Welcome

What did I do?

What did I do?

- Programming

Printed by Jan Kybic

Apr 26, 01 16:44 **splninterp.c** Page 1/7

```c
/* Interpolation of a volume described by B-spline coefficients
   at arbitrary points.

   Jan Kybic, 1999
   $Id: splninterp.c,v 1.1.1.1.2.6 2001/04/26 14:44:47 cvsuser Exp $
*/

#ifdef BIGSPLINES
#include "BIGsplines.h"
#include <math.h>
#endif


extern int mfoldmirroronbound(int k,int n)
/* having a signal 0..n-1, fold k using mirror on boundary conditions,
   i.e. k=n-1 gives n-1, k=n gives n-2, k=-1 gives 1 etc. */
{ int m ;
  if (n<=1) return 0 ;
  m=2*(n-1) ;
  k=(k<0) ? k%m+m : k%m ;
  return k>=n ? m-k : k ;
}

extern double mfolddmirroronbound(double k,int n)
/* having a signal 0..n-1, fold k using mirror on boundary conditions,
   i.e. k=n-1 gives n-1, k=n gives n-2, k=-1 gives 1 etc. */
{ int m,q ;
  m=2*(n-1) ;
  q=floor(k/m) ; k-=q*m ;
  return k>n-1 ? m-k : k ;
}

extern int mfoldmirroroffbound(int k,int n)
/* having a signal 0..n-1, fold k using mirror off boundary conditions,
   i.e. k=n-1 gives n-1, k=n gives n-1, k=-1 gives 0 etc. */
{ int m ;
  if (n<=1) return 0 ;
  m=2*n ;
  k=(k<0) ? k%m+m : k%m ;
  return k>=n ? m-1-k : k ;
}

extern double mfolddmirroroffbound(double k,int n)
/* having a signal 0..n-1, fold k using mirror off boundary conditions,
   i.e. k=n-1 gives n-1, k=n gives n-1, k=-1 gives 0 etc. */
{ int m,q ;
  m=2*n ;
  q=floor(k/m) ; k-=q*m ;
  return k>n-1 ? m-1-k : k ;
}

extern int evalbspln(double *x,double *y, int n, int degree)
/* Evaluates B-spline of degree 'degree' at n-points x[0],...,x[n-1] */
/* results are put into y[0],..,y[n-1] */
{ int i,supp ;
  double (*evsplnx)(double) ;    /* pointer to the evaluation functions */
  if (choosespln(degree,&evsplnx,&supp,0)) {
    myErrMsg("Unsupported degree.") ;
    return 1 ;
  }

  for (i=0;i<n;i++) y[i]=(*evsplnx)(x[i]) ;
```

Apr 26, 01 16:44 **splninterp.c** Page 2/7

```c
  return 0 ;
}



extern int    splinterp
/* Takes an input matrix of size nxi*nyi*nzi, containing B-spline
    coefficients of degree 'degree'. Samples the resulting function at
    nxc*nyc*nzc points given by coord, each point described by ndc
    coordinates. Uses mirror boundary conditions. Everything should be
    allocated in advance.
    In this routine, x is simply the index that changes fastest and
    z the index which changes slowest. C convention is applied for indexing,
    i.e. the first element of 'input' is assumed to correspond to
    point (0,0,0). If coord gives multidimensional coordinates, they are laid
    consecutively, i.e., as the fastest changing (sub)index - even faster
    than x.
*/
(
 double *input,                /* B-spline coefficients */
 double *coord,                /* 3D coordinates */
 double *output,               /* output values, one for each coord */
 int    nxi, int nyi,int nzi,          /* input size */
 int    nxc,int nyc,int nzc,           /* output size */
 int    ndc,                   /* coordinate range dimensionality */
 int    degree,                /* 0 - Haar, 1 - Linear etc. */
 int    dflag,                 /* 0 - value, 1,4,16 - first derivative with
                                  respect to x,y,z ; 2,8,32 - second der. */
 int bcond
 )
{
  int ix,iy,iz,ofs,jx,jy,jz,kz,lx,ly,lz ;
  double (*evsplnx)(double) ;    /* pointer to the evaluation functions */
  double (*evsplny)(double),(*evsplnz)(double) ;
  int supp; double hsupp ;      /* spline support */
  double x,y,z,sum,sumt ;
  double *tabx,*taby,*ptr ;
  int *foldx,*foldy ;
  int (*mfold)(int,int) ; double (*mfoldd)(double,int) ;

  switch(bcond) {
      case MirrorOffBounds:
        mfold=mfoldmirroroffbound ;
        mfoldd=mfolddmirroroffbound ;
        break ;
      case MirrorOnBounds:
        mfold=mfoldmirroronbound ;
        mfoldd=mfolddmirroronbound ;
        break ;
  default:
        myErrMsg("Unsupported boundary conditions.") ;
        return 1 ;
  }



  //  mexPrintf("splninterp called with nxi=%d nyi=%d nzi=%d nxc=%d nyc=%d nzc=%
d ndc=%d degree=%d\n", nxi, nyi, nzi, nxc, nyc, nzc, ndc, degree) ;
  if (choosespln(degree,&evsplnx,&supp,dflag & 3) |
      choosespln(degree,&evsplny,NULL,(dflag & 12)>>2) |
      choosespln(degree,&evsplnz,NULL,(dflag & 48)>>4)) {
```

Tuesday July 17, 2001 splninterp.c 1/4

What did I do?

- Programming
- Mathematics

$$B(\mathbf{f},\mathbf{g}) = \int_{\mathbb{R}^{2m}} \mathbf{f}^T(\mathbf{x})\mathsf{V}(\mathbf{x},\mathbf{y})\mathbf{g}(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}$$

$$B(\mathbf{f},\mathbf{g}) = \frac{1}{(2\pi)^m}\int_{\mathbb{R}^m} \hat{\mathbf{f}}^T(\boldsymbol{\omega})\,\hat{\mathsf{U}}(\boldsymbol{\omega})\,\hat{\mathbf{g}}^*(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}$$

Lemma 0 *A function* $\mathbf{f}_{out}$ *from* F *satisfying* $\langle \mathsf{H}, \mathbf{f}_{out}\rangle = \mathbf{s}$ *solves the variational problem* $\mathcal{P}$, *if and only if there is a real vector* $\boldsymbol{\lambda}$ *such that for all* $\mathbf{g} \in F$

$$B(\mathbf{f}_{out},\mathbf{g}) = \boldsymbol{\lambda}^T\langle \mathsf{H},\mathbf{g}\rangle$$

$$D^M f = \left[\frac{\partial^M f}{\partial x_1^M},\ldots,\frac{\partial^M f}{\partial x_{k_1}\ldots\partial x_{k_M}},\ldots,\frac{\partial^M f}{\partial x_m^M}\right]$$

$$\mathrm{E}\left[\frac{\partial^\alpha}{\partial t^\alpha}B_H(t)\,\frac{\partial^\alpha}{\partial s^\alpha}B_H(s)\right] = -\frac{\partial^\alpha}{\partial t^\alpha}\frac{\partial^\alpha}{\partial s^\alpha}|t-s|^{2H} = \rho(t-s)$$

$$\underbrace{\begin{bmatrix}\mathsf{A} & \mathsf{Q}_1\\ \mathsf{Q}_2 & \mathbf{0}\end{bmatrix}}_{\mathsf{B}}\begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{a}\end{bmatrix} = \begin{bmatrix}\mathbf{s}\\ \mathbf{0}\end{bmatrix}$$

$$\frac{\partial E}{\partial c_{\mathbf{j},m}} = \sum_{\mathbf{i}\in I_b}\frac{\partial e_{\mathbf{i}}}{\partial f_w(\mathbf{i})}\,\frac{\partial f_t^c(\mathbf{x})}{\partial x_m}\bigg|_{\mathbf{x}=\mathbf{g}(\mathbf{i})}\beta_{n_m}\left(\mathbf{i}/\mathbf{h}-\mathbf{j}\right)$$

What did I do?

- Programming
- Mathematics
- Papers

1

Unwarping of Unidirectionally Distorted EPI Images

Jan Kybic†, Philippe Thévenaz, Arto Nirkko and Michael Unser

Abstract— **Echo-planar imaging (EPI) is a fast nuclear magnetic resonance imaging method. Unfortunately, local magnetic field inhomogeneities induced mainly by the subject's presence cause significant geometrical distortion, predominantly along the phase-encoding direction, which must be undone to allow for meaningful further processing. So far, this aspect has been too often neglected.**

In this paper, we suggest a new approach using an algorithm specifically developed for the automatic registration of distorted EPI images with corresponding anatomically correct MRI images. We model the deformation field with splines, which gives us a lot of flexibility while comprising the affine transform as a special case. The registration criterion is least-squares. Interestingly, the complexity of its evaluation does not depend on the resolution of the control grid. The spline model gives us good accuracy thanks to its high approximation order. The short support of splines leads to a fast algorithm. A multiresolution approach yields robustness and additional speed-up.

The algorithm was tested on real as well as synthetic data, and the results were compared with a manual method. A wavelet-based Sobolev-type random deformation generator was developed for testing purposes. A blind test indicates that the proposed automatic method is faster, more reliable, and more precise than the manual one.

Keywords—**image registration, splines, geometrical distortion, unwarping**

I. Introduction

A. EPI features

Echo planar imaging (EPI) [1] is a fast magnetic resonance imaging (MRI) technique permitting an acquisition of a two-dimensional slice using a single excitation, which leads to very short scan times. It is used mainly for functional imaging (fMRI), the *in vivo* non-invasive study of the temporal, spatial and behavioral dependencies of brain activations. The basis of fMRI lies in the fact that deoxyhemoglobin (the hemoglobin without a bound oxygen molecule) is paramagnetic. Neural activation in the cerebral cortex leads to an increase of blood flow, hence to a decrease of deoxyhemoglobin concentration.[1] This results in a measurable alteration of the magnetic field and in a consequent increase of signal intensity in the appropriately weighted MRI images (blood oxygen-level dependent, BOLD). It is therefore difficult to compensate for the unwanted magnetic field inhomogeneities induced mainly by the spatially varying magnetic susceptibility of the subject [2]. In contrast to conventional MRI, where the number of excitations per slice is equal to the number of scan lines, in EPI the magnetic field gradients have to encode two coordinates simultaneously in one excitation. As one of the gradients (the so-called phase-encoding gradient) is several orders of magnitude weaker than the other, the inhomogeneous magnetic field will manifest itself mainly as a geometrical distortion of the 2D slice image along the direction of this gradient. This effect is clearly visible in Figure 1. Since the stronger gradient is less affected, the distortion is essentially unidirectional. Letting g be the unknown warping (deformation) function, we have

$$f^o(g(x,y),y) \simeq f^u(x,y) \tag{1}$$

where f^o is the observed EPI image and f^u is the hypothetical ideal undistorted EPI image. We can consider each slice separately, as the shift in the z axis due to patient's movement is insignificant because his head is attached. Should there be such a displacement, it can by readily corrected by existing algorithms [3].

B. The reasons to unwarp

The amplitude of the deformation g can be as large as 3–5 mm [4] (confirmed by our own observations), which typically amounts to several pixels. In some cases, as in Figure 1, specifically intended to illustrate EPI distortion, the deformation can be even more pronounced. Moreover, g can vary significantly from slice to slice and from acquisition to acquisition. For localization applications like stereotactic surgery, this inaccuracy is much larger than the required limit of 1 mm and therefore EPI cannot currently be used to this end. It also severely hinders the performance of the statistical processing of sets of fMRI images used to obtain activation information. Since the task-induced signal changes represent typically only 5–10% of the mean signal intensity in fMRI [1, 5], they will not stand out clearly unless the perturbations caused by the deformation g are undone.

