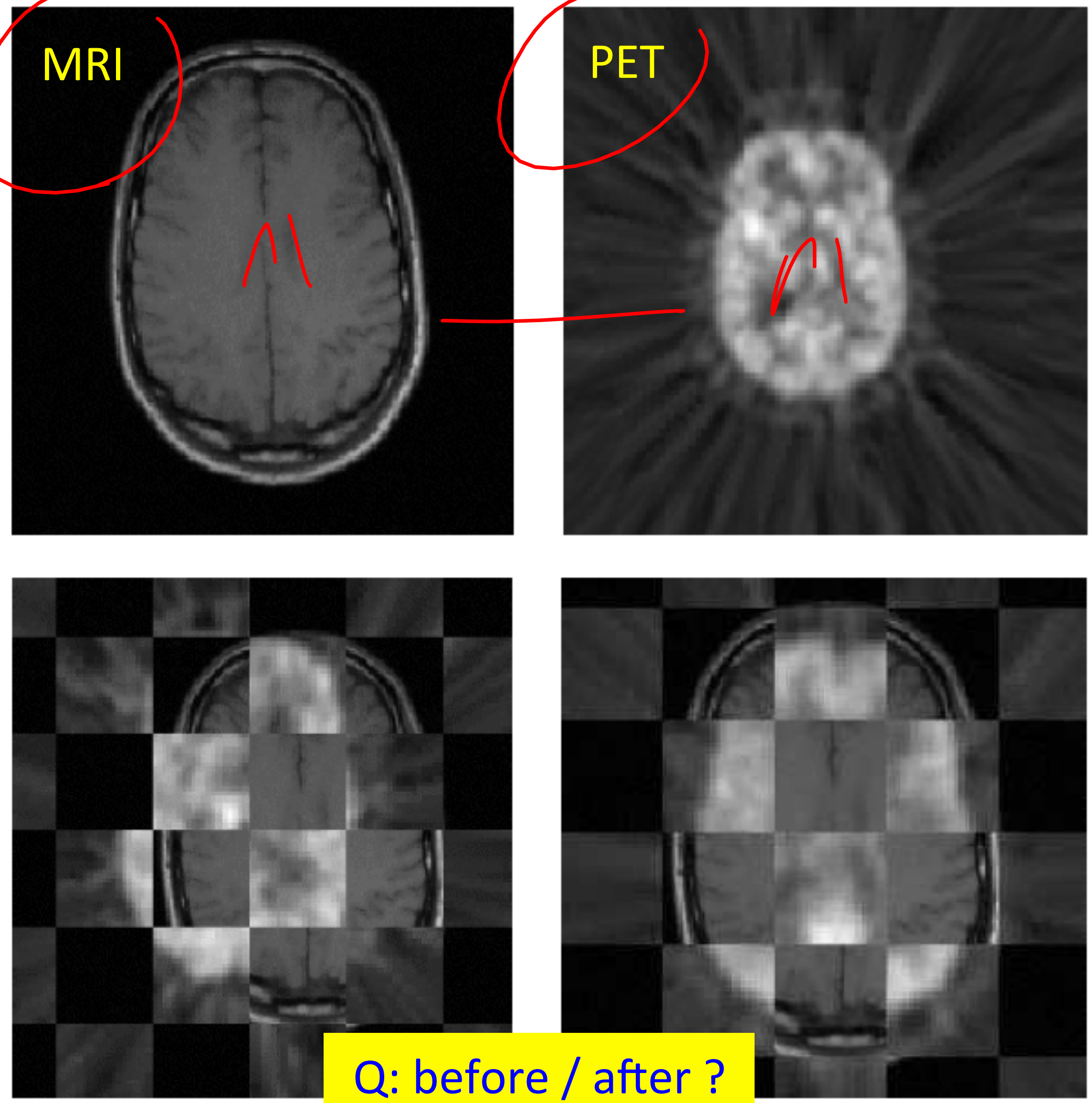
Satellite picture (e.g. GoogleMaps)

Zoom! translation

Examples

Multi-modal image fusion

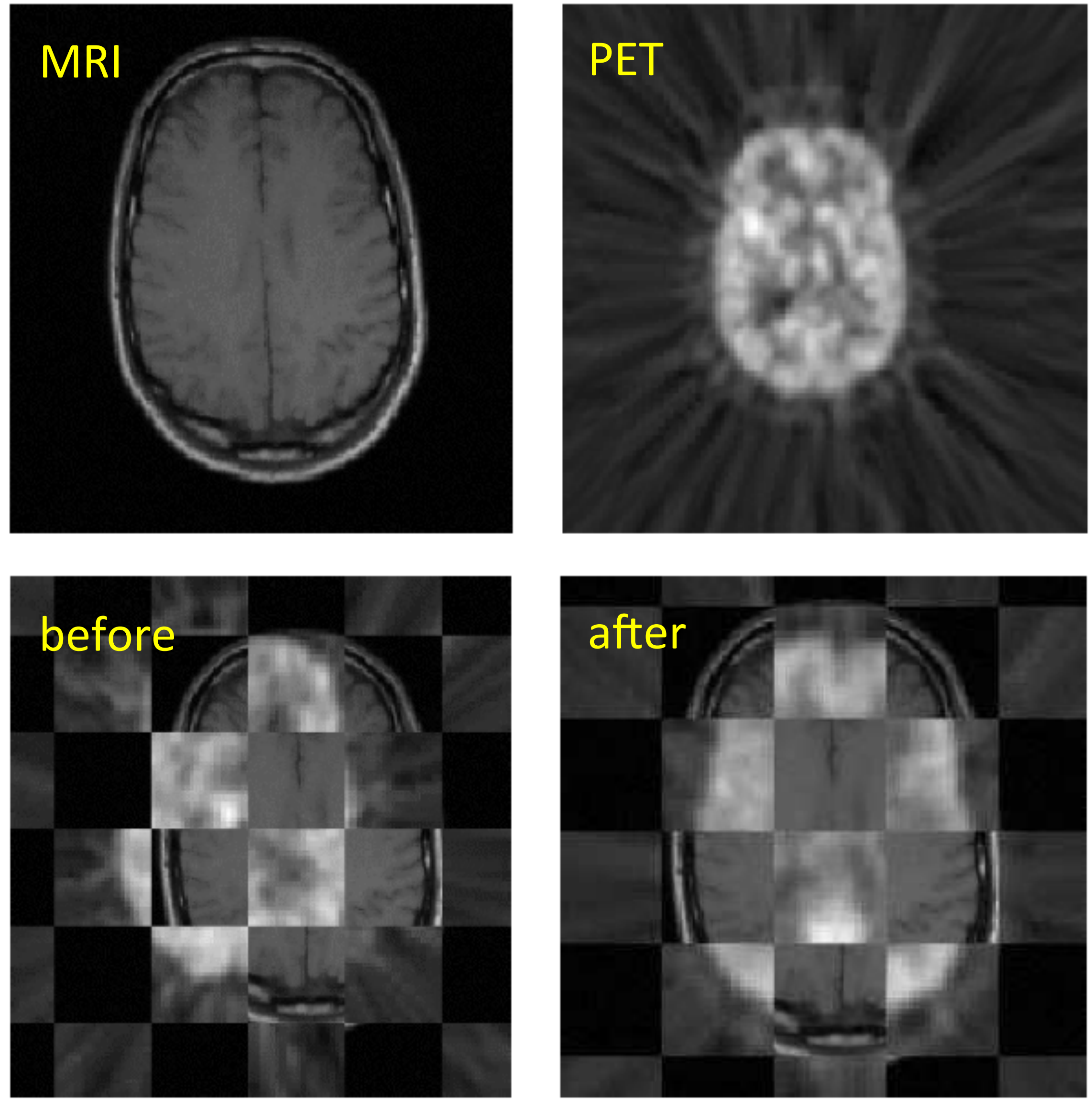
T linéaire ?
(si c'est le
même
patient)



Q: before / after ?

Examples

Multi-modal image fusion

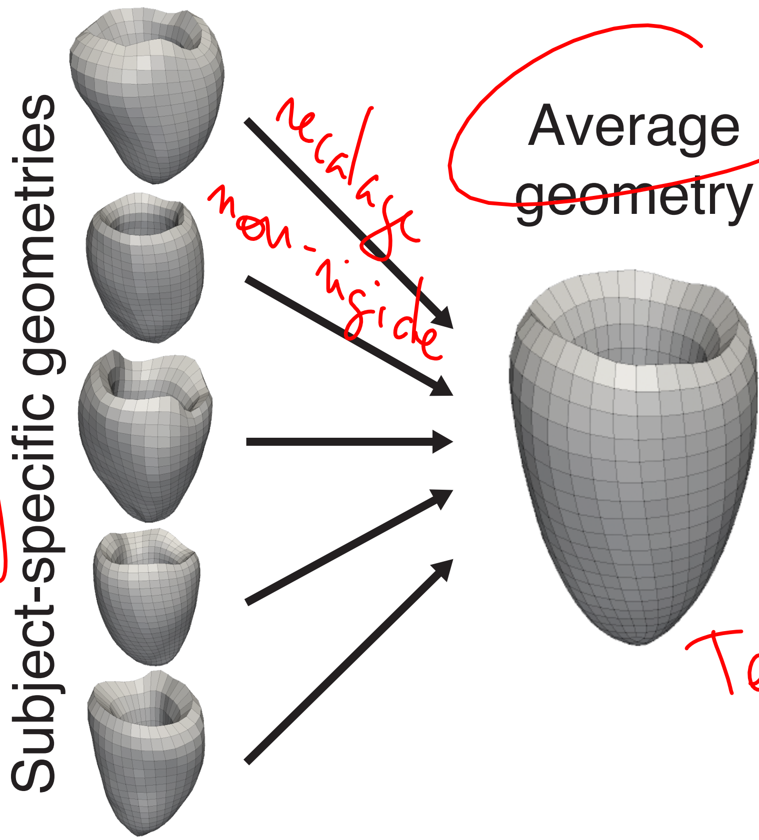


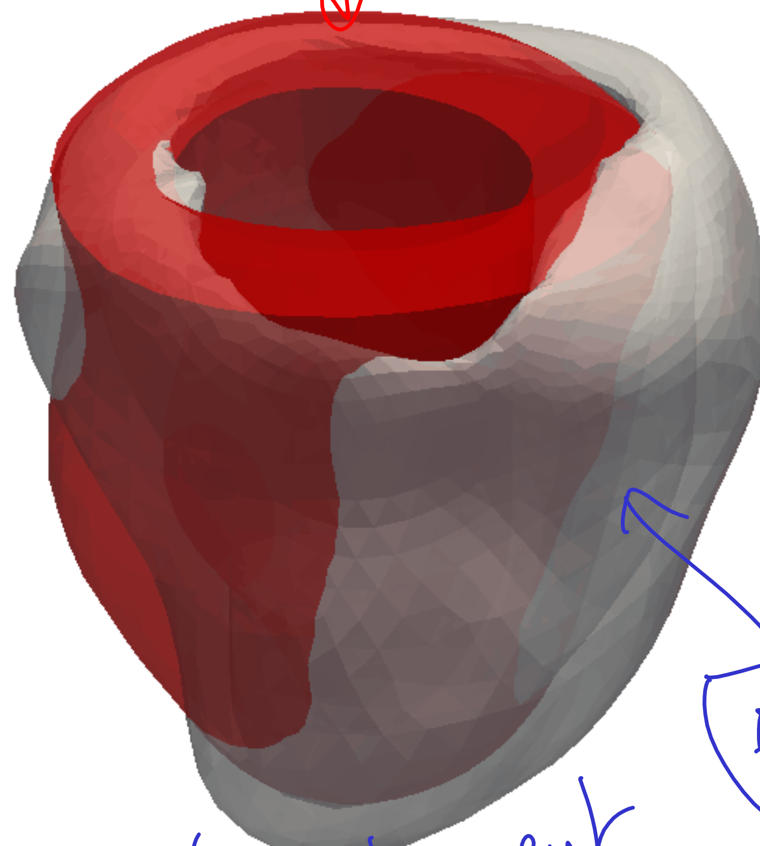
Crucial for accurate comparisons !!!

Examples

Statistics / machine learning ?

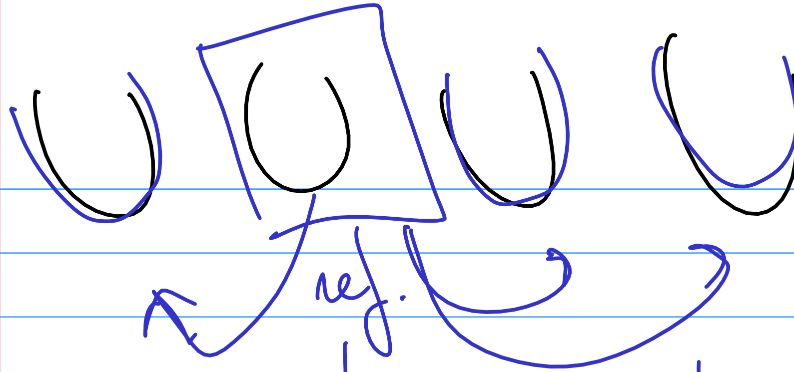
Atlas





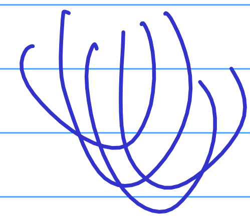
⚠ pas nécessairement le même nb de points !

2 stratégies

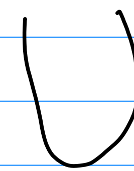


↳ on recale réf sur tous les autres,
donc ↑ nb de points + corresp.

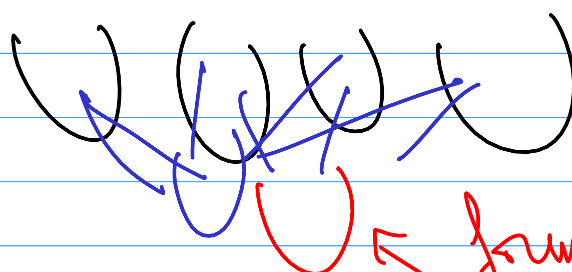
option #1



→ moyenne
std
...



option #2



← forme "centrale"

stats. sur transfos
+ processus itératif.