C. Existing distortion correction techniques

One approach consists in changing the acquisition procedure [2, 4, 6]. However, this is often not practical due to technical or organizational limitations, for example lack of support or approval. Furthermore, while the alternative acquisition sequences reduce the distortion, the distortion is never removed completely, and the methods usually sacrifice either sensitivity or acquisition speed.

The second group of methods uses a two-step procedure [4, 7]. First, a field map or a deformation map is

† indicates corresponding author. Jan Kybic, Philippe Thévenaz, and Michael Unser are with Biomedical Imaging Group, DMT/IOA, Swiss Federal Institute of Technology Lausanne, CH-1015 Lausanne EPFL, Switzerland, email: Jan.Kybic@epfl.ch. Arto Nirkko is with Inselspital, Bern, Switzerland

[1] This effect prevails over the increase of oxygen consumption.

What did I do?

- Programming
- Mathematics
- Papers
- Conferences

Elastic Image Registration using Parametric Deformation Models

Jan Kybic

Overview

- Registration and its applications
- Manual registration
  - Interpolation
  - Variational reconstruction
- Splines
- Automatic registration
  - Algorithm
  - Semi-automatic registration
  - Applications
- Conclusions
- Party

Overview

- Registration and its applications
- Manual registration
  - Interpolation
  - Variational reconstruction
- Splines
- Automatic registration
  - Algorithm
  - Semi-automatic registration
  - Applications
- Conclusions
- Party

What did we do?

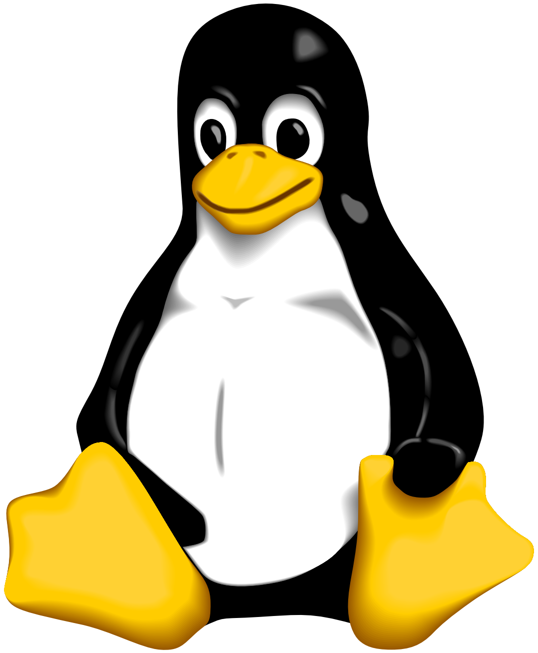
Overview

- Registration and its applications
- Manual registration
  - Interpolation
  - Variational reconstruction ←
- Splines
- Automatic registration
  - Algorithm ←
  - Semi-automatic registration ←
  - Applications ←
- Conclusions
- Party

What did we do?

What is image registration?