Examples

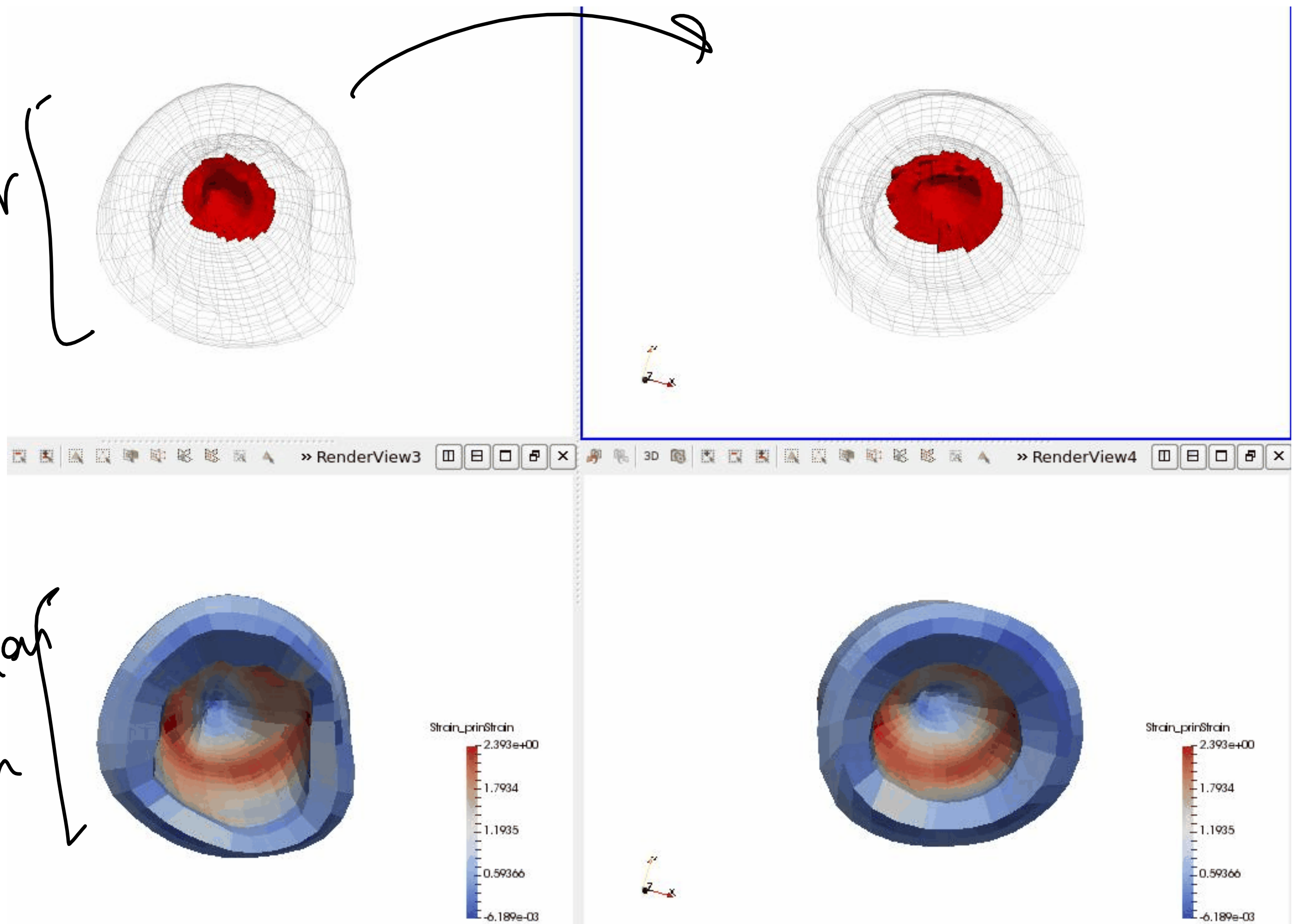
recalage + transport de l'information

Statistics / machine learning ?

before after

infarct

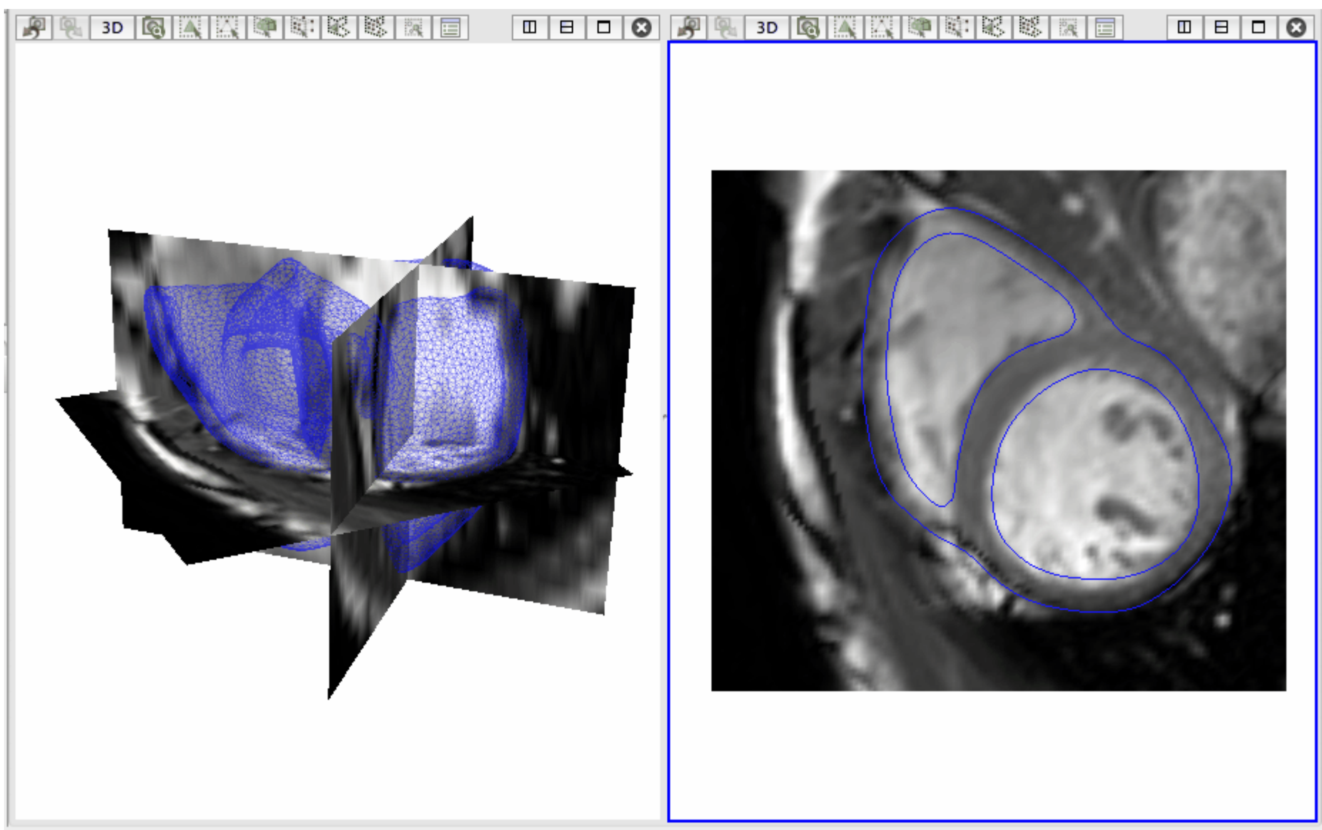
déformation du coeur



Examples

Motion analysis / tracking

cf. TP3…



Summary

- Find an "**optimal**" transformation
- **Inputs = images/meshes**… may differ in:
  - Resolution
  - Time (longitudinal changes)
  - Space (2D / 2D+t / 3D / 3D+t)
  - Modality (MRI, US, CT, PET, …)
  - Subject
  - …

+ **many possible output transformations**

= many algorithms !!!

Tools

- Code: ITK / VTK (C++), …

  images meshes

  Python

- Toolboxes: Elastix, Deformetrica, …

  TP#1 atlas

- Visualization: Paraview, …

  TP#3

Formalism

Q: $I = ?$
$J = ?$
$T = ?$
$S = ?$

images / maillages

meilleure

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

transf. candidates

similarité ?
(+ régularisation)

Formalism

Q: I = ?
J = ?
T = ?
S = ?

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Notation

- I = reference / fixed image
- J = moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- S = similarity measure
- arg max = optimization
- $\hat{T}$ = solution

famille des possibles

Digital images

résolution

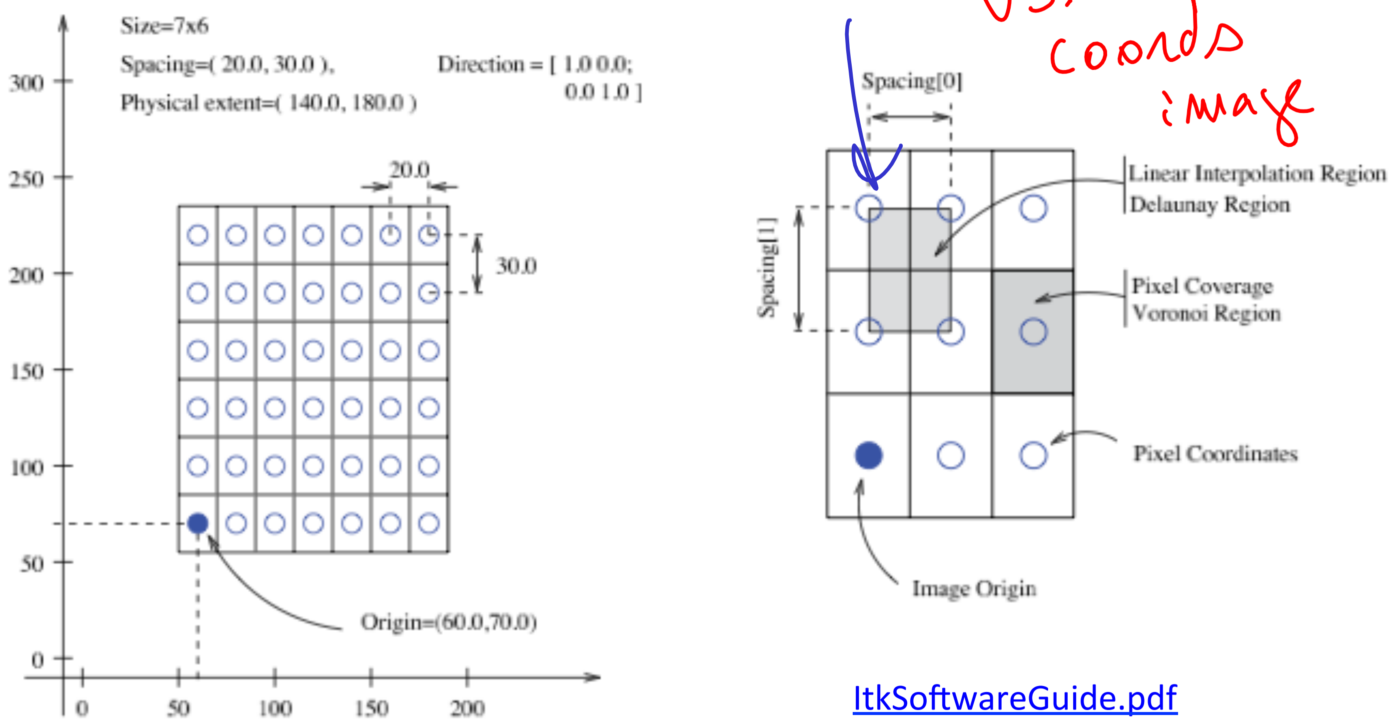
- **Size**: nb pixels + spacing
- **Pixel values**: scalar (CT) / vector (RGB) / matrix (diffusion MRI)
- **Colorscale** / dynamic range
- **Coordinates system**: origin + orientation

3x3 tenseur

coords physiques

vs.

coords image



ItkSoftwareGuide.pdf

Figure 4.1: Geometrical concepts associated with the ITK image.

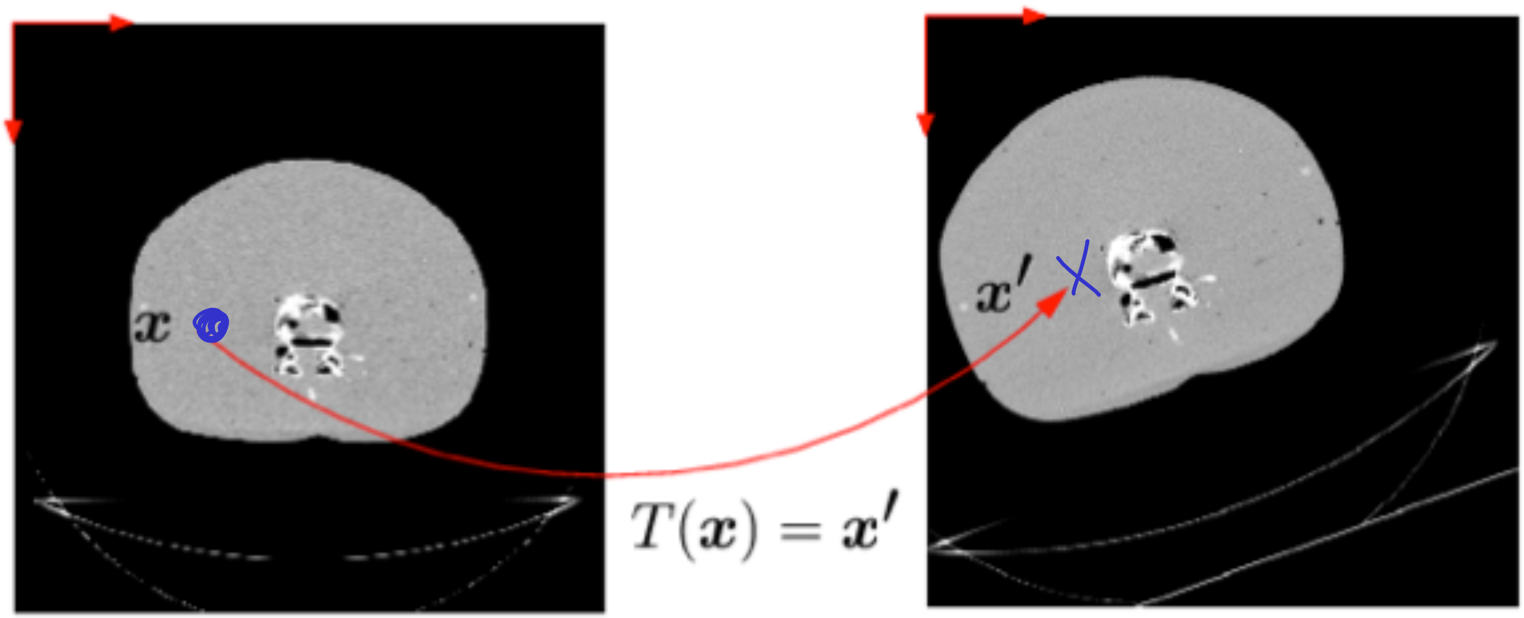
Transformations

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Notation

- I = reference / fixed image
- J = moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- S = similarity measure
- arg max = optimization
- $\hat{T}$ = solution

Transformations



- Cartesian coordinates: $\boldsymbol{x} = (x, y, z, \cancel{t})^T$

- Transformation: $$\begin{array}{lcl} T : \mathbb{R}^{\cancel{4}\,3} & \rightarrow & \mathbb{R}^{\cancel{4}\,3} \\ \boldsymbol{x} & \rightarrow & T(\boldsymbol{x}) = \boldsymbol{x}' \end{array}$$

Transformations: search space

One element of *T* is a transformation T.

$\mathcal{T}$

<u>Depends on:</u>

- **Degrees of freedom** = number *n* of parameters needed to describe T
  = dimension of the optimization problem

- **Boundaries** of each parameter (if any).

pb classique en optim° = bien définir l'espace des possibles

<u>Success conditioned by:</u>

- The solution belongs to the search space *T*
- The search space *T* is small enough vs. the input data (1 solution can be reached)

Transformations types

non-linear ≠ non-rigid

<u>Linear:</u>

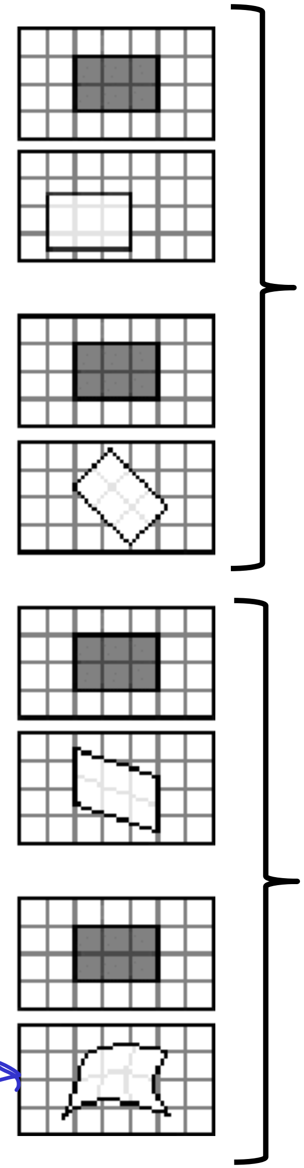
translation, rotation

+ scaling
+ shearing (affine)

<u>Non-linear:</u>

- Deformable
- Elastic
- Fluid

propriétés vs. anatomie...



Rigid

Non-rigid

Boucher et al. *ICPR* 2010

Transformations types

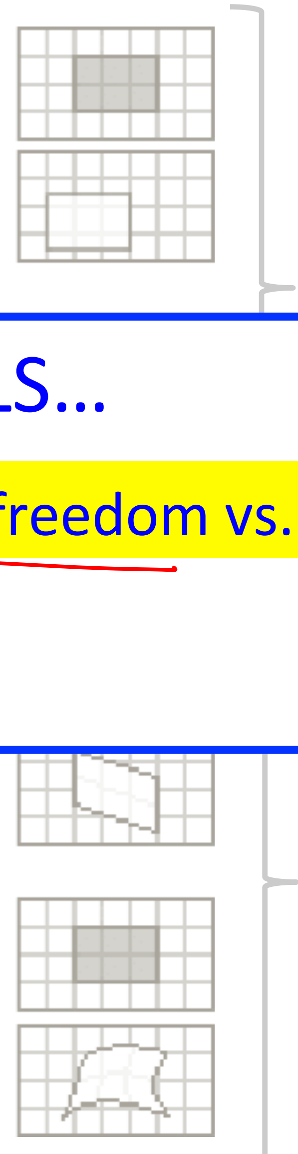
Linear:

translation, rotation

+ scali
+ shea

Non-

- De
- El
- Fluid



Rigid

Non-rigid

ALL are imposed MODELS…

Q: what if I want more freedom vs. the data?

Boucher et al. *ICPR* 2010

Transformations types

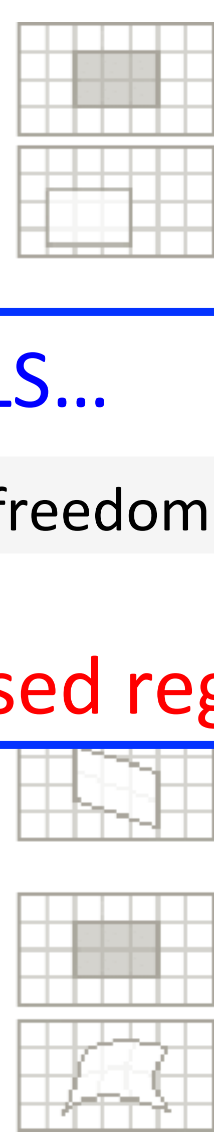
Linear:

translation, rotation

+ scali
+ shea

Non-

- De
- El
- Fluid



Rigid

Non-rigid

ALL are imposed MODELS…

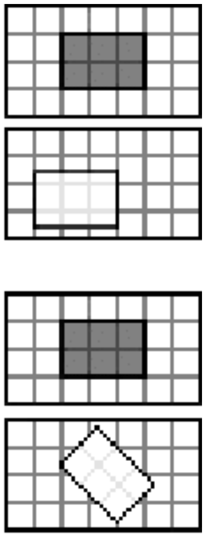
Q: what if I want more freedom vs. the data?

… **machine learning**- based registration?

cf. cours D. Sarrut?

Boucher et al. *ICPR* 2010

Rigid transform

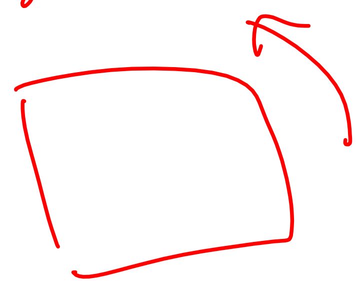


translation + rotation

Q: degrees of freedom = parameters = ?

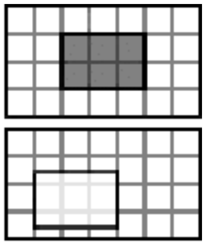
2D : transl : V, H = 2 degrés ?

rotation :

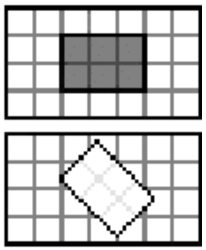


θ angle

Rigid transform



translation + rotation



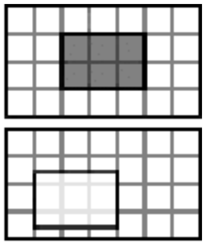
Q: degrees of freedom = parameters = ?

2D/2D: 3 parameters (2 translations + 1 rotation)

3D/3D: 6 parameters (3 rotations + 3 translations)

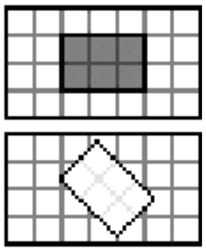
2D/3D: ~~6~~ 7 parameters (3 rotations + 3 translations + projection operator)

Matrix representations



translation + rotation

$$T_{rigid}(\mathbf{x}) = \underset{2\times 2}{\mathbf{R}}\,\mathbf{x} + \underset{2\times 1}{\mathbf{t}}$$



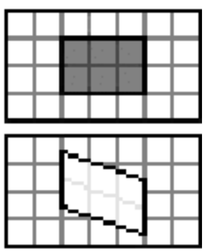
$$T_{affine}(\mathbf{x}) = \mathbf{A}\,\mathbf{x} + \mathbf{t} \qquad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

2×2 2×1

Q: dimension of x, R, M and t ?

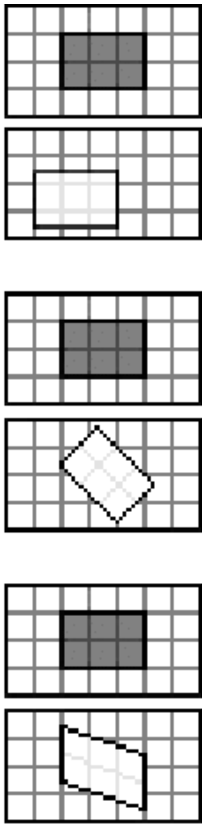
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad t = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

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



affine

Matrix representations



$$T_{rigid}(\mathbf{x}) = \mathbf{R}\,\mathbf{x} + \mathbf{t}$$

$$T_{affine}(\mathbf{x}) = \mathbf{A}\,\mathbf{x} + \mathbf{t}$$

analogie avec le terme de "biais" dans neural networks

Q: dimension of **x, R, M** and **t** ?

***

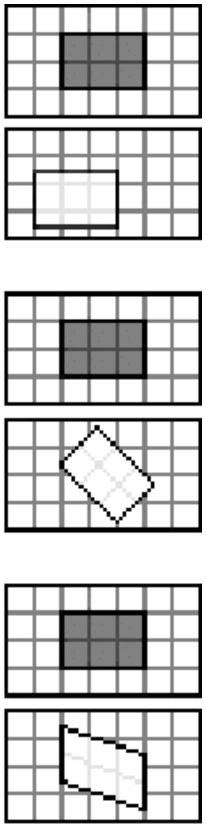
"Homogeneous" coordinates:

(convenient for projection transforms)

$$\mathbf{x} = (\,x\,,\,\underbrace{y\,,\,z\,,\,1}\,) \longrightarrow T_{affine}(\mathbf{x}) = \mathbf{M}\,\mathbf{x}$$

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrix representations



Q: linear transform = ?

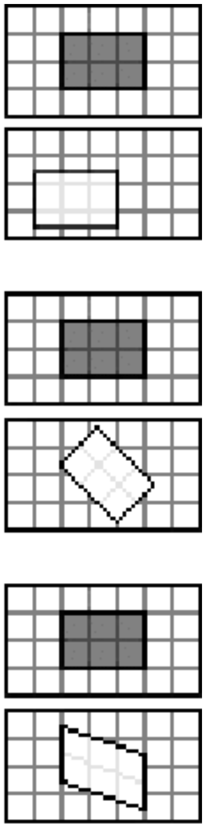
$$T_{affine}(\mathbf{x}) = \mathbf{M}\ \mathbf{x}$$

$$T(a\ \mathbf{x}_1 + \mathbf{x}_2) = a\ T(\mathbf{x}_1) + T(\mathbf{x}_2)$$

ex: scaling: $T(x) = \lambda x$

$$T(a x_1 + x_2) = \lambda (a x_1 + x_2)$$
$$= \lambda a x_1 + \lambda x_2$$
$$= a\, T(x_1) + T(x_2$$

Matrix representations



Q: linear transform = ?

$$T_{affine}(\mathbf{x}) = \mathbf{M}\ \mathbf{x}$$

$$T(a\ \mathbf{x}_1 + \mathbf{x}_2) = a\ T(\mathbf{x}_1) + T(\mathbf{x}_2)$$

intérêt de notation en coords homogènes

Q: operations = ?

- Inverse: $T^{-1}(\mathbf{x}) = \mathbf{M}^{-1}\ \mathbf{x}$ (if **M** is invertible)
- Composition: $T_2(\ T_1(\ \mathbf{x}\ )\ ) = (T_2 \circ T_1)(\mathbf{x}) = \mathbf{M}_2\ \mathbf{M}_1\ \mathbf{x}$
- …

Matrix representations: parameters → ex : 3D

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

= 12 parameters to optimize !

For registration:

- Option #1 = optimize the 12 parameters
- Option #2 = decompose in simpler transforms (rotation, translation, …)

bien plus intéressant si pb complexe

Pause : important = dimensionalité des objets :

Homogènes : $\vec{x} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \in \mathbb{R}^{4\times 1}$ $\quad M \in \mathbb{R}^{4\times 4}$ $\quad \vec{t} \in \mathbb{R}^{3\times 1}$

$$M = \begin{bmatrix} \boxed{R \in \mathbb{R}^{3\times 3}} & \vec{t} \\ 0 \; 0 \; 0 & 1 \end{bmatrix}$$

$$\underbrace{T(\vec{x})}_{\mathbb{R}^{4\times 1}} = \underbrace{M}_{\mathbb{R}^{4\times 4}} \underbrace{\vec{x}}_{\mathbb{R}^{4\times 1}}$$

Concept de déterminant : info sur déformation

$$\det(\underbrace{R}_{\in \mathbb{R}^{3\times 3}}) \in \mathbb{R}$$

Matrix representations: translations

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

~~= 12 parameters to optimize !~~

2D: 2 parameters

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & t_0 \\ 0 & 1 & t_1 \\ 0 & 0 & 1 \end{bmatrix}$$

3D: 3 parameters

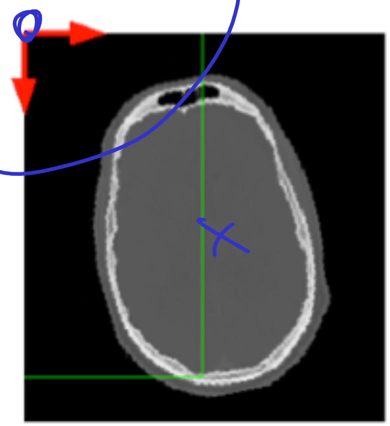
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrix representations: rotation

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

~~= 12 parameters to optimize !~~

⚠ origin



2D, around the origin: 1 parameter

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix representations: rotation

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

~~= 12 parameters to optimize !~~

2D, around the origin: 1 parameter

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Q: around another point?

$\mathbf{p} = (p_0 , p_1)$

translation + rotation

$$\begin{bmatrix} 1 & 0 & p_0 \\ 0 & 1 & p_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -p_0 \\ 0 & 1 & -p_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix representations: 3D rotation

$\mathbf{M\,x} \longleftrightarrow \mathbf{R\,x + t}$

- Rotation = 3 x 3 matrix
- 3 degrees of freedom = 3 parameters → 3 angles
- Constraints

  Orthogonal matrix: $\mathbf{R\,R^T = I}$ and $\det(\mathbf{R}) = 1$

  ***

  (inverse: $\mathbf{R}^{-1} = \mathbf{R}^T$)

Matrix representations: 3D rotation

- Rotation = 3 x 3 matrix
- 3 degrees of freedom = 3 parameters
- Constraints

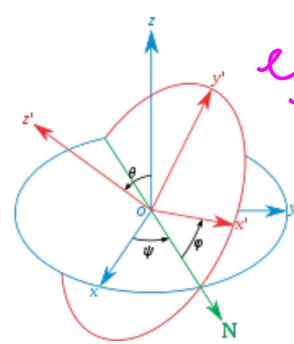
?

comment bien définir ça ?

- Several decompositions (from 9 values to 3 parameters):
  - Euler angles
  - Axis angle
  - Quaternions
  - …

The parameters have a different meaning in each case

> ➢ Check illustrations
> ➢ Q: pros vs. cons?



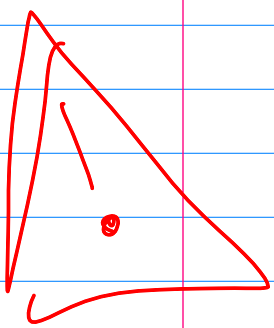
ex: Euler angles

Interlude : det

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

$$\underbrace{\det(R)}_{**} = \cos^2\theta - (-\sin^2\theta)$$
$$= 1$$

det : mesure de " déformation "



$\cancel{\det(M)}$ $\quad$ $\det(J_M)$ ← Jacobien

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

2D

$$\begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix} \quad \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}$$

$$Rx = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \overbrace{\cos\theta\, x_1 - \sin\theta\, x_2}^{f_1} \\ \sin\theta\, x_1 + \cos\theta\, x_2 \end{bmatrix}$$

$$J_R = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} = \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = R \,!$$

Dans le cadre des notations : $J_R = R$

$$\underbrace{\det(J_R) = \det(R) = 1}_{\text{on garde ça en tête ...}}$$

Matrix representations: scaling

$$M = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R

M x ⟷ **R x + t**

~~= 12 parameters to optimize !~~

3D: 3 parameters

$$R = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

s_1, s_2, s_3

⚠ R ≠ rotation

$$Rx = \begin{bmatrix} s_1 x_1 \\ s_2 x_2 \\ s_3 x_3 \end{bmatrix}$$

20

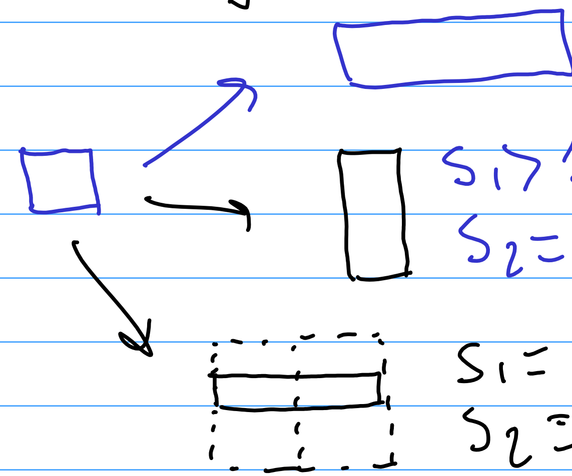
$\det(J_{scaling}?)$

$$f(x) = \begin{pmatrix} s_1 x_1 \\ s_2 x_2 \end{pmatrix} \begin{matrix} f_1 \\ f_2 \end{matrix}$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$$

$\det(J) = s_1 s_2$

ex :

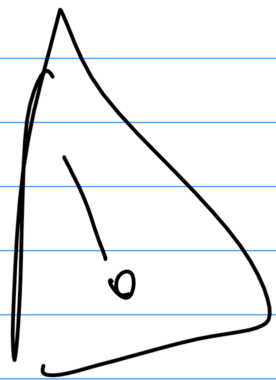


$s_1 = 1$, $s_2 > 1$: $\det(J) > 1$

$s_1 > 1$, $s_2 = 1$: $\det(J) > 1$

$s_1 = 0{,}5$, $s_2 = 2$: $\det(J) = 1$

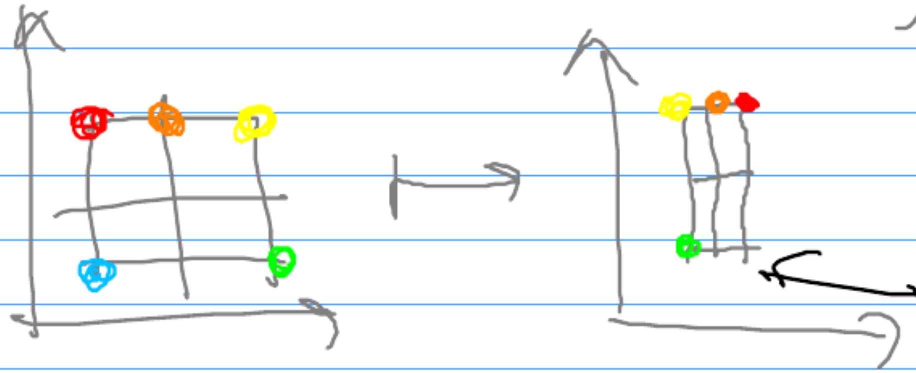
intuition que $\det(J)$ = mesure de "déformation"



calculs, faits en un point précis !