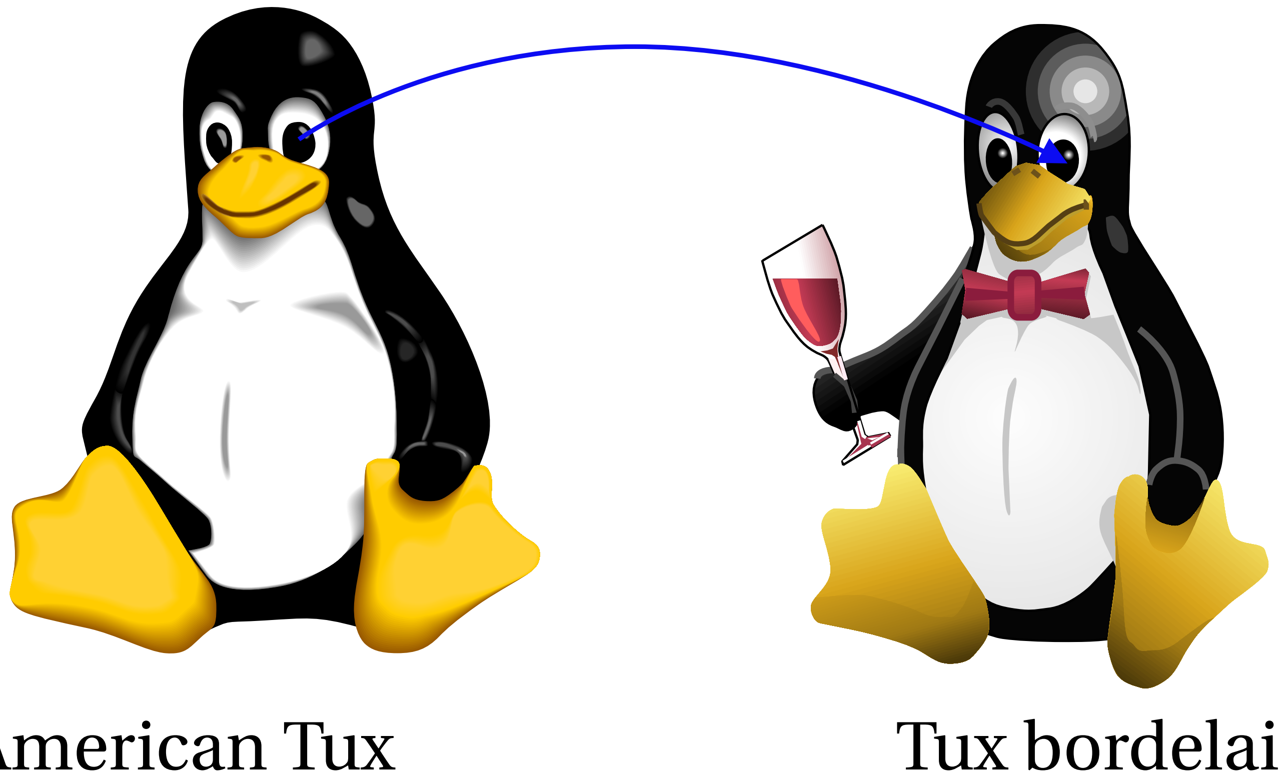
Find corresponding points



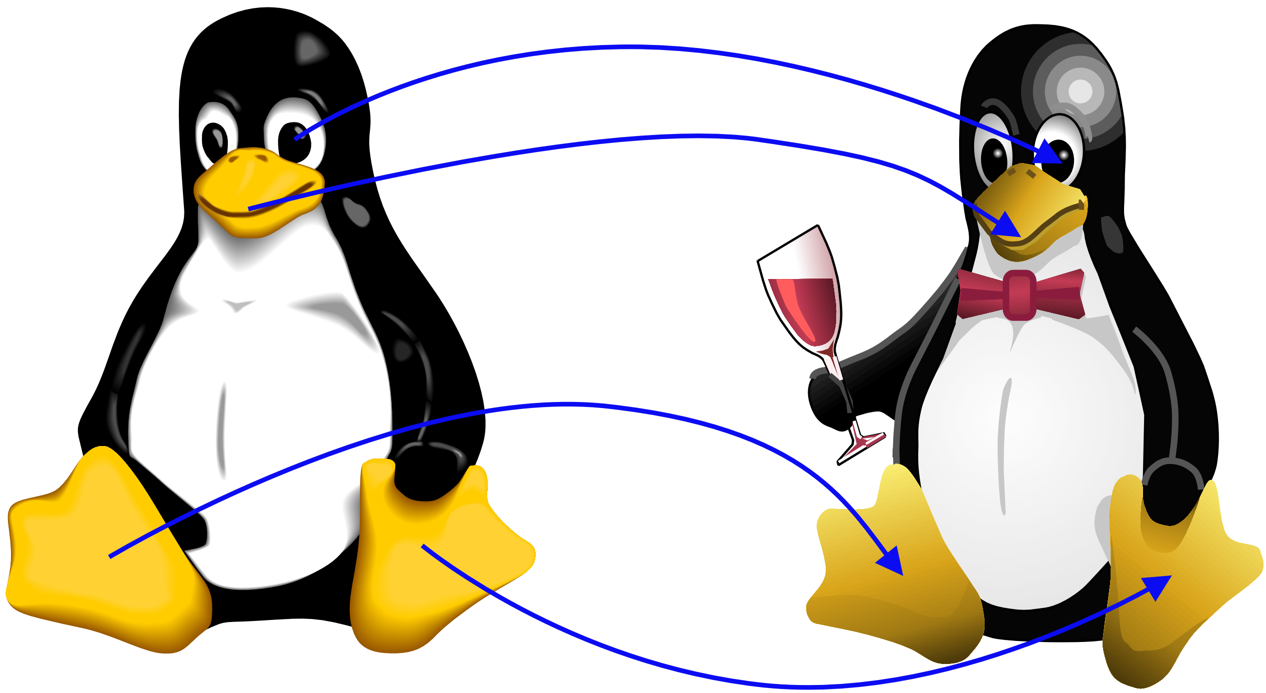
American Tux

Tux bordelais

Find corresponding points

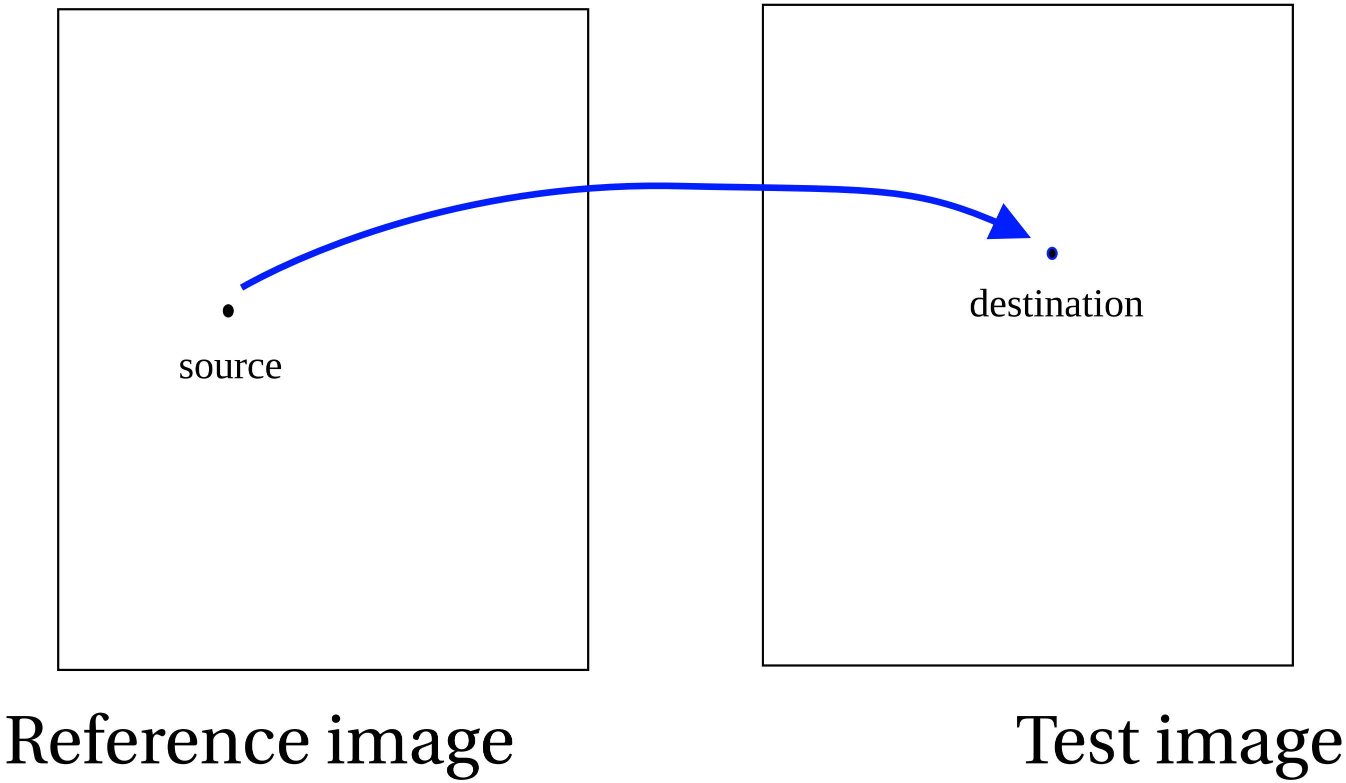


Find corresponding points

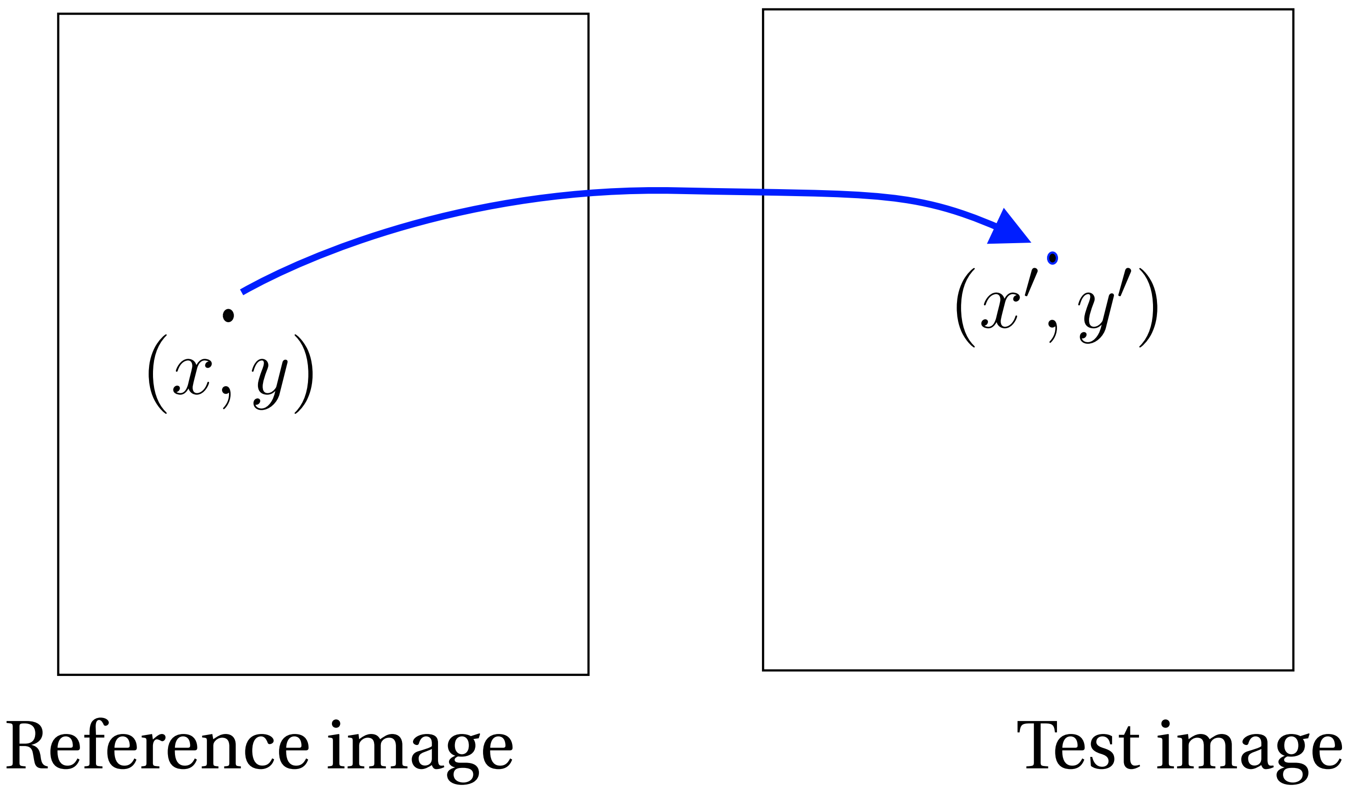


American Tux　　　　Tux bordelais

Correspondence function

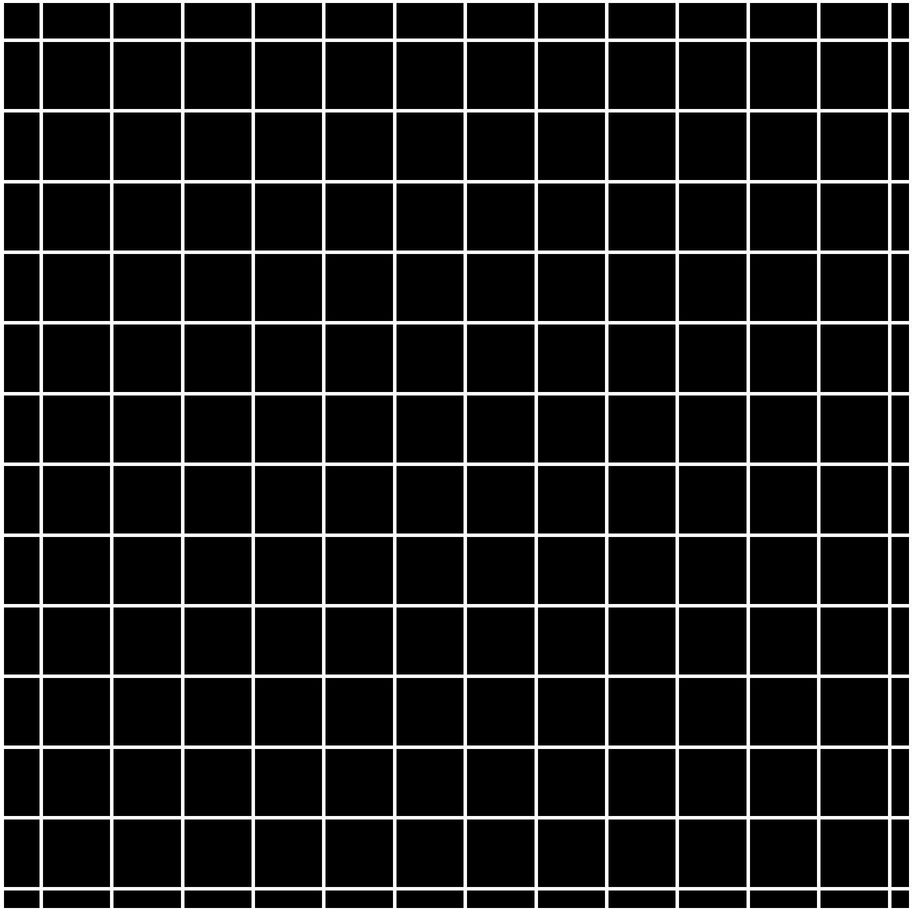


Correspondence function



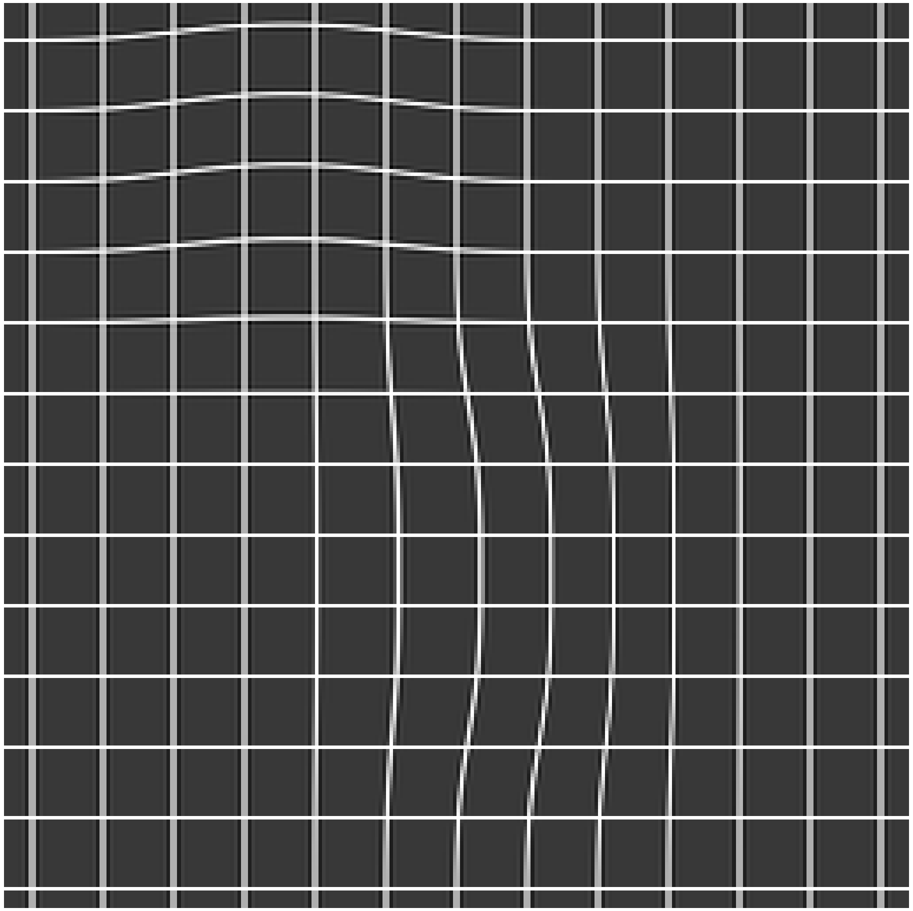
$$\mathbf{g}\left([x\ y]^T\right) = [x'\ y']^T$$