Conclusion : $\det(R)$
- > 1 grossit
- $= 1$ pareil (on peut translater | rotation)
- < 1 réduit
- < 0) inverse orientation

ex : $\det(R) < 0$



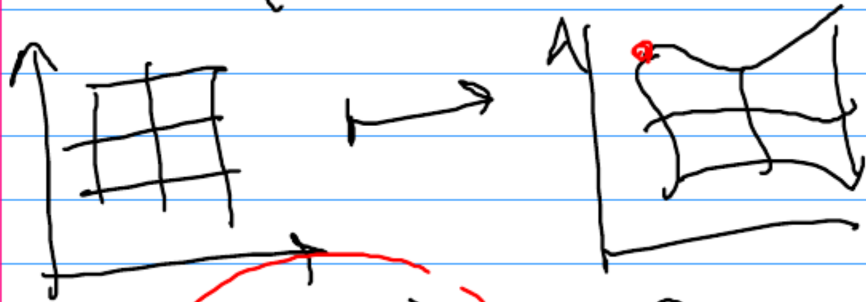
$\det(R) < 0$

$\det(R)$ caractérise déformation

Transformations linéaires

$$T(\vec{x}) = M\vec{x}$$

Rq : transformations non-linéaires



$T(\vec{x}) = ?$

pas de formulation matricielle

$\det(\cancel{R})$ $\qquad$ $\det(?)$

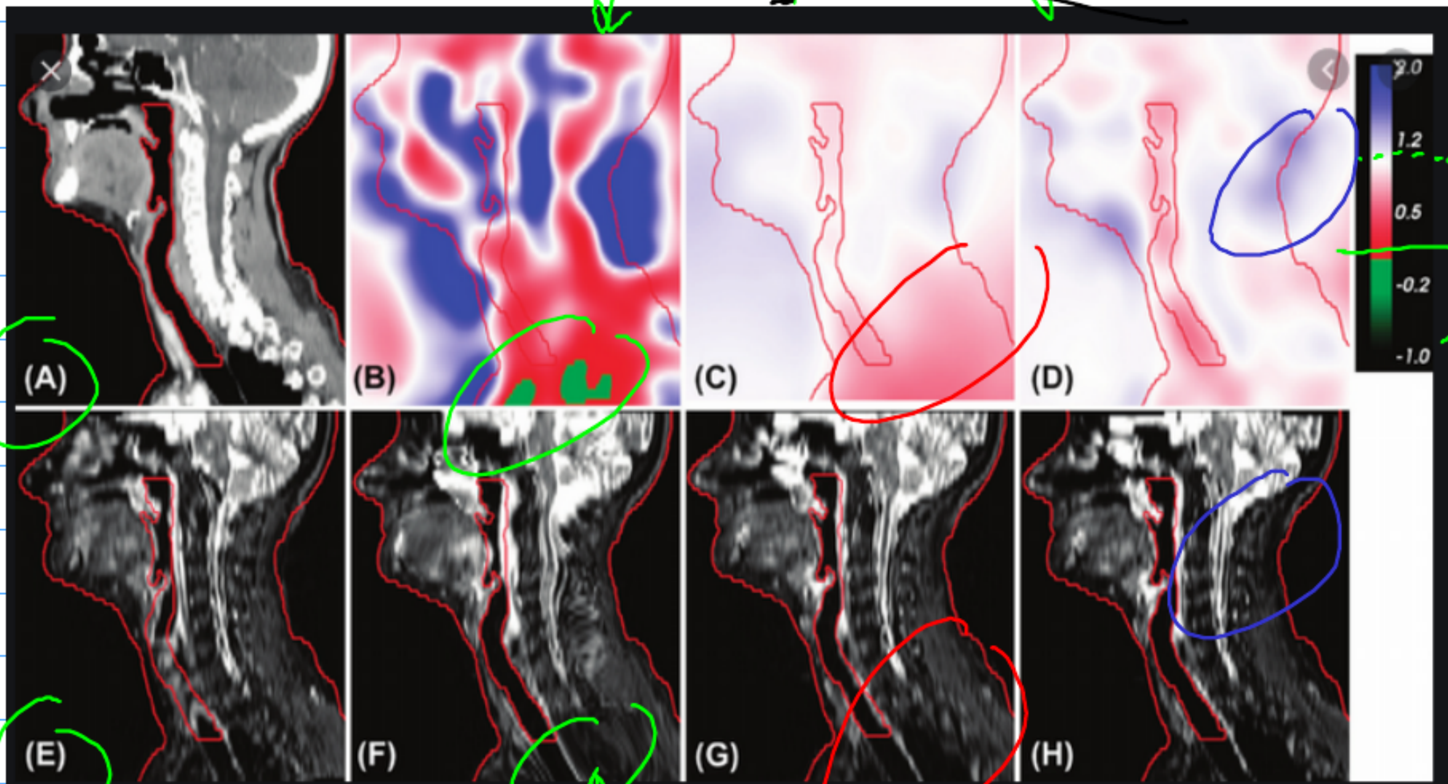
$\hookrightarrow \det\Big(\overbrace{J}^{\text{Jacobien}}\underbrace{(\vec{x})}_{\text{en chaque point}}\Big)$ $\quad \begin{cases} > 1 \\ = 1 \\ < 1 \end{cases}$

ex : 2D

$$J = \begin{bmatrix} \dfrac{\partial T_x}{\partial x} & \dfrac{\partial T_x}{\partial y} \\[2ex] \dfrac{\partial T_y}{\partial x} & \dfrac{\partial T_y}{\partial y} \end{bmatrix}$$

exemple

$\det(J(T(\vec{x})))$



$\det(J) > 1$ grossit localement

$= 1$

< 1

< 0

inversé localement orientation = pb!

= transf non inversible localement

pas souhaité en img médicale

→ on cherchera des transf° difféomorphiques

(inversible + lisse + ...

($\det(J) > 0$

Matrix representations: shear

$$M = \begin{bmatrix} \boxed{\begin{matrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{matrix}} & \begin{matrix} t_0 \\ t_1 \\ t_2 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

R

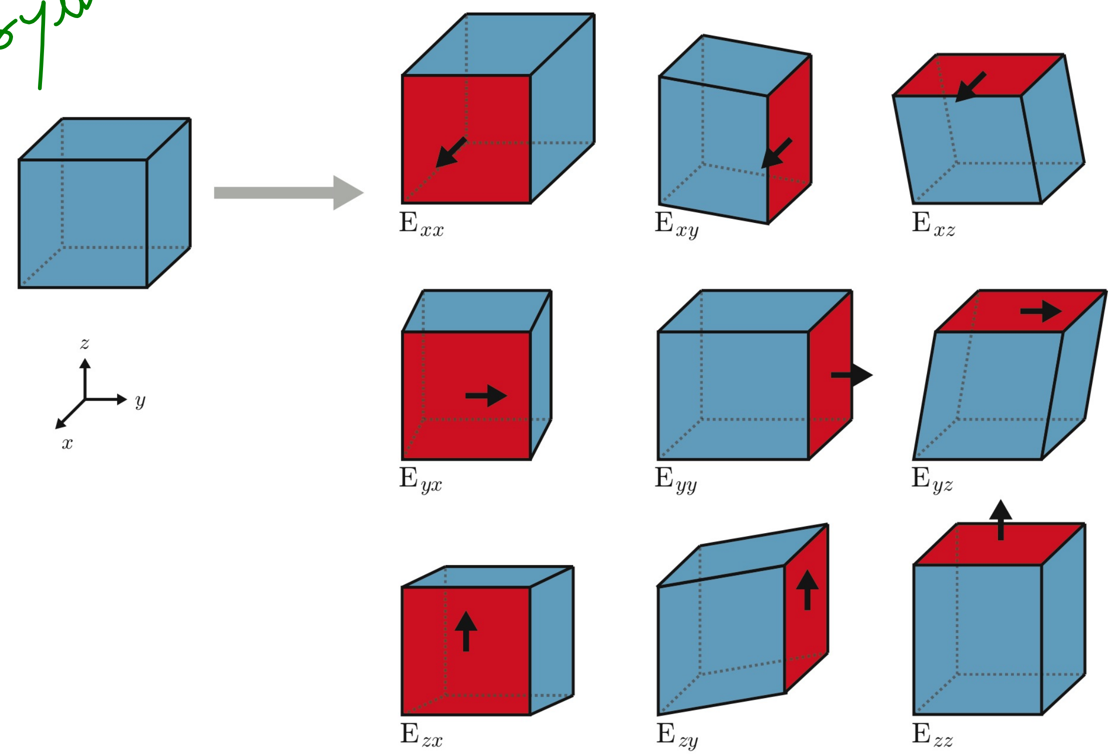
M x ⟷ **R x + t**

~~= 12 parameters to optimize !~~

ou 3 ? (symétrique)

3D: 6 parameters

$$R = \begin{bmatrix} 1 & s_{yx} & s_{zx} \\ s_{xy} & 1 & s_{zy} \\ s_{xz} & s_{yz} & 1 \end{bmatrix}$$



Matrix representations: skew

$$\mathbf{M} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & t_0 \\ R_{10} & R_{11} & R_{12} & t_1 \\ R_{20} & R_{21} & R_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

M x ⟷ **R x + t**

~~= 12 parameters to optimize !~~

3D: 3 parameters

$$\mathbf{R} = \begin{bmatrix} 1 & s_{yx} & s_{zx} \\ -s_{yx} & 1 & s_{zy} \\ -s_{zx} & -s_{zy} & 1 \end{bmatrix}$$

Close to shear, but antisymmetric: $\mathbf{R}^T = -\mathbf{R}$

Matrix representations: projection

$2D \rightarrow 3D$

2D 3D

- $\underbrace{(n-1)}_{2D} \times \underbrace{n}_{3D}$ matrix
- Parallel or perspective
- ex: 2D/1D perspective proj. in homogeneous coords:

SDD = Source to Detector Distance
SID = Source to Isocenter Distance

$$\mathbf{M} = \begin{bmatrix} SDD & 0 & 0 \\ 0 & 1 & SID \end{bmatrix}$$

Summary: transformations

Linear matrices can be decomposed in:
~~http://mathworld.wolfram.org/~~
~~http://www.wikipedia.com/~~

Translation parameters	straightforward
Rotation parameters	need caution, several solutions
Scaling parameters	straightforward
Skew parameters	less used

\+ **Projection** for 2D/3D registration

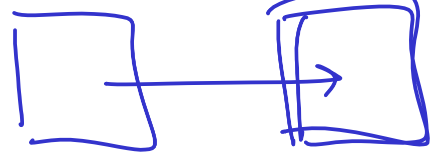
Summary: transformations

Linear matrices can be decomposed in:
~~http://mathworld.wolfram.org/~~
~~http://www.wikipedia.com/~~

Translation parameters	straightforward
Rotation parameters	need caution, several solutions
Scaling parameters	straightforward
Skew parameters	less used

+ **Projection** for 2D/3D registration

1 Rotation + transl.



2 Affine ...

3 Non-rigide

Only a subset of the parameters can (should?) be optimized:

Rigid	**6** parameters (translations + rotations)
Rigid + global scaling	**7** parameters
Rigid + independent scaling	**9** parameters
Affine	**12** parameters

Summary: transformations

Tips and tricks:

Carefully choose the search space

bien définir espace des possibles.

- As little parameters as possible…
- … but make sure the solution is in it !
- Bound the search space, e.g. translations < 2cm and rotations < 10º
- Scale the heterogeneous parameters, e.g. translations in mm and rotations in radians and degrees

Only a subset of the parameters can (should?) be optimized:

Rigid	**6** parameters (translations + rotations)
Rigid + global scaling	**7** parameters
Rigid + independent scaling	**9** parameters
Affine	**12** parameters

Similarity measure

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Notation

- I = reference / fixed image
- J = moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- $\boxed{S \text{ = similarity measure}}$
- arg max = optimization
- $\hat{T}$ = solution

Similarity measure

Measures the **alignment quality** of I and J:

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Q: Any idea?

$S(I, J)$

$S(I, T(J))$

$$S(I,J) = \frac{1}{N}\sqrt{\sum_{i=1}^{N} (I_i - J_i)^2}$$

RMSE

$$MAD = \frac{1}{N} \sum_{i=1}^{N} |I_i - J_i|$$

ici $S(I,J) = 0$
si similaires

$$\hat{T} = \arg\min S(I, T(J))$$

corrélation : $S(I, J) \simeq \begin{cases} 1 & \text{si similaires} \\ 0 & \text{sinon} \end{cases}$

+ plein d'autres $\longrightarrow$ à choisir <u>vs</u>. notre pb

Similarity measure

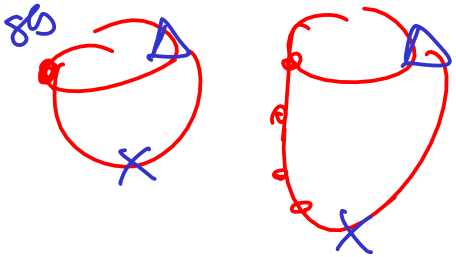
distance

Measures the **alignment quality** of I and J: ou argmin $D(I, T(J))$

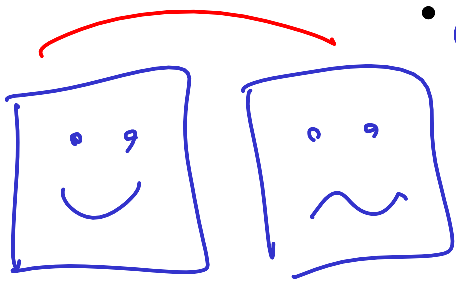
$$\hat{T} = \underset{T \in \mathcal{T}}{\arg\max}\; S(I, J, T)$$

Q: Any idea?

Two families:
- *Feature*-based registration → extraire des choses
- *Image*-based registration ← en tout point



img



Similarity measure: feature-based

Features are extracted **prior to the similarity** measure:

- Points/landmarks
- Lines
- Vectors
- Surfaces
- Volumes
- …

**

$$\hat{T} = \underset{T \in \mathcal{T}}{\arg\max}\ S(\mathcal{F}_I,\ T \circ \mathcal{F}_J)$$

Notation

- $\mathcal{F}_I$ = feature set of reference / fixed image
- $\mathcal{F}_J$ = feature set of moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- S = similarity measure
- arg max = optimization
- $\hat{T}$ = solution

Similarity measure: feature-based

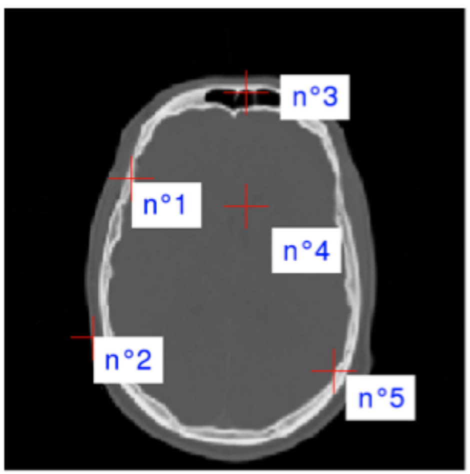
Features are extracted prior to the similarity measure:

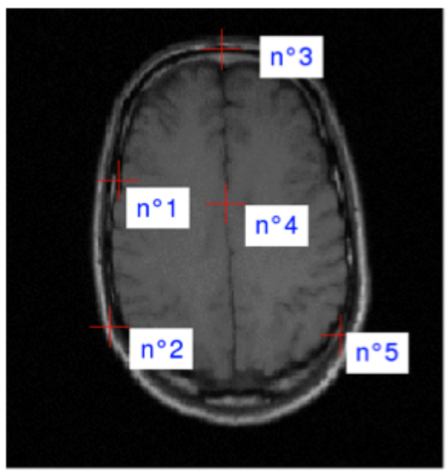
- Points/landmarks
- Lines
- Vectors
- Surfaces
- Volumes
- …

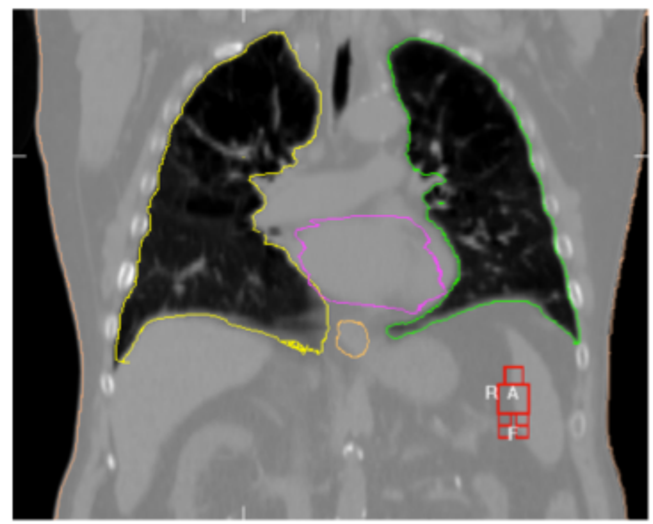
(machine learning?)

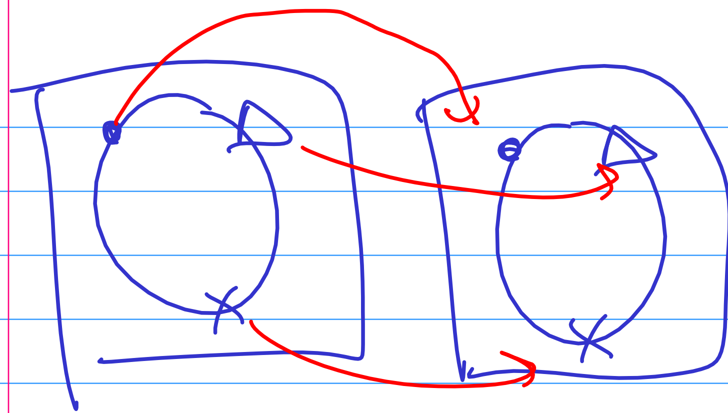
- **Manually** or **automatically** extracted
- Can be **unpaired** = not same sampling, amount, …

landmarks pré-identifiés









Q : que se passe-t-il entre les poses clés ?

↓

interpoler entre

⚠ Régularisation ***

↓

TP#3

Similarity measure: feature-based

(Dis-)similarity measure for features:

- Sum of distances:
  - L1 norm $$|\boldsymbol{x}|_1 = \sum_{r=1}^{n} |x_r|$$ → MAD
  - L2 norm $$|\boldsymbol{x}|_2 = \sqrt{\sum_{r=1}^{n} x_r^2}$$ ← SSD RMSE
- Quadratic sum (faster): $$\sum_{r=1}^{n} x_r^2$$ ...
- ...

Similarity measure: feature-based

(Dis-)similarity measure for features:

+ flexible

+ **distance map** = map of the distance between each point and the closest feature
= fast + useful for unpaired features

segmenté

pas forcément besoin de l'info dedans

Image Features = bones Distance map

- **Computed only once** for the reference image
- **Fast** computations: Chamfer distance (not Euclidean)
  separable algorithms [Coeurjolly et al. PAMI 2007]

Similarity measure: feature-based

$\boxed{I} \rightleftarrows \boxed{J}$

Symmetry?

~~xAx~~

$S(\mathcal{F}_I, T \circ \mathcal{F}_J)$ or $S(\mathcal{F}_J, T \circ \mathcal{F}_I)$?

ou combiner les deux ?

Different !

→ algos explicitement symétriques

$$S(I, J, T) = \sum_{F \in \mathcal{F}_J} d_I \circ T(F) \qquad S(I, J, T) = \sum_{F \in \mathcal{F}_I} d_J \circ T(F)$$

ex :

➢ Use the distance map of the image with the easiest-to-extract features

Similarity measure: feature-based

Summary:

- Preliminary steps to **identify / extract** features
- **Pairing** algorithm may be required
+ Generally **fast** (depends on the feature type and size)
+ Registration of **some features only**…

**

➢ Still interesting, especially for non-rigid registration…

++

combiner
landmarks + img + ...

Similarity measure: feature-based

Summary:

- Preliminary steps to **identify / extract** features
- **Pairing** algorithm may be required
+ Generally **fast** (depends on the feature type and size)
+ Registration of **some features only**…

➢ Still interesting, especially for non-rigid registration…

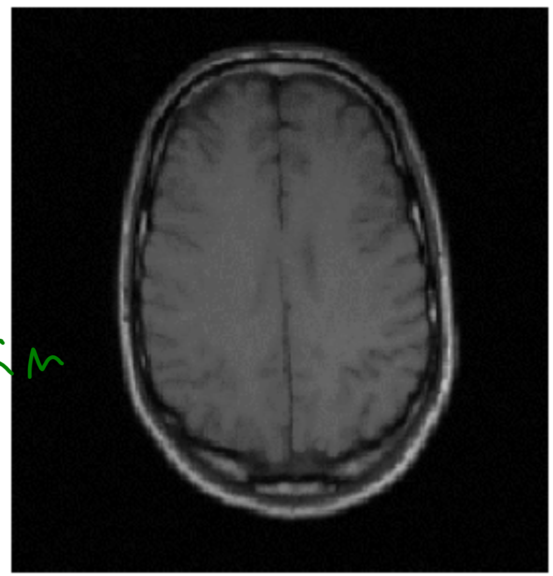
landmarks pertinents, mais difficiles à identifier

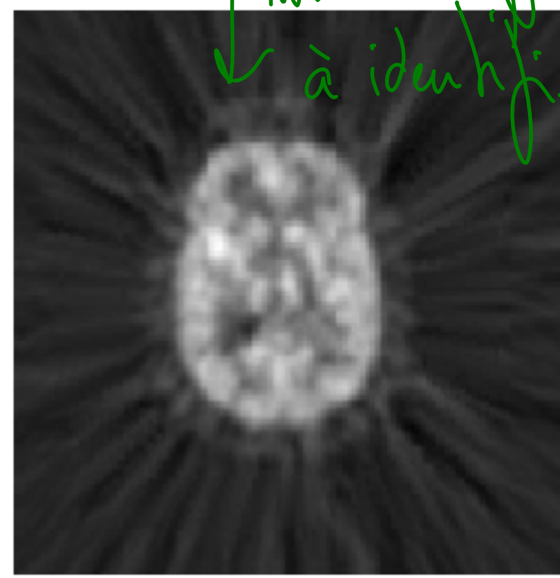
Another limitation:

Multi-modal images

ex : NMI
cf. + loin

➢ *Image-based* registration





Similarity measure: intensity-based

(assuming a functional dependence between the pixel intensities)

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, \boxed{J \circ T})$$

Notation

- I = reference / fixed image
- J = moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- S = similarity measure
- arg max = optimization
- $\hat{T}$ = solution

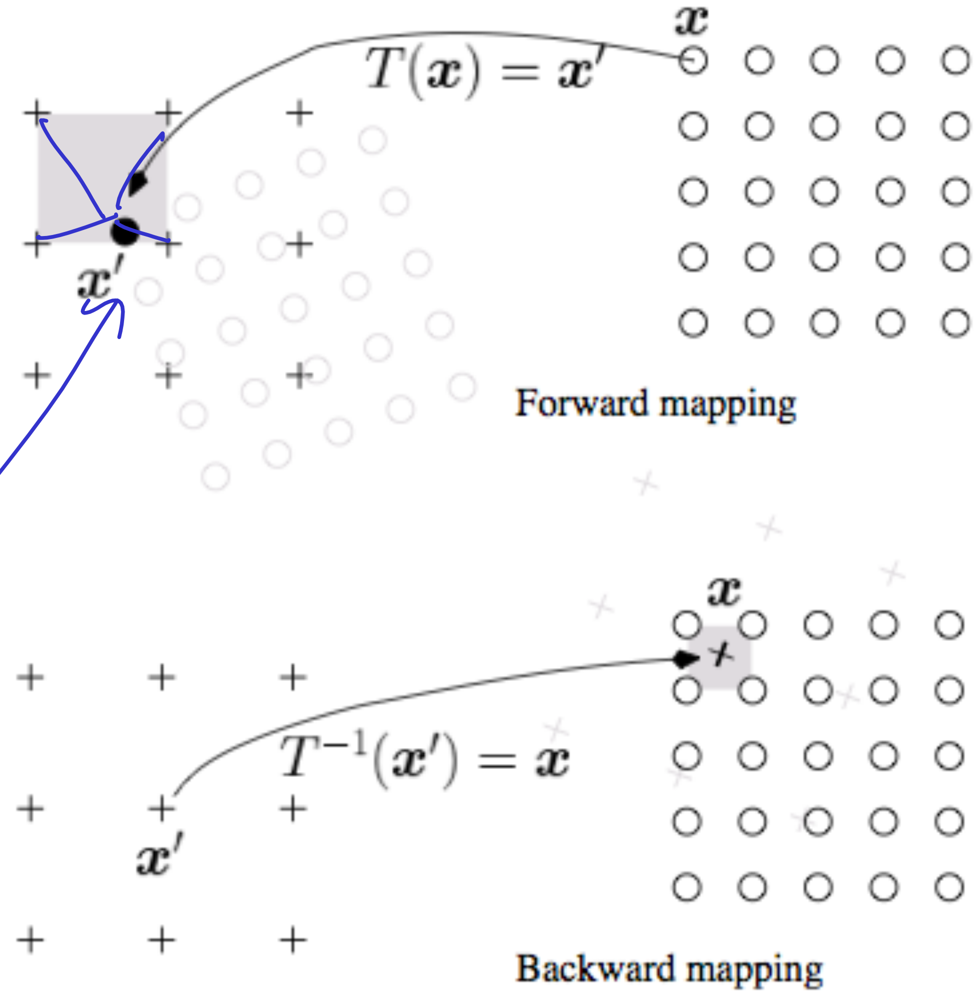
Similarity measure: intensity-based

Forward and backward mappings:

Required to compute $\cancel{J \circ T}$ $T(J)$

interpolation ! ***

ne tombe pas forcément sur un pixel



Similarity measure: intensity-based

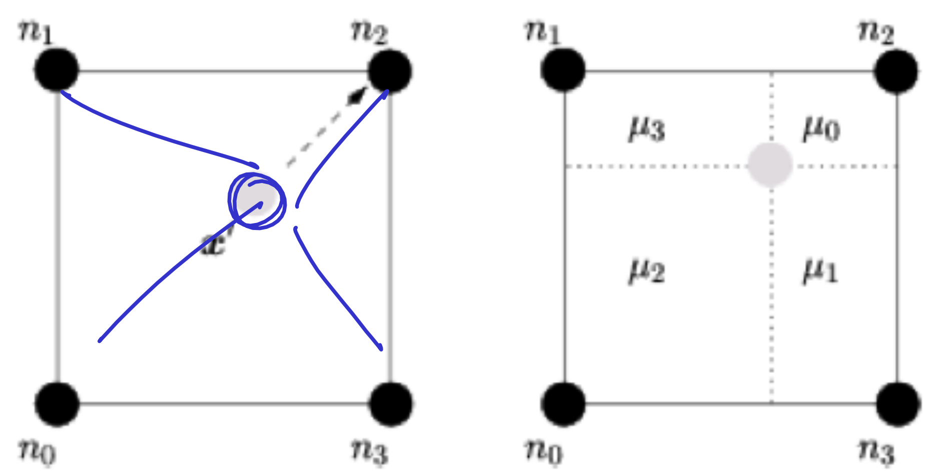
Forward and backward mappings:

Required to compute $J \circ T$

Forward requires an additional weight map + potential holes
Backward usually preferred: not a problem
- If T is invertible (ok with affine, otherwise should be enforced)
- Optimize T^{-1} and invert it at the end

Beware: **interpolation** required !
- Nearest neighbor
- Linear
- More accurate: quadratic, spline, …

= compromise between speed and accuracy



Similarity measure: intensity-based

(Dis-)similarity measure for intensities:

$$SSD(I,J) = \sum_{\boldsymbol{x}\in\Omega} (I(\boldsymbol{x}) - J(\boldsymbol{x}))^2$$

Sum-of-squared differences (SSD):

- Fast
- Needs to be **normalized** to the domain size (in voxels)
- Functional dependence between intensities:
  - **Ok if same modality + noise**
  - Impossible if multimodal !

sensible au bruit

- Similar to *Sum-of-absolute differences* (SAD)

Similarity measure: intensity-based

(Dis-)similarity measure for intensities:

$$CC(I,J) = \boxed{\frac{cov(I,J)}{\sigma_I \sigma_J}}$$

$$= N \frac{\sum_i (I_i(\mathbf{x}) - m_I)(J_i(\mathbf{x}) - m_J)}{\sqrt{\sum_i (I_i(\mathbf{x}) - m_I)^2} \sqrt{\sum_i (J_i(\mathbf{x}) - m_J)^2}}$$

Sum-of-squared differences (SSD)

(Pearson) correlation coefficient:

- **Range = [-1 , 1]**
  - 1 = perfect increasing linear relationship
  - - 1 = …
  - 0 = independent

  > Similarity measure = | $CC(I, J)$ |

- **Fast** (with computational tricks)

Similarity measure: intensity-based

(Dis-)similarity measure for intensities:

Sum-of-squared differences (SSD)

(Pearson) correlation coefficient:

$$CC(I,J) = \frac{cov(I,J)}{\sigma_I \sigma_J}$$

$$= N \frac{\sum_i (I(_i\boldsymbol{x}) - m_I)(J_i(\boldsymbol{x}) - m_J)}{\sqrt{\sum_i (I_i(\boldsymbol{x}) - m_I)^2} \sqrt{\sum_i (J_i(\boldsymbol{x}) - m_J)^2}}$$

- **Range = [-1 , 1]**
  - 1 = perfect increasing linear relationship
  - - 1 = …
  - 0 = independent

  > Similarity measure = $| \; CC(I , J) \; |$

- **Fast** (with computational tricks)

- **More robust** than SSD for 2 images from the same modality / **not good for multimodal**

- Functional dependence between intensities:
  - SSD : $I(\boldsymbol{x}) = J(T(\boldsymbol{x})) \quad \forall \boldsymbol{x}$
  - CC : $I(\boldsymbol{x}) = aJ(T(\boldsymbol{x})) + b \quad \forall \boldsymbol{x}$

Similarity measure: intensity-based

(Dis-)similarity measure for intensities:

Sum-of-squared differences (SSD)

(Pearson) correlation coefficient:

$$CC(I,J) = \frac{cov(I,J)}{\sigma_I \sigma_J}$$

$$= N \frac{\sum_i (I(_i\mathbf{x}) - m_I)(J_i(\mathbf{x}) - m_J)}{\sqrt{\sum_i (I_i(\mathbf{x}) - m_I)^2} \sqrt{\sum_i (J_i(\mathbf{x}) - m_J)^2}}$$

- **Range = [-1 , 1]**
  - 1 = perfect increasing linear relationship
  - \- 1 = …
  - 0 = independent

  > Similarity measure = | $CC(\ I\ ,\ J\ )$ |

- **Fast** (with computational tricks)

- **More robust** than SSD for 2 images from the same modality / **not good for multimodal**

\- Functional dependence between intensities:

- SSD : $I(\mathbf{x}) = J(T(\mathbf{x})) \quad \forall \mathbf{x}$
- CC : $I(\mathbf{x}) = a\,J(T(\mathbf{x})) + b \quad \forall \mathbf{x}$

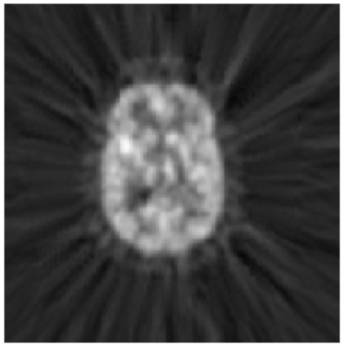
\- Other functional relationship? = **information theory:**

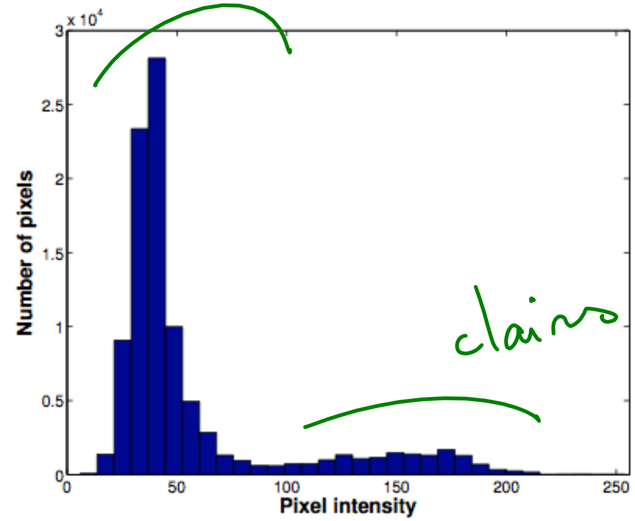
  \- *Mutual information, coefficient ratio…* both require **joint histograms**

Similarity measure: intensity-based

sombres

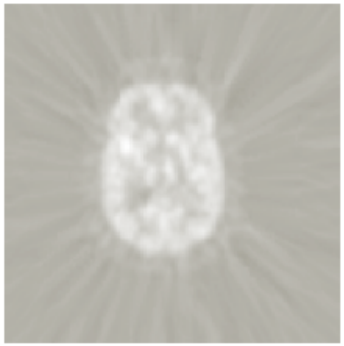
Histogram:

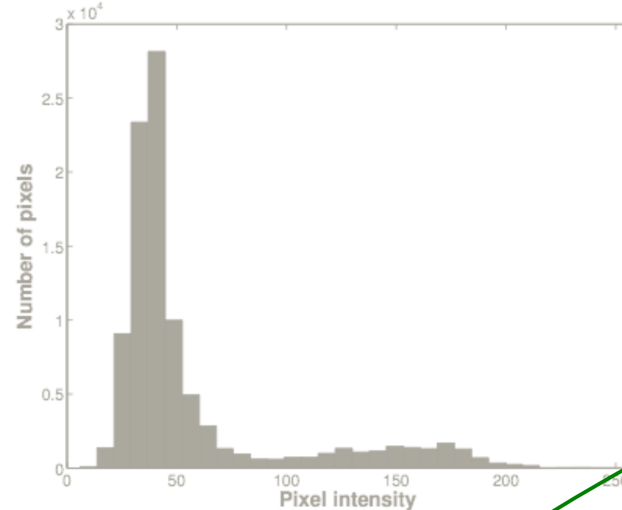




Similarity measure: intensity-based

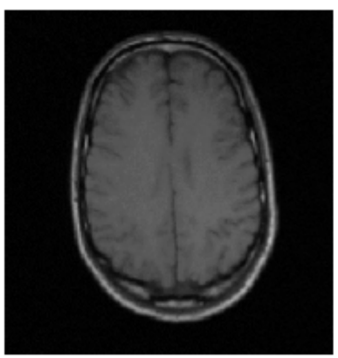
Histogram:

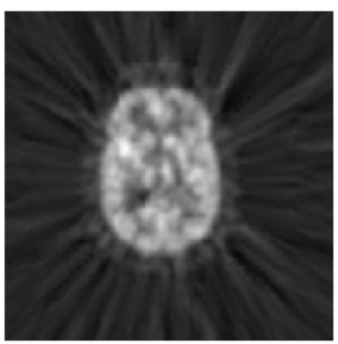


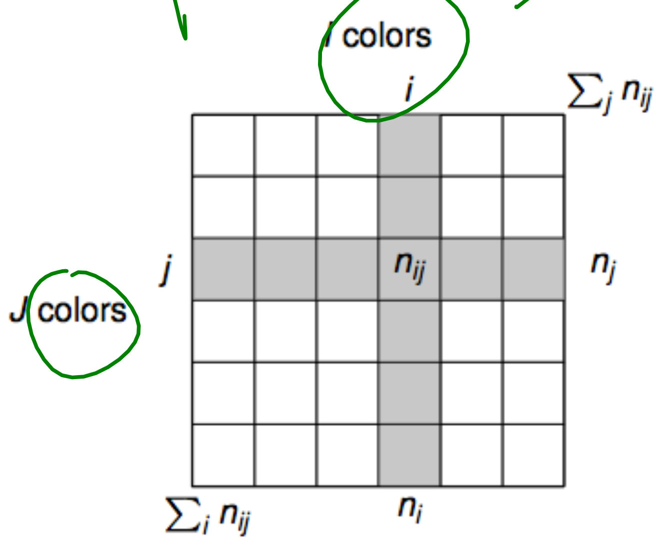


⚠ ! pas pixels mais nb bins dans l'histogram

Joint histogram:



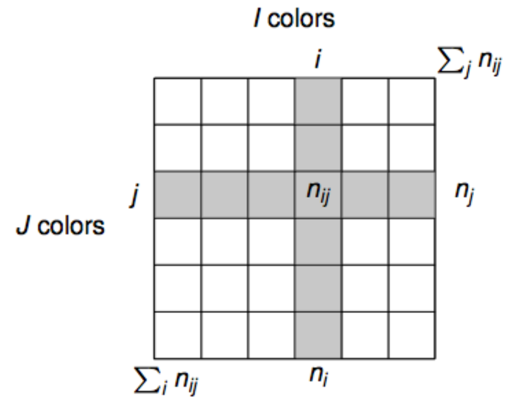




Not pixel indices !

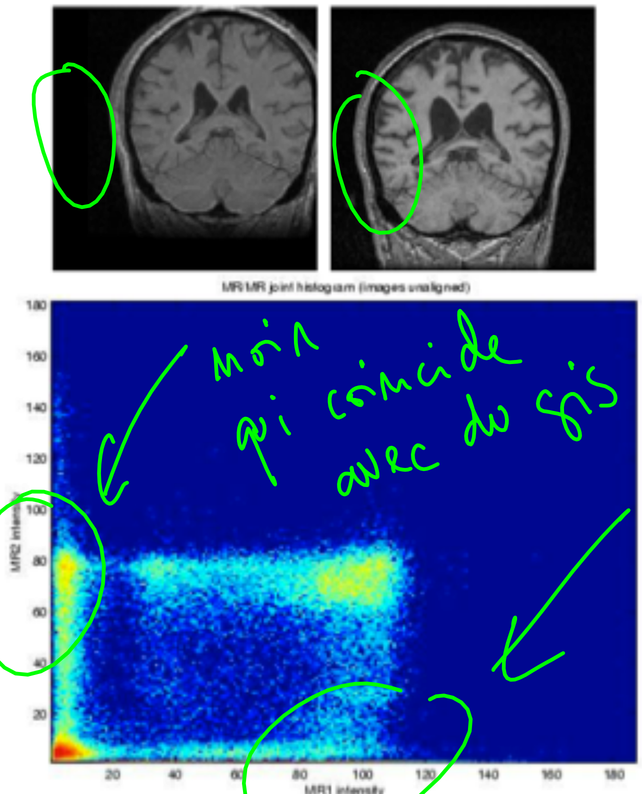
- i, j : (binned) colors
- n_{ij}: number of pixels with color i in I and j in J
- n_i, n_j: marginal values, i.e., histograms of I and J

Similarity measure: intensity-based



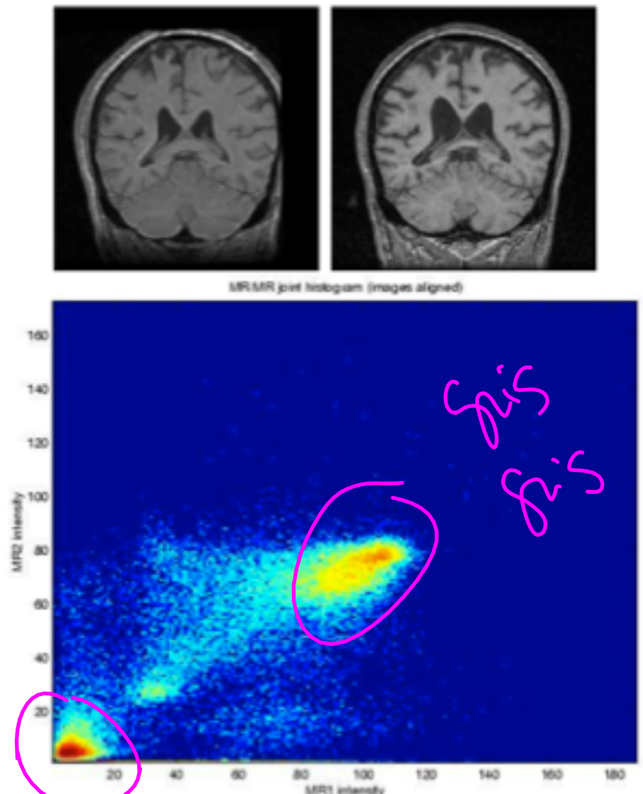
Joint histogram:

- **2D distribution** of the intensity pairs at each voxel
- Recomputed **at each step** of the optimisation



noir qui coïncide avec du gris

noir noir

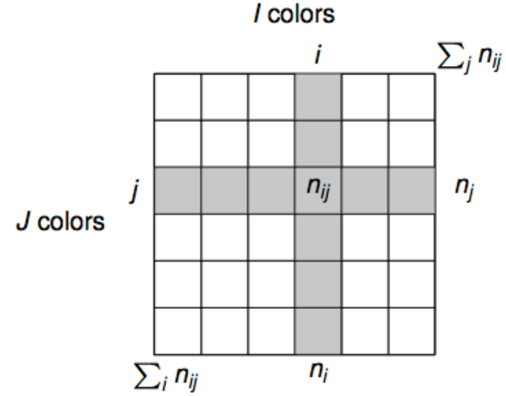


gris gris

Mono-modal

[Roche, *PhD*, 2001]

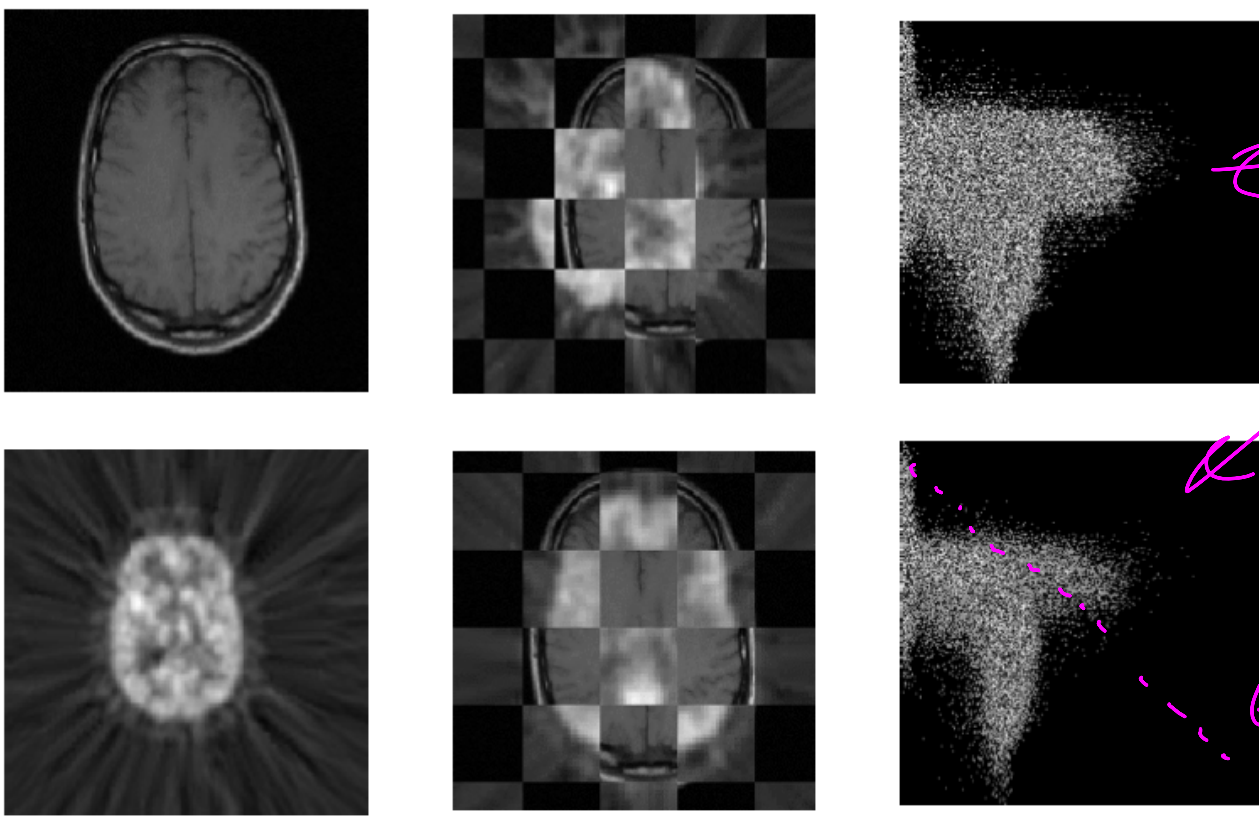
Similarity measure: intensity-based



Joint histogram:

- **2D distribution** of the intensity pairs at each voxel
- Recomputed **at each step** of the optimisation

plus subtil !



Multi-modal

diagonale inversée ?
(dans cette publi)

Similarity measure: intensity-based

Joint histogram: what for?

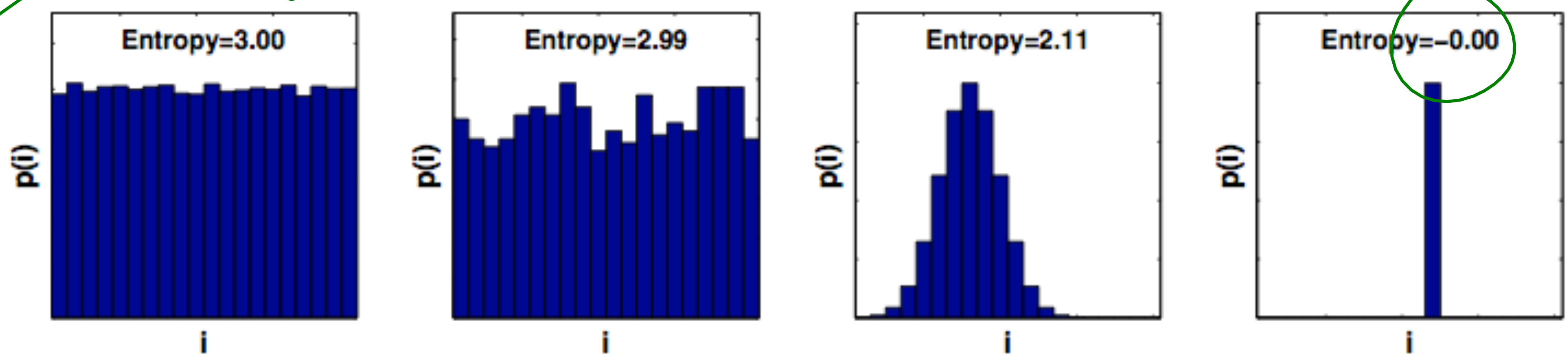
- **Mutual information – entropy** = measure of information as registration metric

Entropy (Shannon-Wiener):

$$H = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i$$

~ désordre

histogramme de 1 ~~seule~~ image



bruit pur | img bruitée | img | img constante

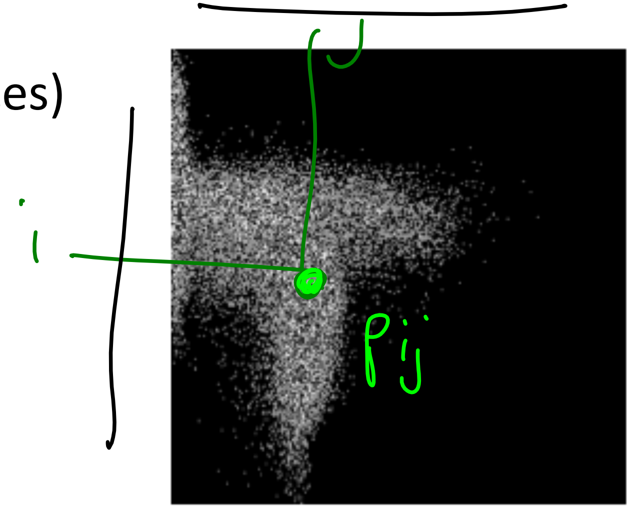
Similarity measure: intensity-based

Joint histogram: what for?

- Mutual information – entropy = measure of information as registration metric “colors”

- **Mutual information – joint entropy** (2 images = 2 subindices)

$$H = -\sum_{i,j} p_{ij} \log p_{ij}$$



> The more similar the distributions… = the lower the joint entropy (compared to the sum of individual entropies)

cf. + loin

$$H(I,J) \leq H(I) + H(J)$$

Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric

- **Mutual information – joint entropy** (2 images = 2 subindices)

$H(I,J) \leq H(I) + H(J)$

- **Mutual information:**

$$MI(I,J) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{p_i \, p_j}$$

$$MI(I,J) = H(I) - H(J|IJ) = H(J) - H(I|J) = H(I) + H(J) - H(I,J) \geq 0$$

3.82 6.79 6.98 7.15

[Pluim et al. *IEEE TMI*, 2003]

Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric

- **Mutual information – joint entropy** (2 images = 2 subindices)

- **Mutual information:** $$MI(I,J) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{p_i \, p_j}$$

  - Positive

  - Symmetric: $MI(\ I\ ,\ J\ ) = MI(\ J\ ,\ I\ )$ * *

  - $H(\ J\ |\ I\ ) = H(\ I\ ,\ J\ ) - H(\ I\ )$ = conditional entropy

Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric
- **Mutual information – joint entropy** (2 images = 2 subindices)
- **Mutual information**
- **Normalized mutual information** **
    - **Goal** = overcome the **sensitivity of MI** to variations of the overlap size between I and $J \circ T$
    - Several versions
    - Example = [Studholme et al. Patt Recogn, 1999]

$$NMI(I,J) = \frac{H(I) + H(J)}{H(I,J)}$$

- It's a reference !!!