Deformation field



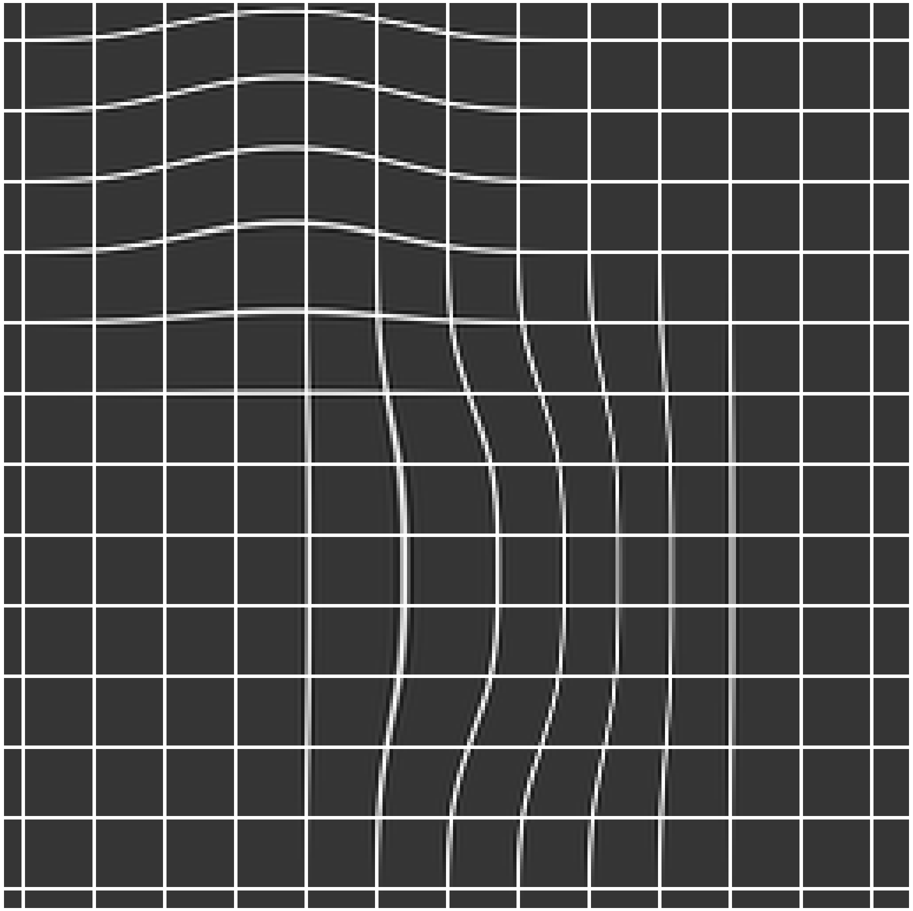
0 % deformation

Deformation field



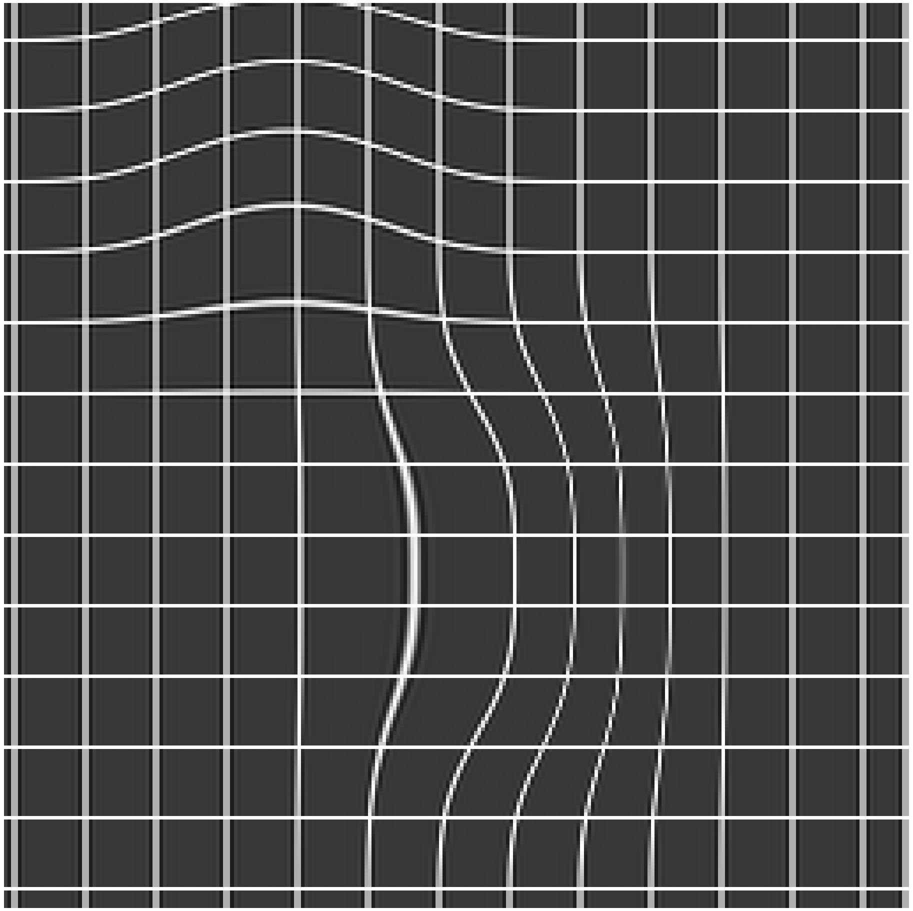
25 % deformation

Deformation field



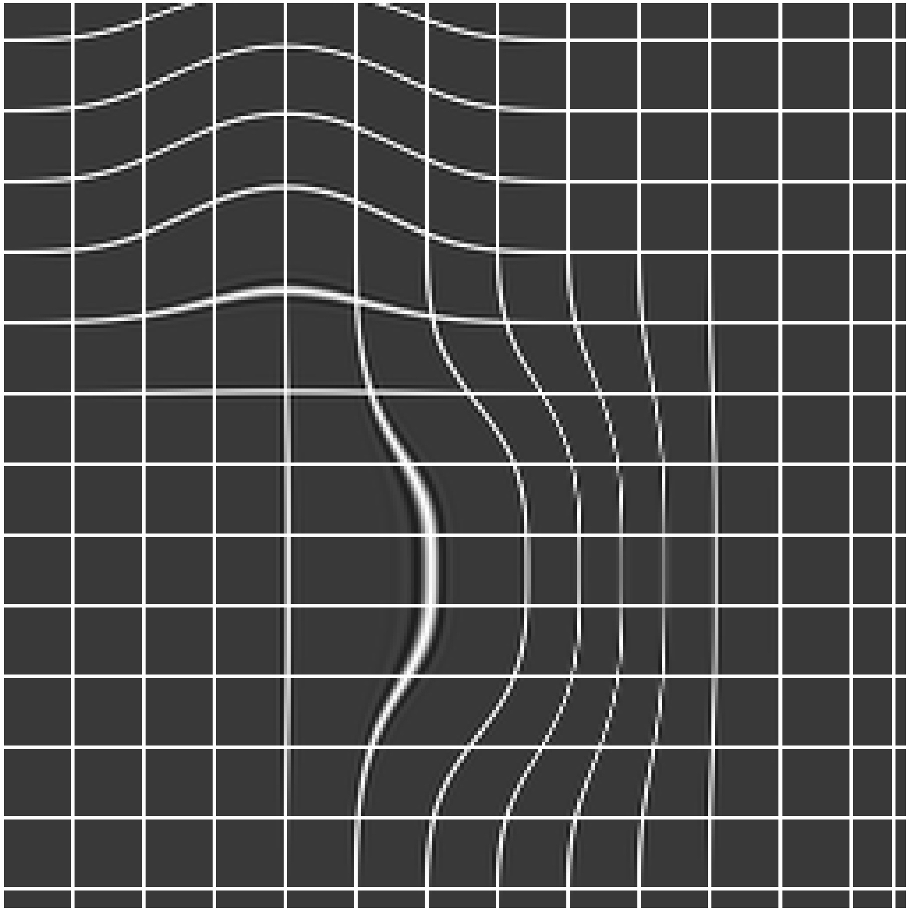
50 % deformation

Deformation field



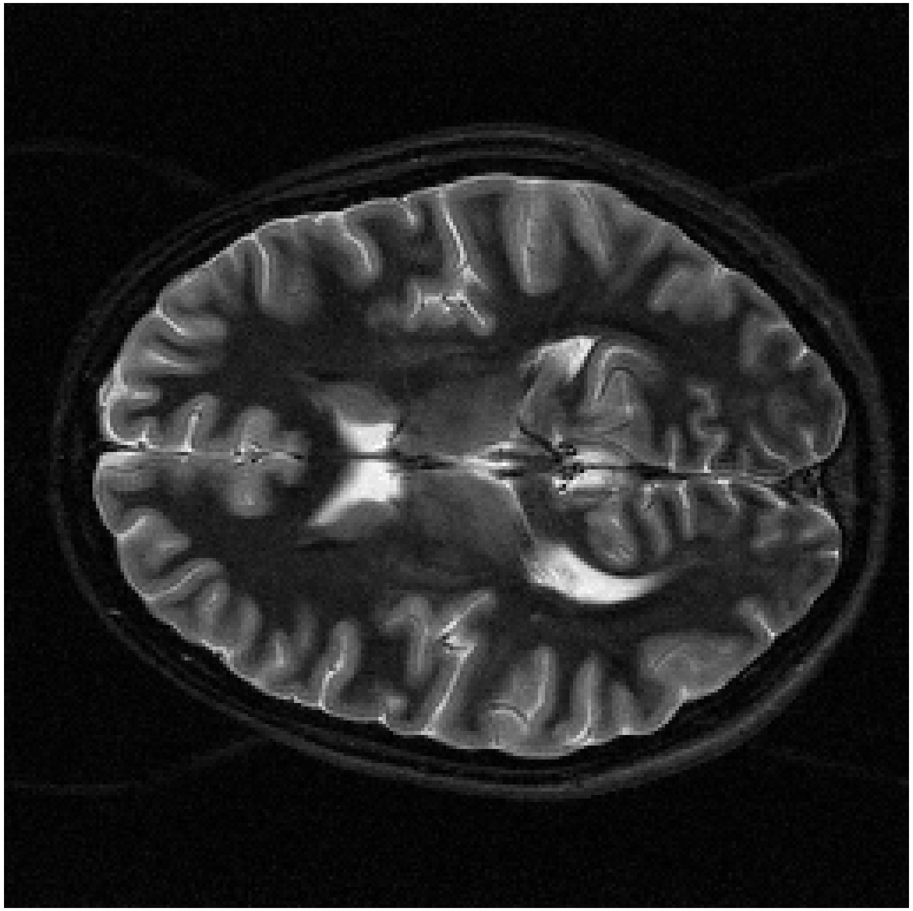
75 % deformation

Deformation field



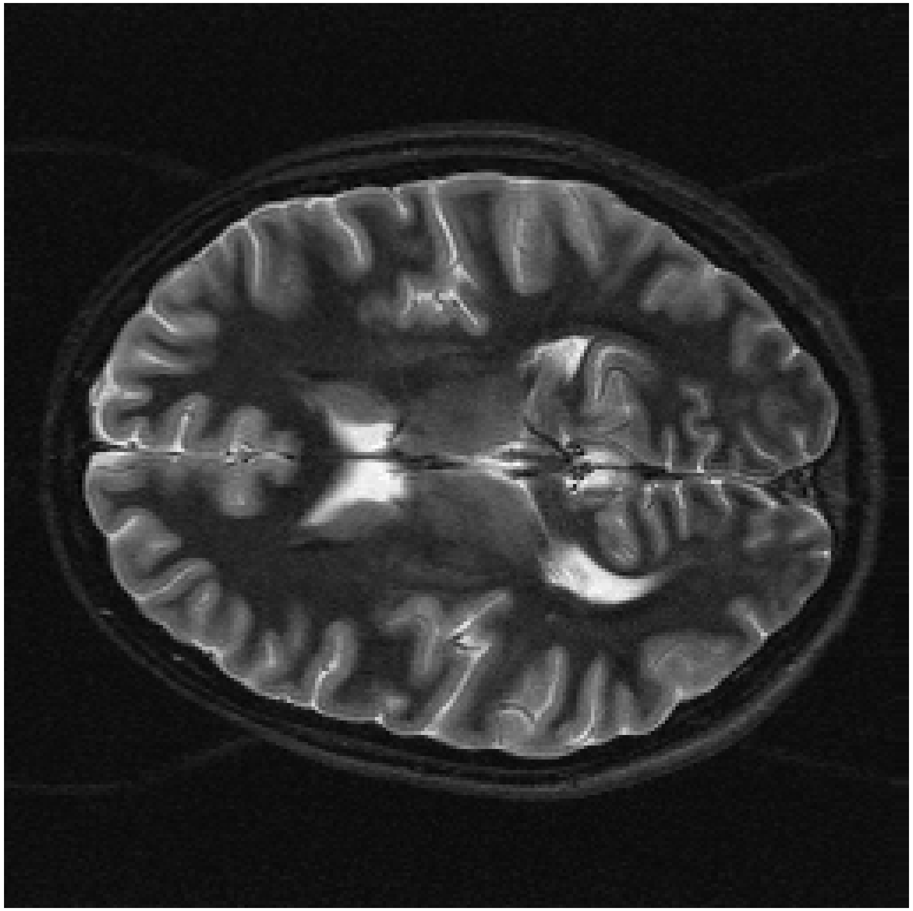
100 % deformation

Image warping



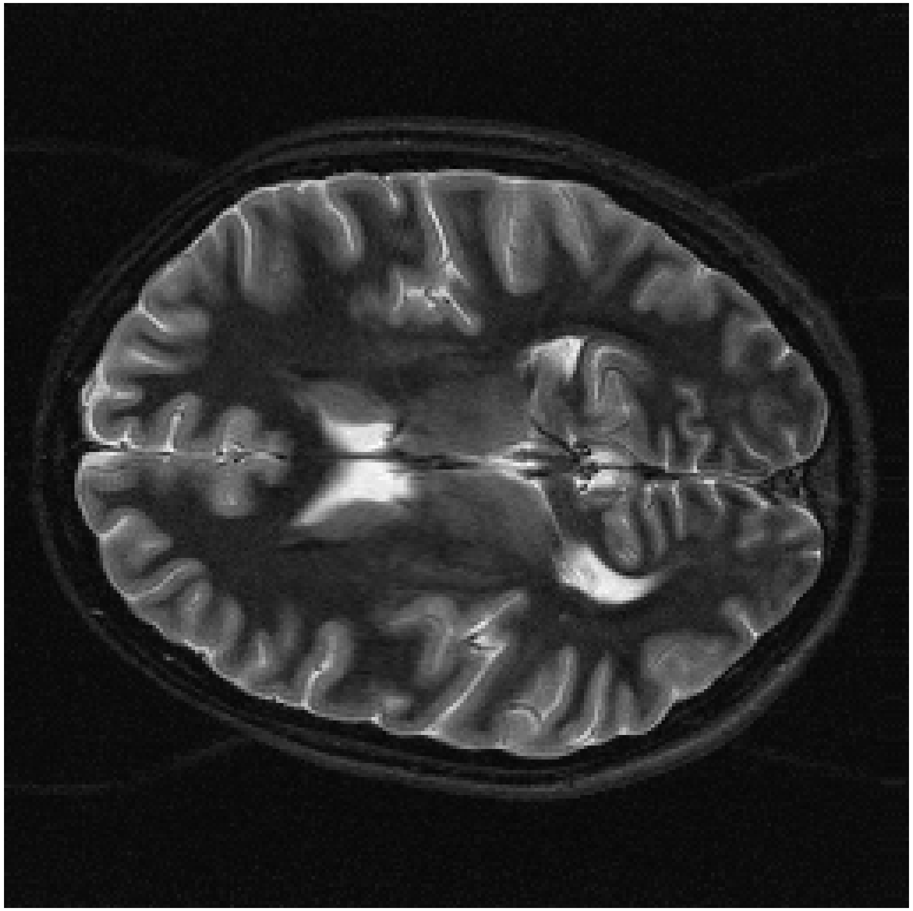
0 % deformation

Image warping



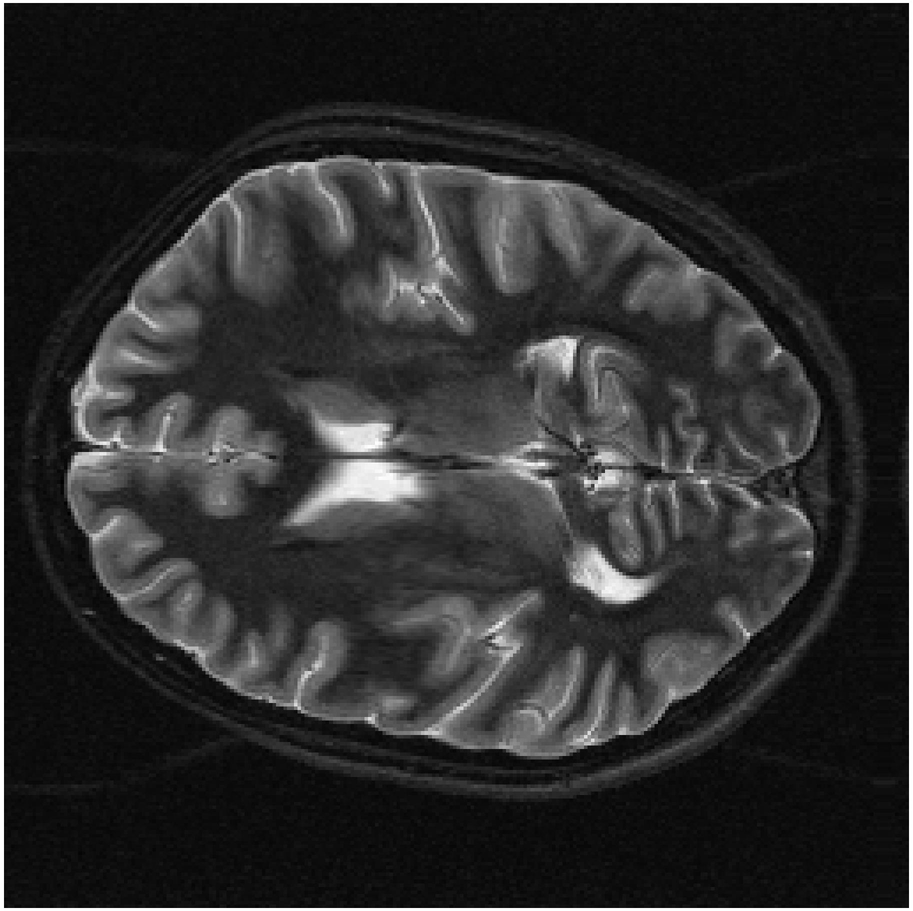
25 % deformation

Image warping



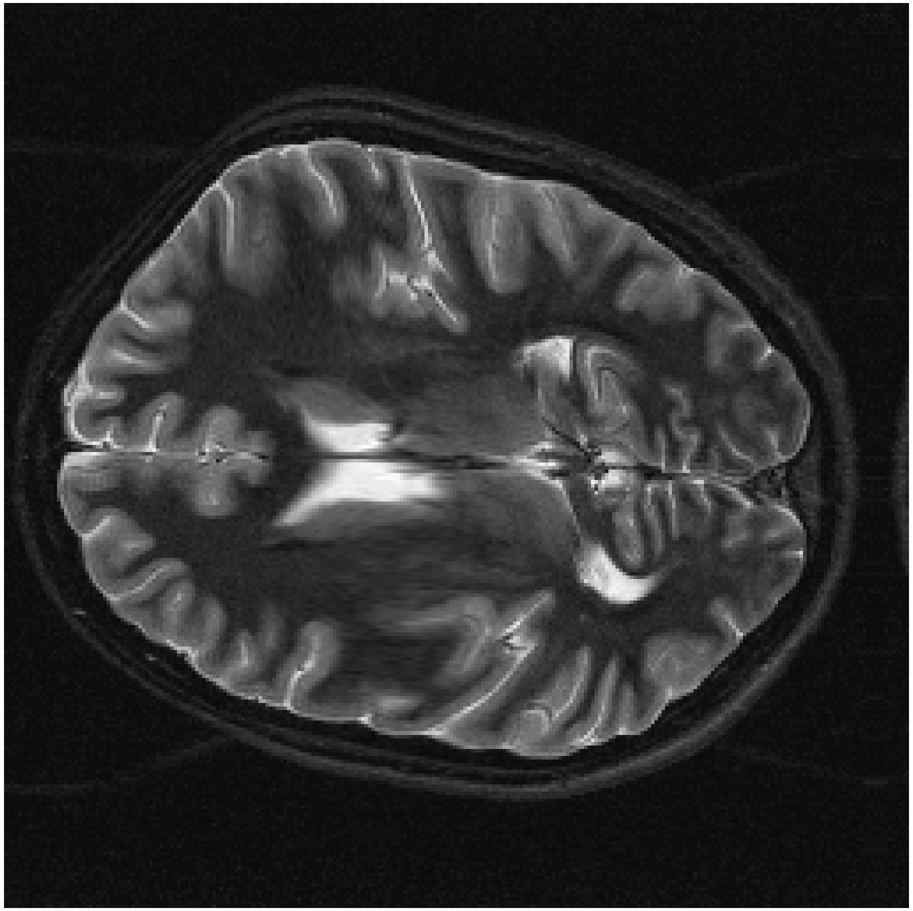
50 % deformation

Image warping



75 % deformation

Image warping



100 % deformation

Image registration

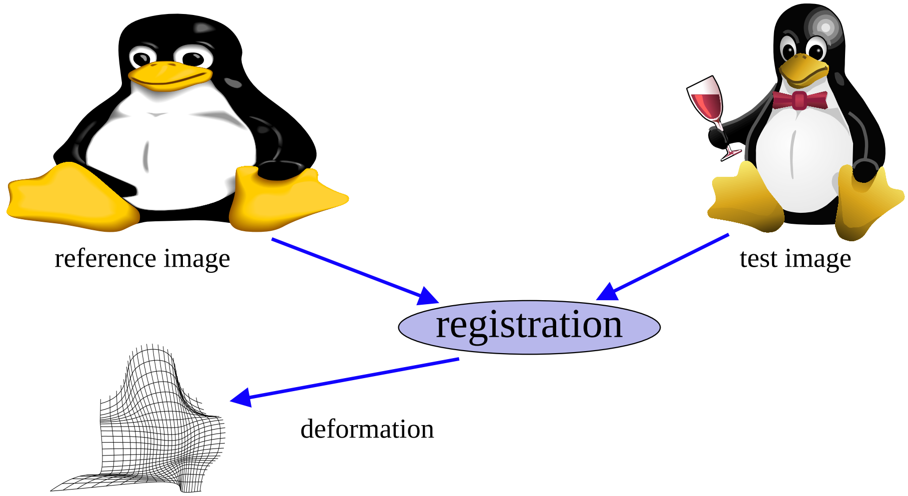
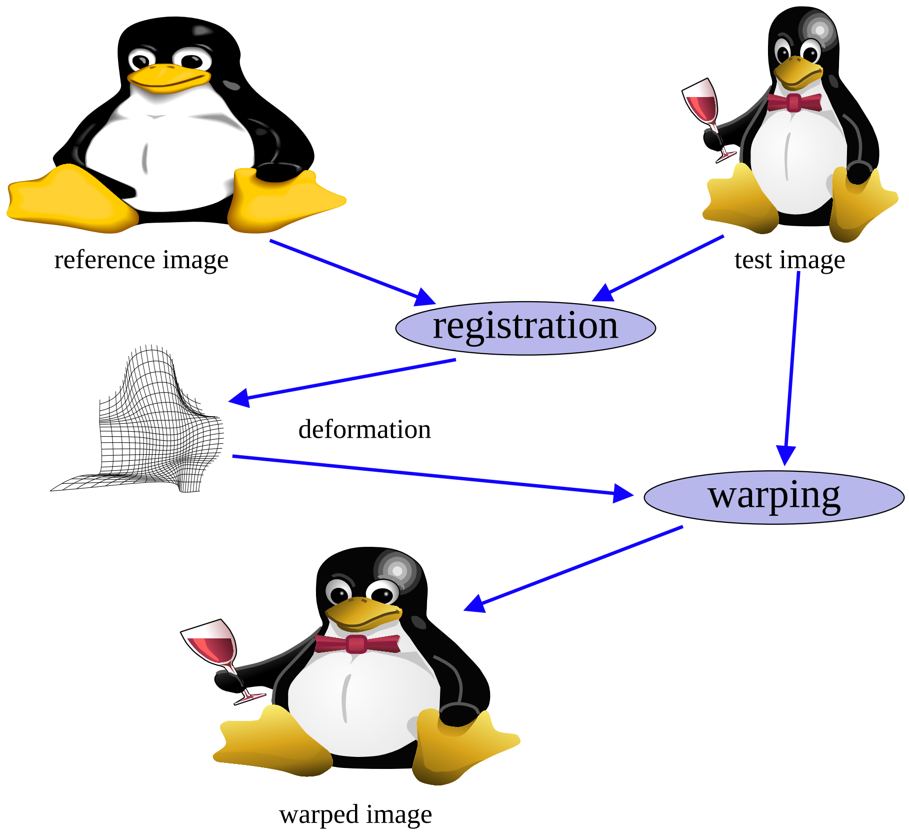


Image registration



Is registration useful?