  First articles: 1995 (Maes et al. [Belgium], Viola et al. [US])
  First journal articles: 1997
  Among the most cited articles in the field…

***

Similarity measure: intensity-based

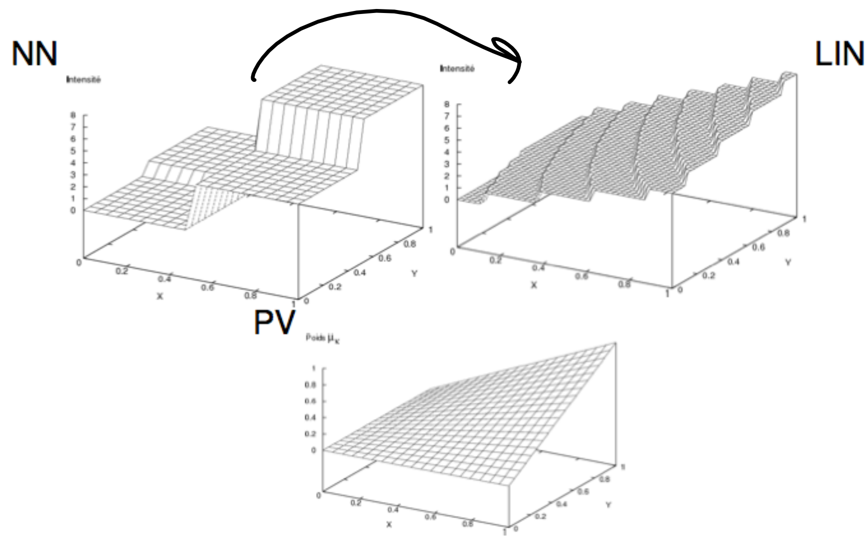
Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric
- **Mutual information – joint entropy** (2 images = 2 subindices)
- **Mutual information**
- **Normalized mutual information**
  - **Currently = #1 similarity measure !** Multi-modal + robust +++
  - **Implementation matters** (as usual…)
    - Survey of methods = [Pluim et al. *IEEE TMI* 2003]

Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric
- **Mutual information – joint entropy** (2 images = 2 subindices)
- **Mutual information**
- **Normalized mutual information**
  - **Currently = #1 similarity measure !** Multi-modal + robust +++
  - **Implementation matters** (as usual…)
    - Specifically:
      - Interpolation

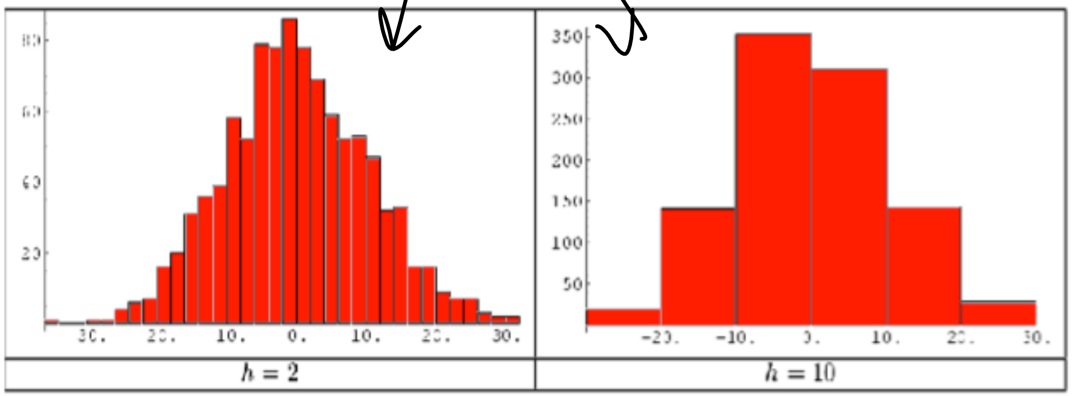


Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric
- **Mutual information – joint entropy** (2 images = 2 subindices)
- **Mutual information**
- **Normalized mutual information**
  - **Currently = #1 similarity measure !** Multi-modal + robust +++
  - **Implementation matters** (as usual…)
    - Specifically:
      - Interpolation
      - Binning

****



Similarity measure: intensity-based

Joint histogram: what for?

- **Mutual information – entropy** = measure of information as registration metric
- **Mutual information – joint entropy** (2 images = 2 subindices)
- **Mutual information**
- **Normalized mutual information**
  - **Currently = #1 similarity measure !** Multi-modal + robust +++
  - **Implementation matters** (as usual…)
    - Specifically:
      - Interpolation
      - Binning
      - Probability computation
      - Normalization (NMI)

Similarity measure: intensity-based

Summary:

- The measures can be classified vs. hidden variables [Malandain, *HDR*, 2006]
- SSD / SAD 0 variable
- CC 2 variables: slope + intercept
- MI size of the joint histogram matrix

- **MI is more general**, hence its success…
- … but more susceptible to **fail in simple situations** / where SSD is enough (e.g. time series)

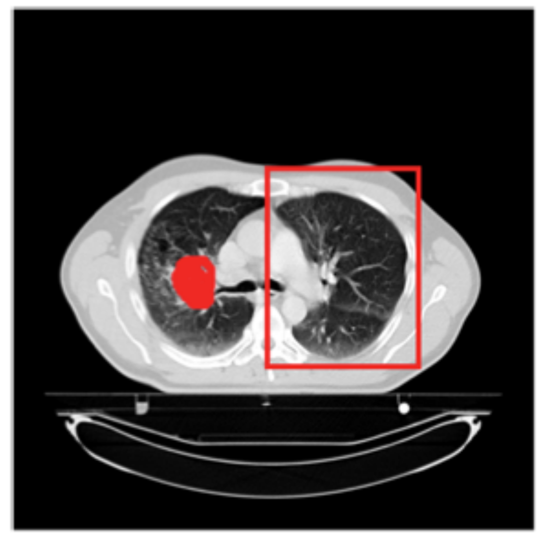
Similarity measure: intensity-based

Summary:

- The measures can be classified vs. hidden variables [Malandain, *HDR*, 2006]
- SSD / SAD 0 variable
- CC 2 variables: slope + intercept
- MI size of the joint histogram matrix

- **MI is more general**, hence its success…
- … but more susceptible to **fail in simple situations**

Remark: **region of interest (ROI)**
> applied to the reference and/or the target
(depending on the similarity measure)



Optimization

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Notation

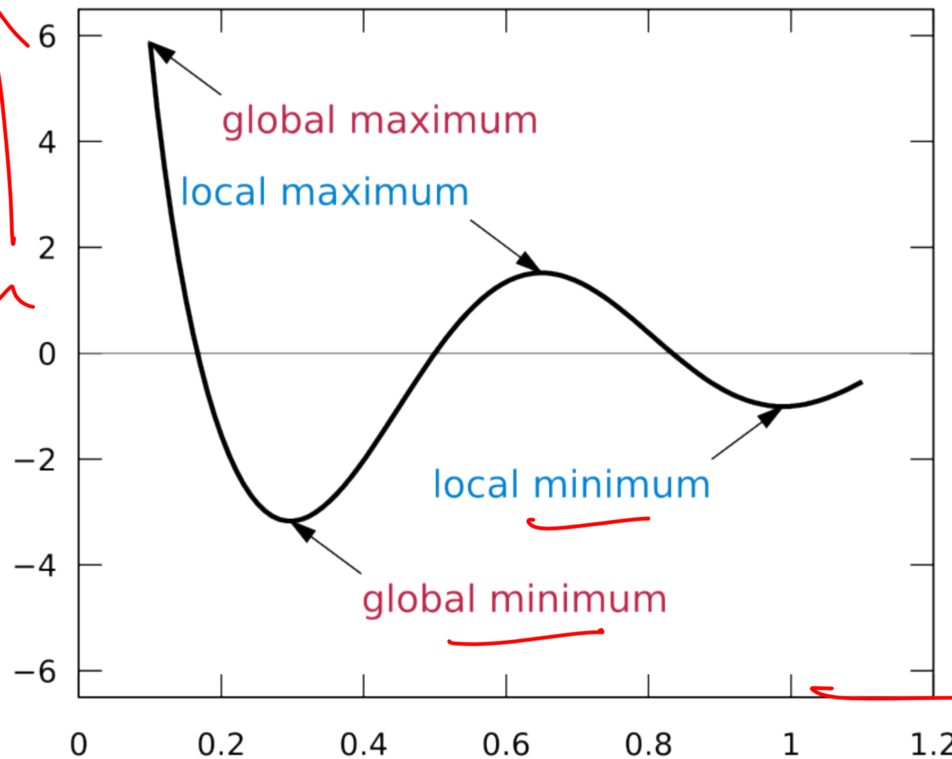
- I = reference / fixed image
- J = moving / floating image
- T = transformation
- $\mathcal{T}$ = search space
- S = similarity measure
- arg max = optimization
- $\hat{T}$ = solution

Optimization

maximum

Challenge = find the **global** minimum... as fast as possible

Energie à minimiser



!

params.

ex : deep : params = poids du réseau

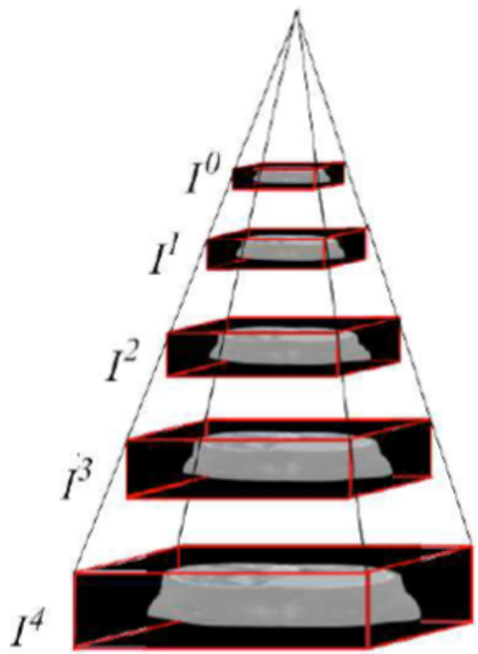
ex : recalage : params : $\boxed{\lambda_1, \lambda_2, \lambda_3, \lambda \ldots}$ = R

Optimization

Challenge = find the **global** minimum… as fast as possible

- **Maximize or minimize** (depending on the energy formulation)
- With or without gradient (if *f* is differentiable)
- **Multi-resolution** *
  - Convergence
  - Robustness to noise + local minima
  - Large deformations

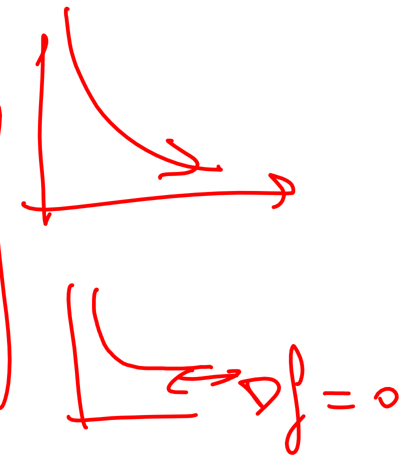
$T_{finale} = T_4 \circ T_3 \circ \dots$



Optimization

Challenge = find the **global** minimum… as fast as possible

- **Maximize or minimize** (depending on the energy formulation)
- With or without gradient (if *f* is differentiable)
- **Multi-resolution**
- Iterative process — iteration $\boldsymbol{x}_k \,/\, f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k)$
- **Stopping criterion** =
  - $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| < \epsilon$
  - $\nabla f(\hat{\boldsymbol{x}}) = 0$
- Update = search $\delta \boldsymbol{x}_k \,/\, \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta \boldsymbol{x}_k$



Optimization

Challenge = find the **global** minimum… as fast as possible

- **Maximize or minimize** (depending on the energy formulation)
- With or without gradient (if *f* is differentiable)
- Multi-resolution
- Iterative process $\quad$ iteration $\boldsymbol{x}_k \;/\; f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k)$
- **Stopping criterion** =
  - $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| < \epsilon$
  - $\nabla f(\hat{\boldsymbol{x}}) = 0$
- Update = $\quad$ search $\delta\boldsymbol{x}_k \;/\; \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta\boldsymbol{x}_k$
- Importance of initialization !!! $\quad \boldsymbol{x}_0$
  - **Pre-align** image origins / centers / mass centers
  - **Rigid alignment before** non-rigid

Optimization

Process / methods = cf. supplementary slides [RIT-SARRUT_optimization.pdf]

Optimization

Summary:

- **Without gradient**
  - Simple
  - Powell (conjugate directions)
- **With gradient:**
  - Gradient descent
  - Conjugate gradient
  - Quasi-Newton
  - Levenberg-Marquart
- **Also:**
  - Genetic algorithms
  - Simulated annealing
  - ...

Validation

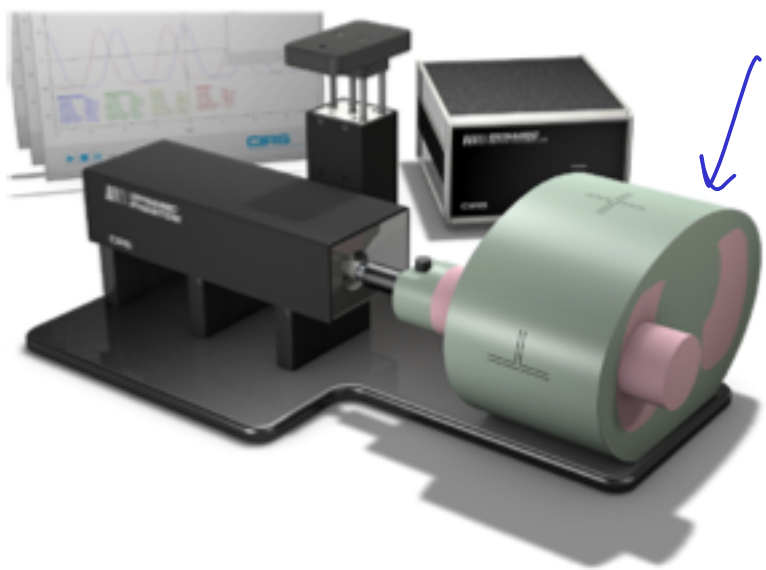
(further developed in D. Sarrut's part)

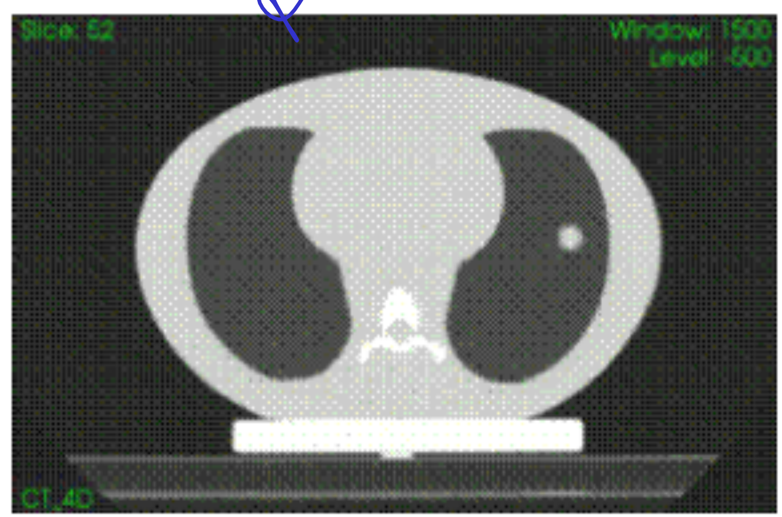
	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	− ⟶				+
Clinical Realism	+ ⟵				−

Validation

(further developed in D. Sarrut's part)

	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	− ⟶	⟶	⟶	⟶	⟶ +
Clinical Realism	+ ⟵	⟵	⟵	⟵	⟵ −