Is registration useful?

Yes!

(Biomedical) applications

. . . of image registration

- Comparing images
  - Different times
  - Different methods
  - Different subjects
- Analyzing sequences
  - Motion estimation
  - Segmentation

Qualitative and quantitative information.

Image alignment

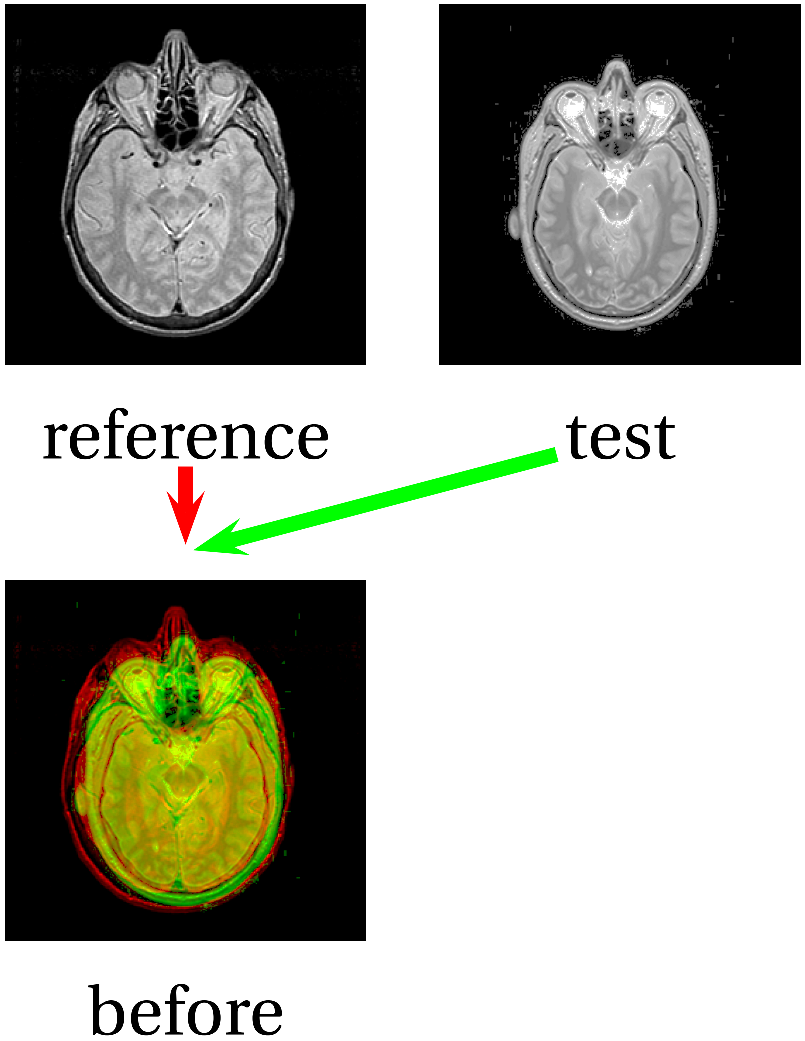
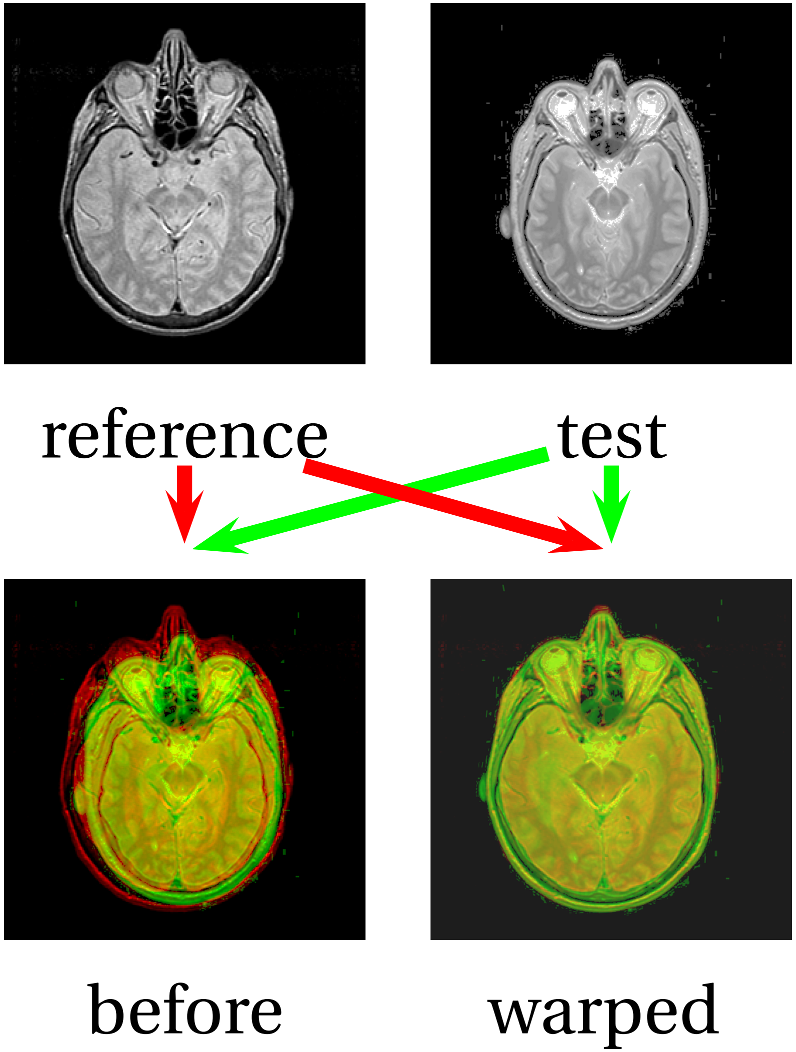


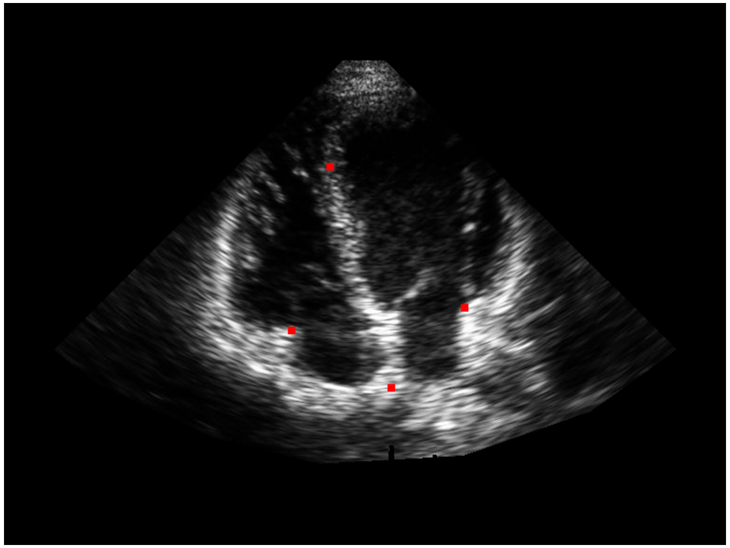
Image alignment

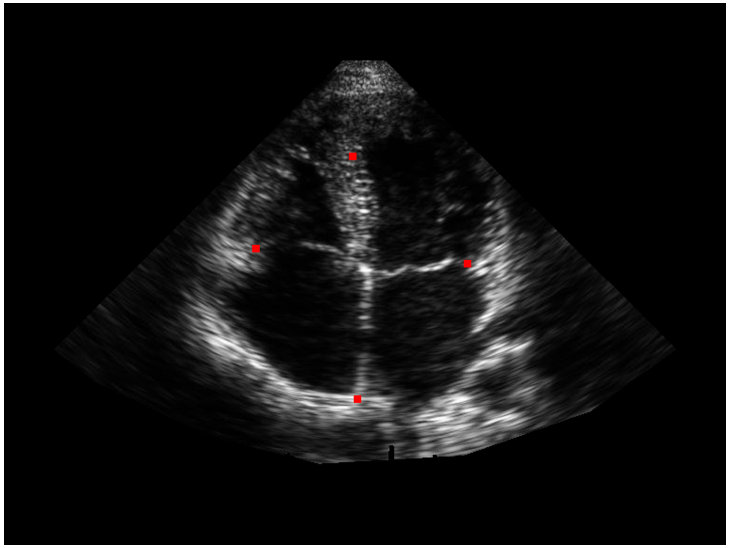


Registration types

- Manual
- Automatic
- Semi-automatic

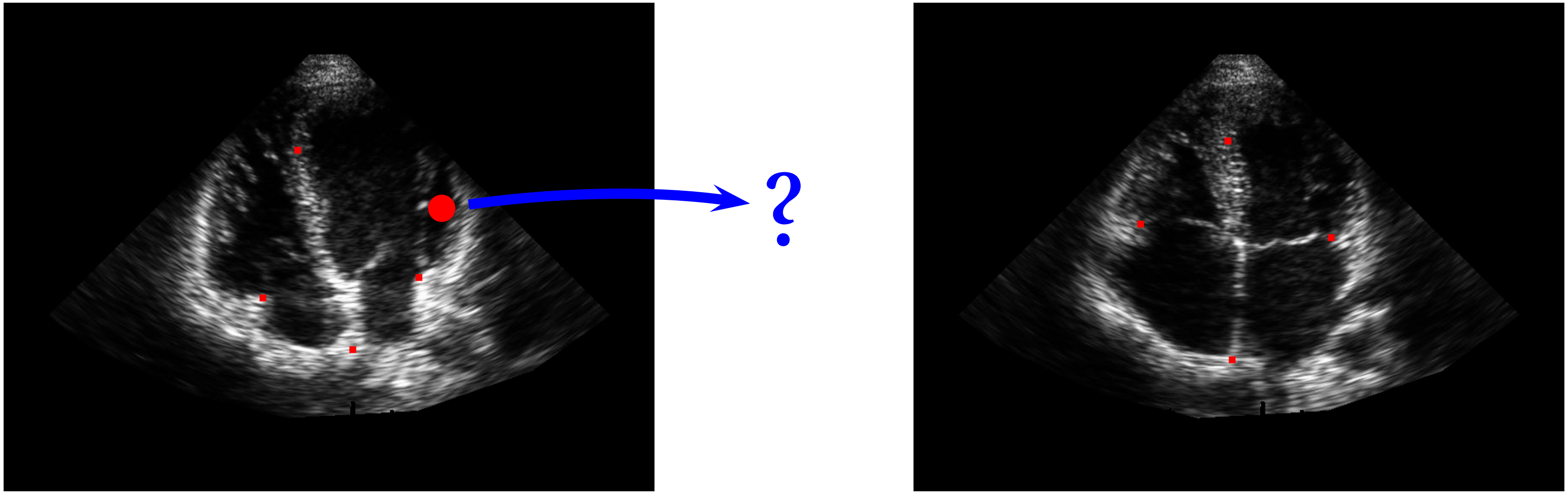
Manual registration





- Landmark identification

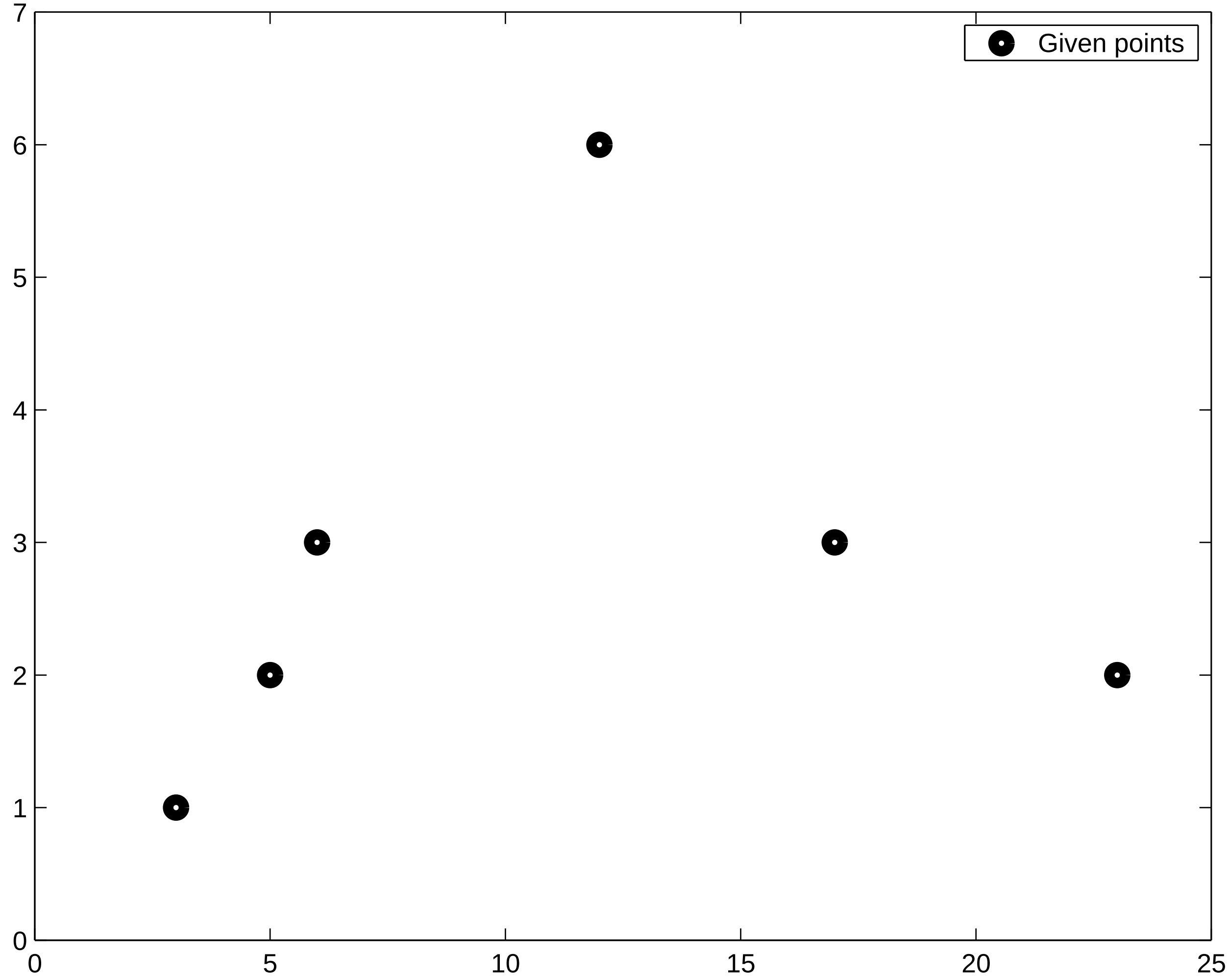
Manual registration



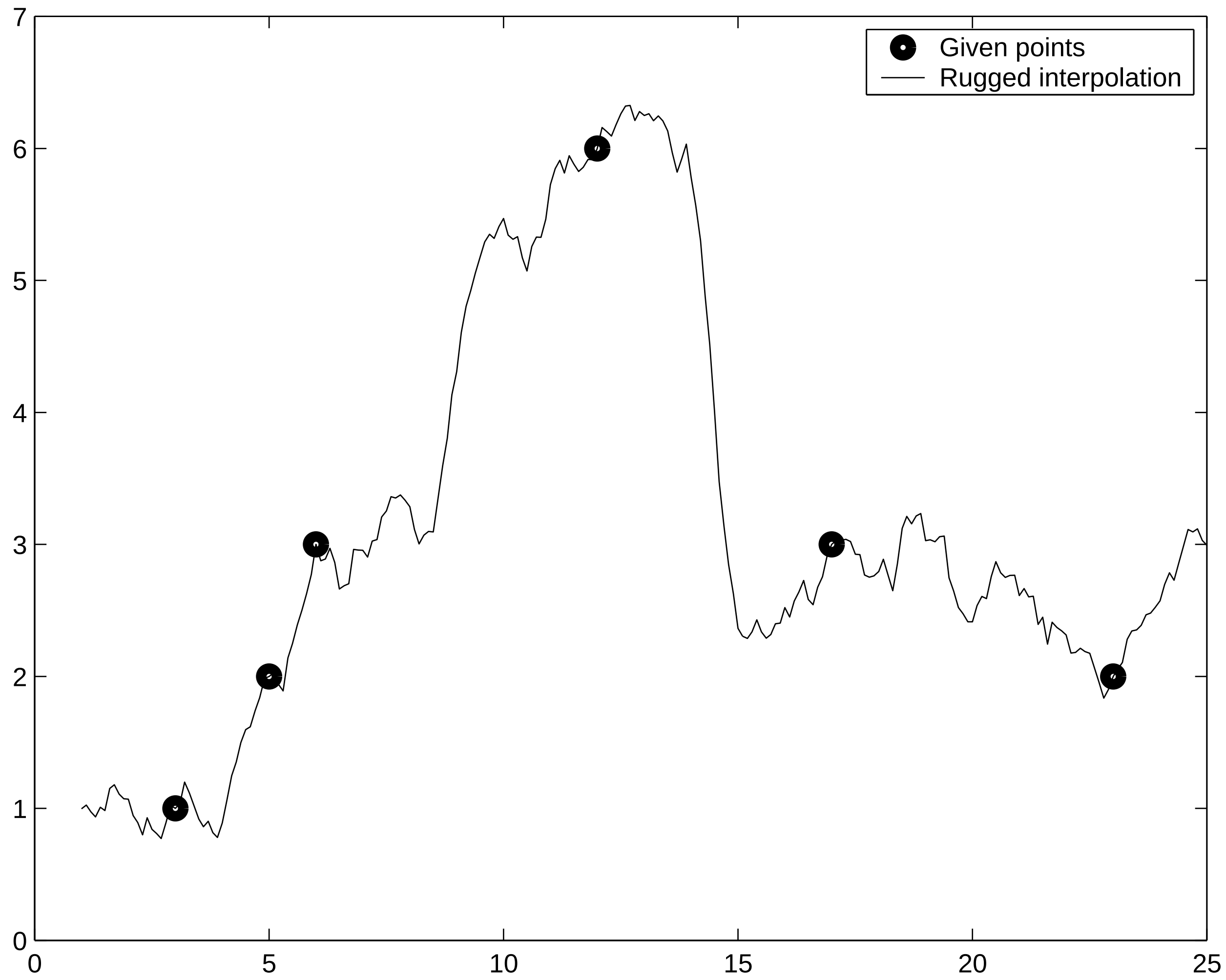
- Landmark identification
- Landmark interpolation

What is interpolation?