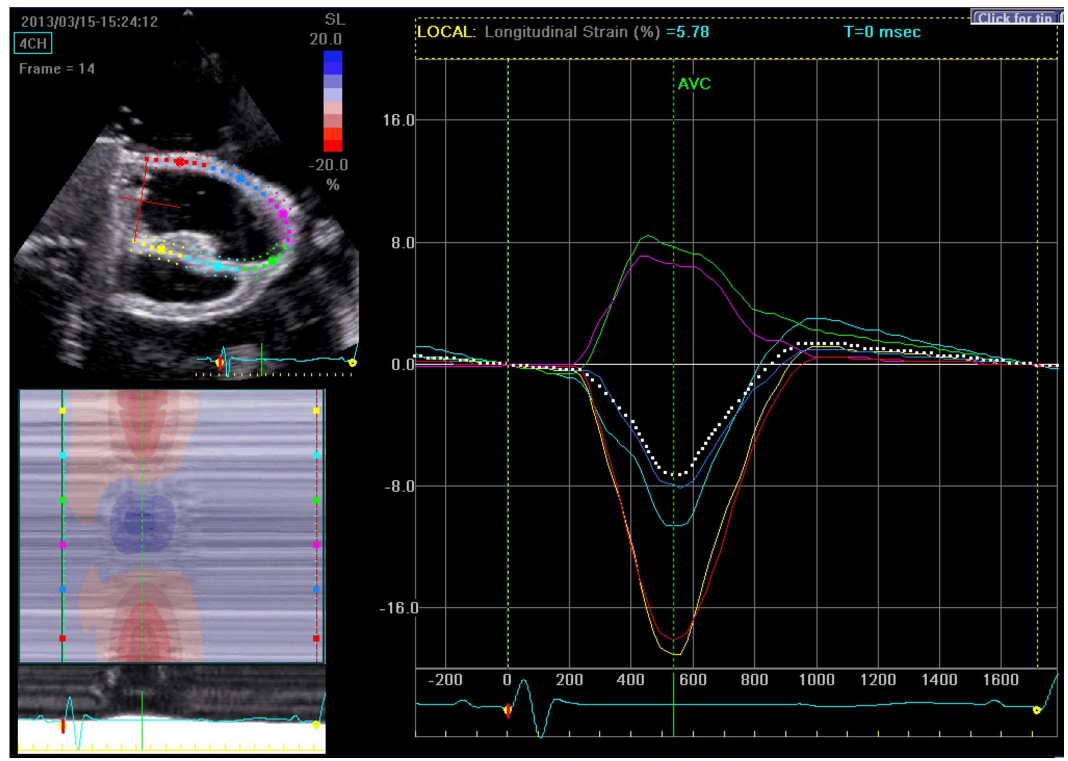




Validation

(further developed in D. Sarrut’s part)

	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	− ⟶				+
Clinical Realism	+ ⟵				−

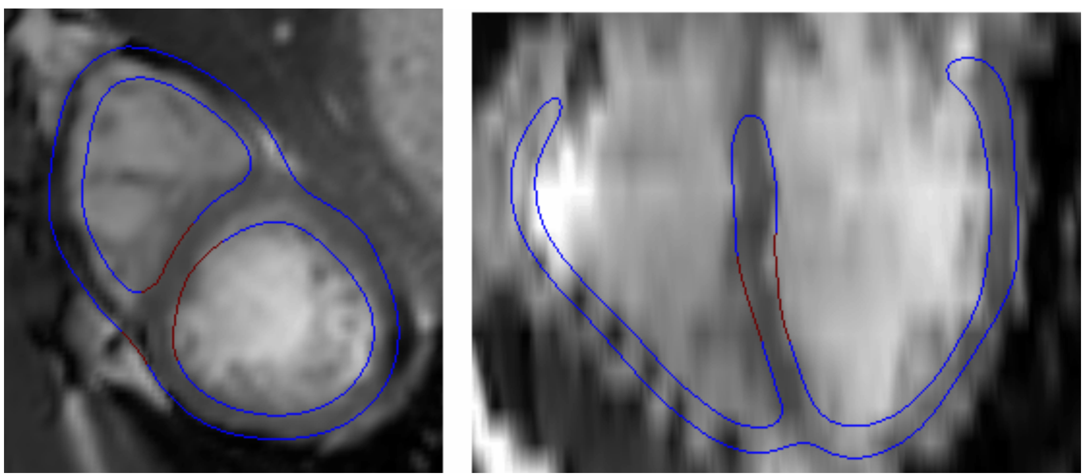


CLÍNIC
BARCELONA
Hospital Universitari

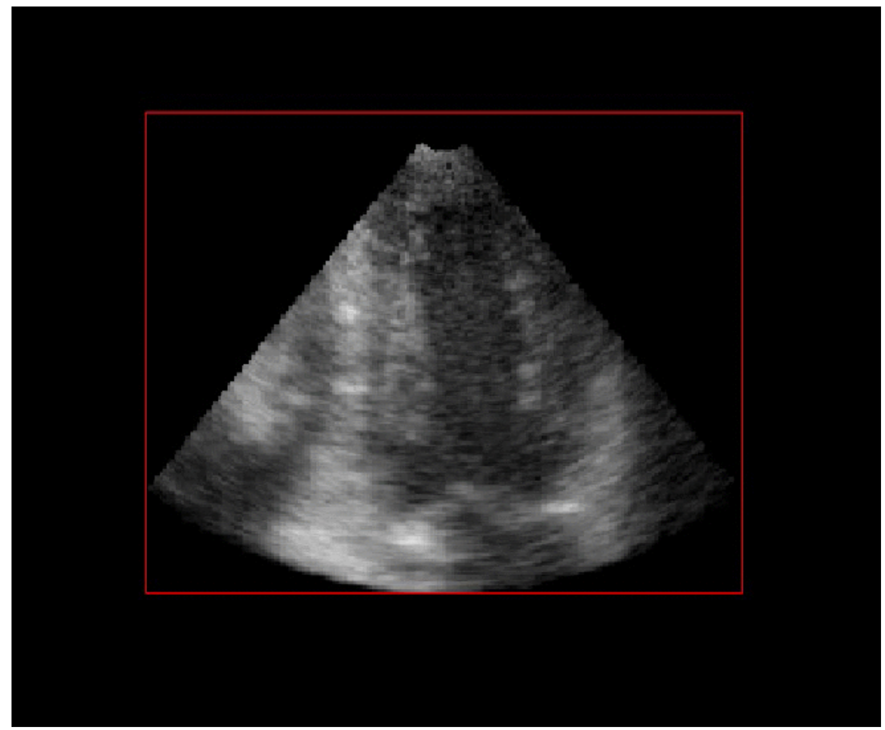
Validation

(further developed in D. Sarrut's part)

	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	–				+
Clinical Realism	+				–



[Duchateau et al. *IEEE TMI*, 2018]



[Zhou et al. *IEEE TMI*, 2018]

Validation

kaggle ...

[(further developed in D. Sarrut's part)]

+ challenges

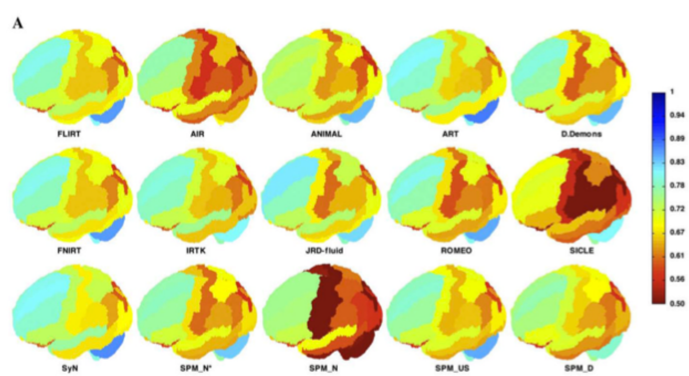
conf: MICCAI...

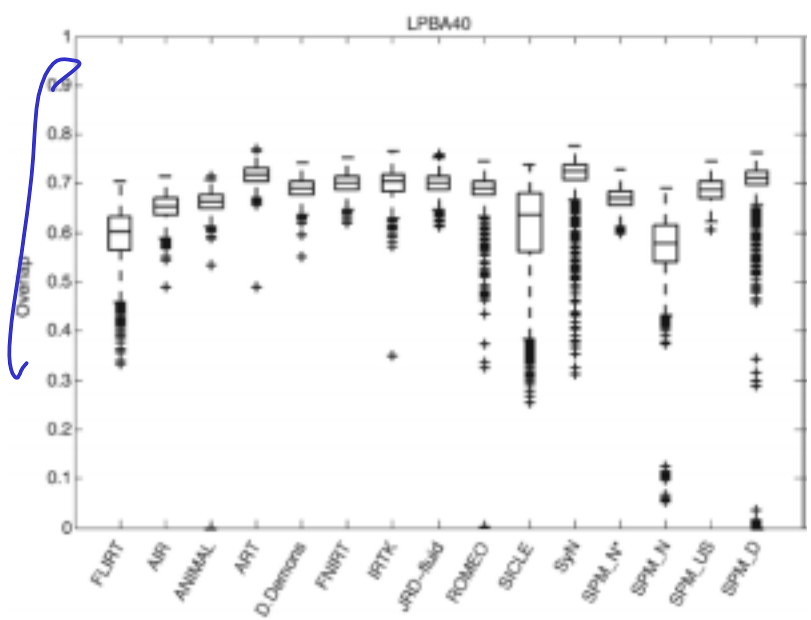
	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	–	⟶	⟶	⟶	+
Clinical Realism	+	⟵	⟵	⟵	–

Neuroimage. 2009 July 1; 46(3): 786–802. doi:10.1016/j.neuroimage.2008.12.037.

Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration

Arno Klein[a,*], Jesper Andersson[b], Babak A. Ardekani[c,d], John Ashburner[e], Brian Avants[f], Ming-Chang Chiang[g], Gary E. Christensen[h], D. Louis Collins[i], James Gee[f], Pierre Hellier[j,k], Joo Hyun Song[h], Mark Jenkinson[b], Claude Lepage[i], Daniel Rueckert[m], Paul Thompson[g], Tom Vercauteren[n,l], Roger P. Woods[o], J. John Mann[a], and Ramin V. Parsey[a]





Validation

(further developed in D. Sarrut's part) + challenges

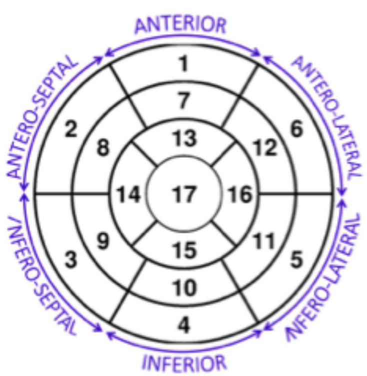
	Clinical data	Cadavers	Physical phantoms	Realistic simulations	Numerical simulations
Easy Control	–	⟶	⟶	⟶	+
Clinical Realism	+	⟵	⟵	⟵	–

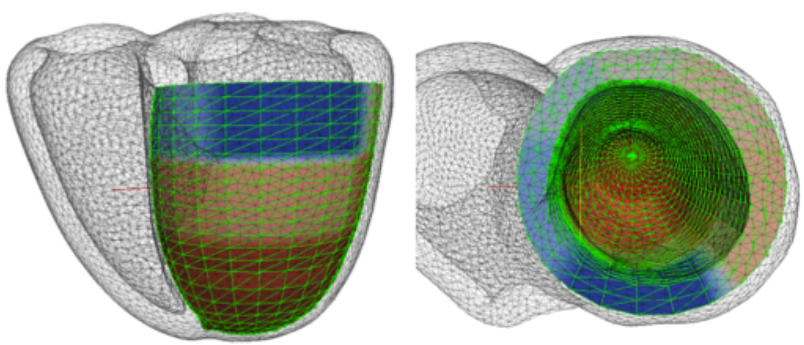
IEEE Trans Med Imaging. 2016 Aug;35(8):1915-26. doi: 10.1109/TMI.2016.2537848. Epub 2016 Mar 3.

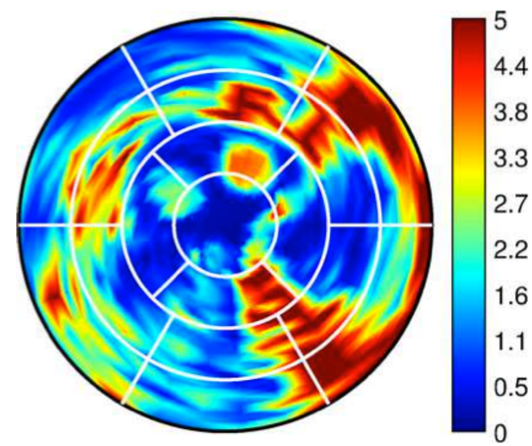
Detailed Evaluation of Five 3D Speckle Tracking Algorithms Using Synthetic Echocardiographic Recordings.

mouvement cardiaque

Alessandrini M, Heyde B, Queiros S, Cygan S, Zontak M, Somphone O, Bernard O, Sermesant M, Delingette H, Barbosa D, De Craene M, ODonnell M, Dhooge J.







Conclusion

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

pour du recalage non-rigide

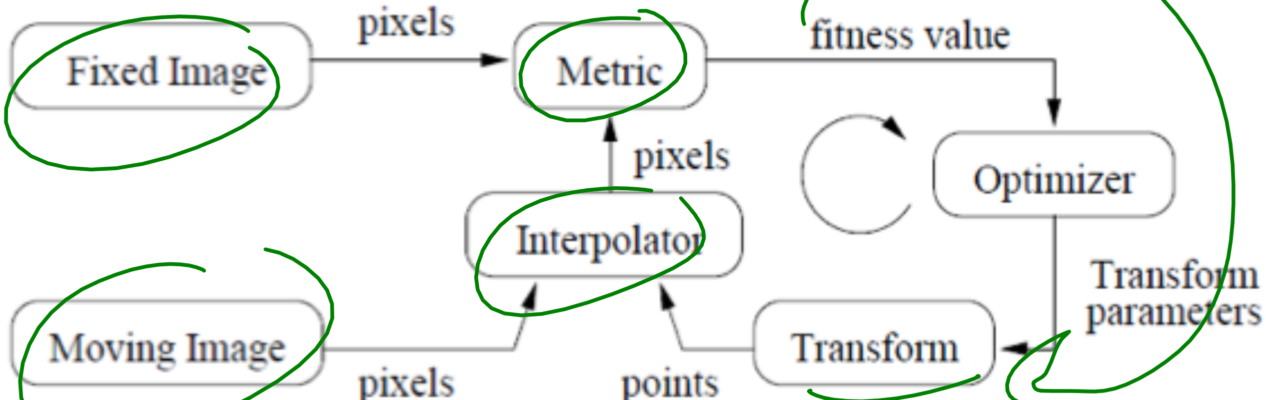
+ Regularisation

Reg(T)

Conclusion

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

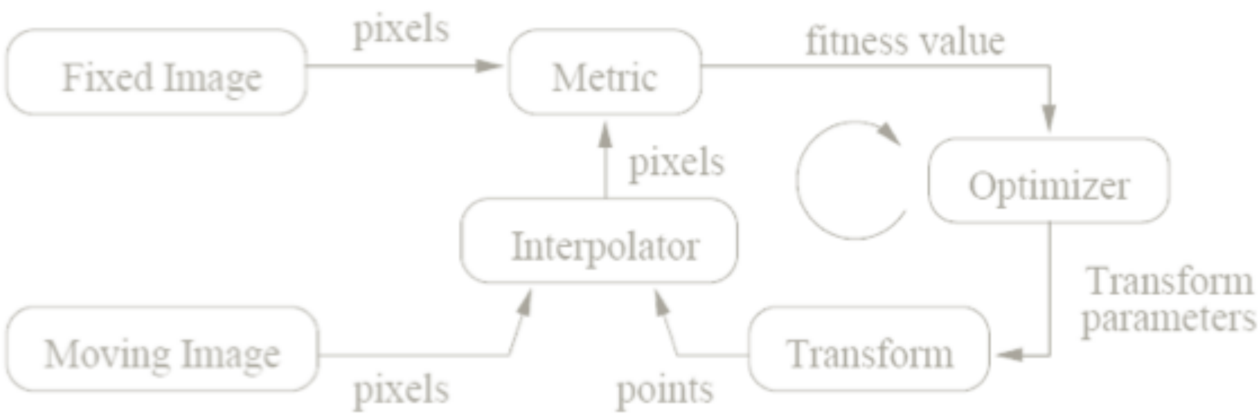
Your own solution…



Conclusion

$$\hat{T} = \arg\max_{T \in \mathcal{T}} S(I, J, T)$$

Your own solution…



… or existing solutions

- ITK / VTK
- Elastix, Deformetrica, …