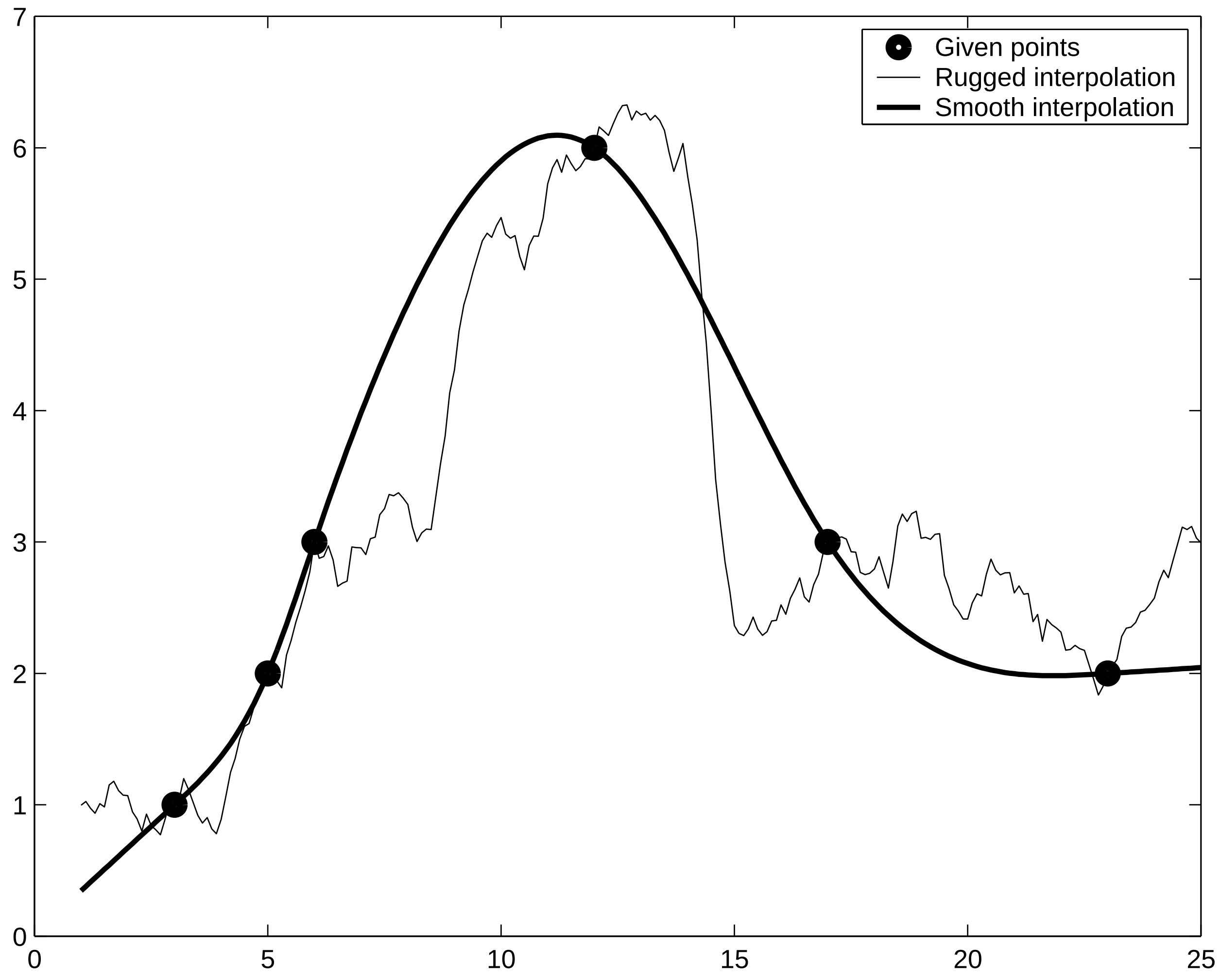
Find a function



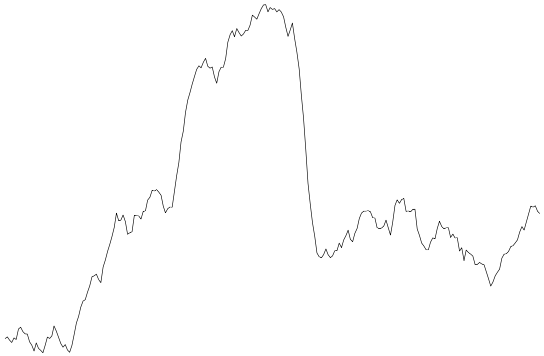
Find a function

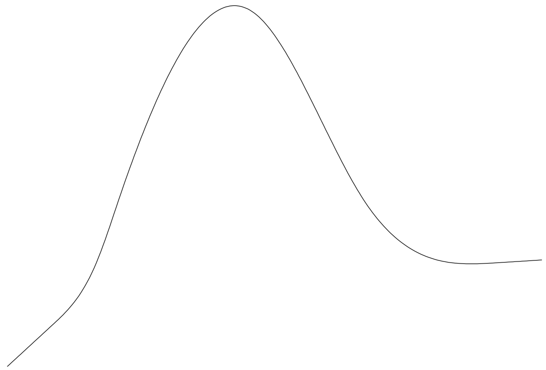


Find a function



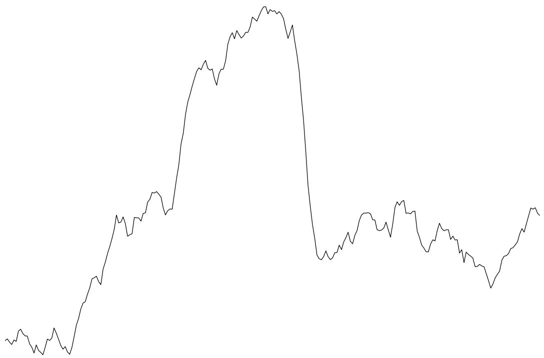
Rank functions

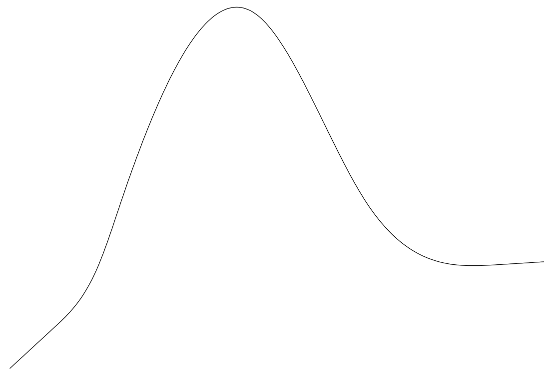




ugliest . . . prettiest

Rank functions





ugliest . . . prettiest

Variational criterion

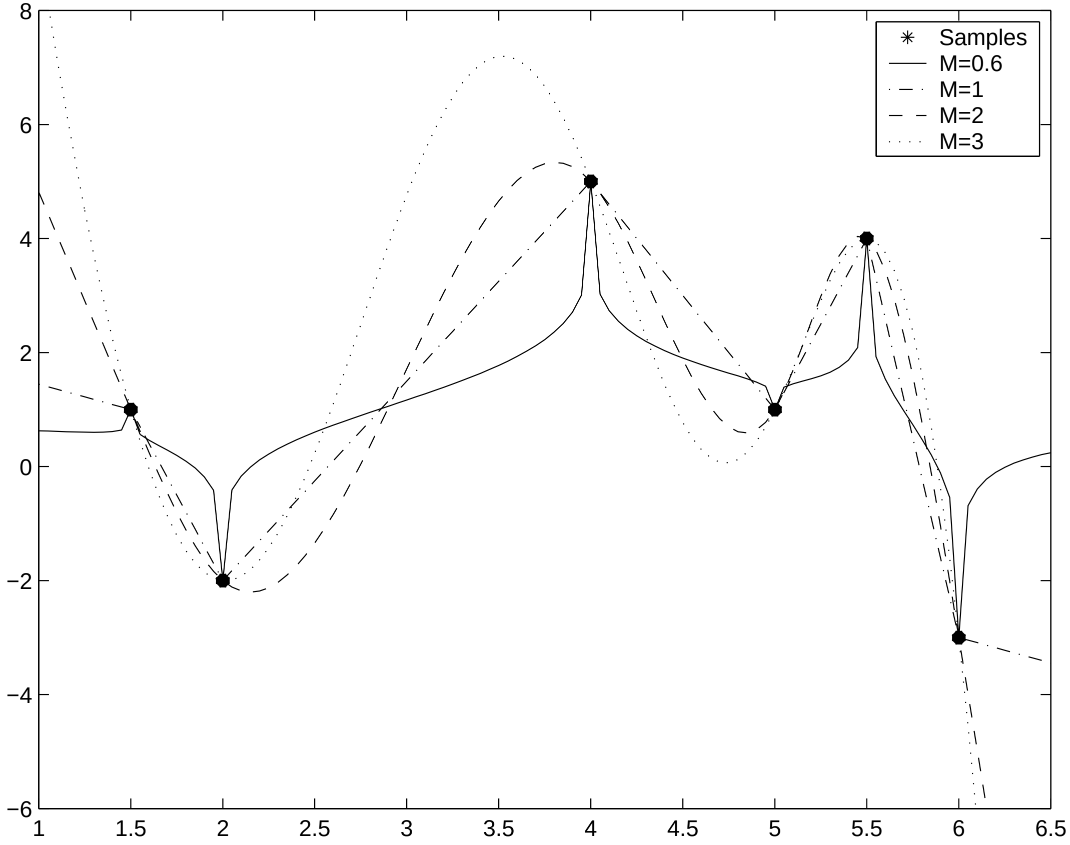
$$J : \left(\mathbb{R}^m \longrightarrow \mathbb{R}^n\right) \longrightarrow \mathbb{R}_0^+$$

$$J(\mathbf{f}) \geq 0$$

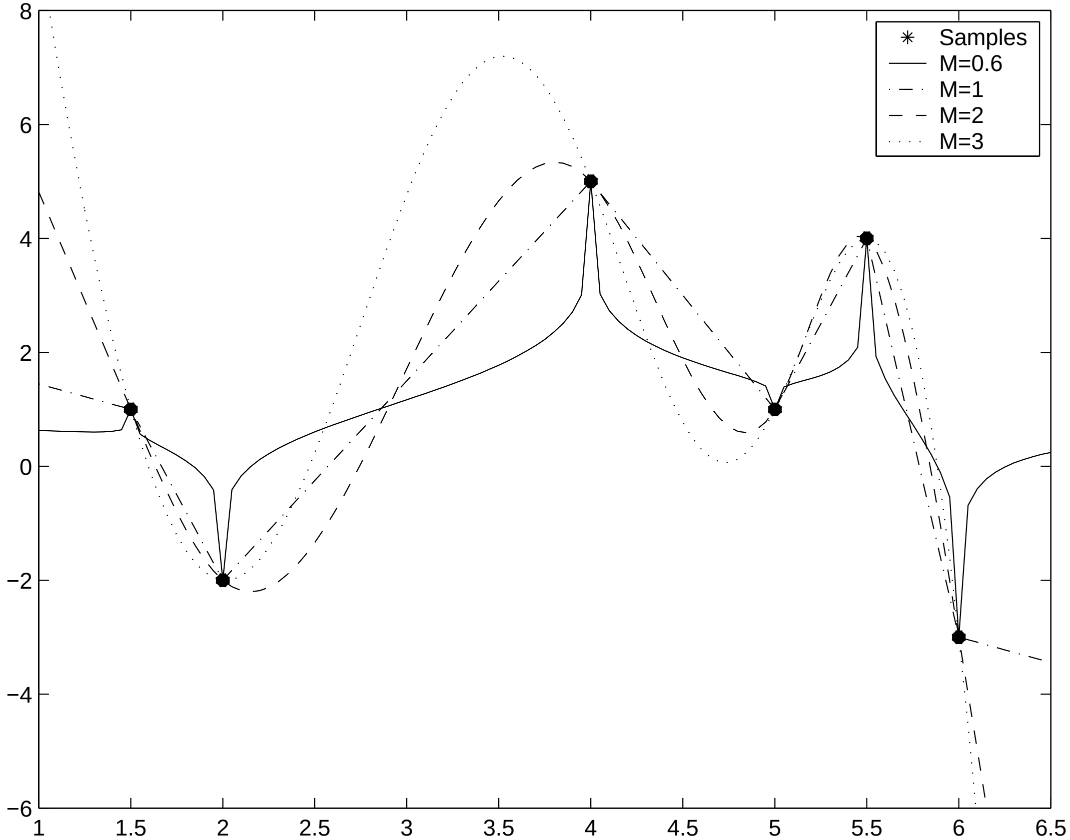
Variational reconstruction

Find the best function
satisfying the constraints.

Tunable 1D interpolation

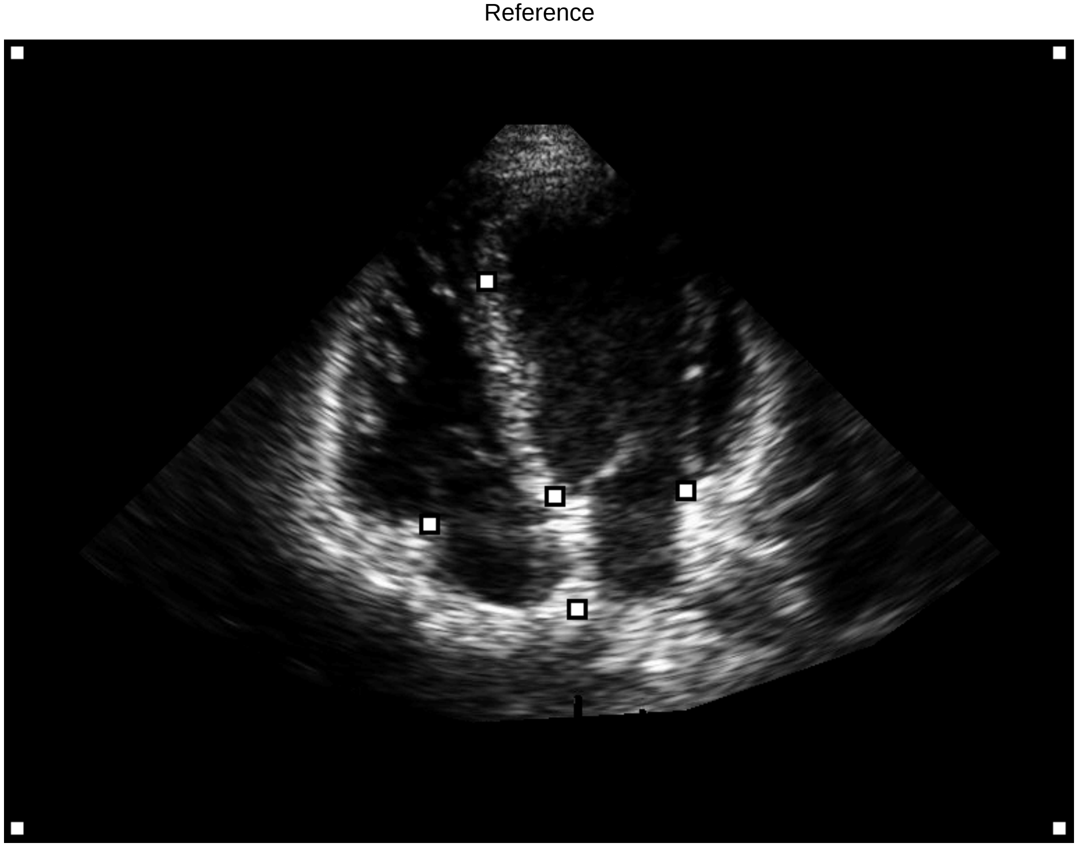


Tunable 1D interpolation



$$J(f) = \left\| \frac{\partial^M f}{\partial x^M} \right\|^2$$

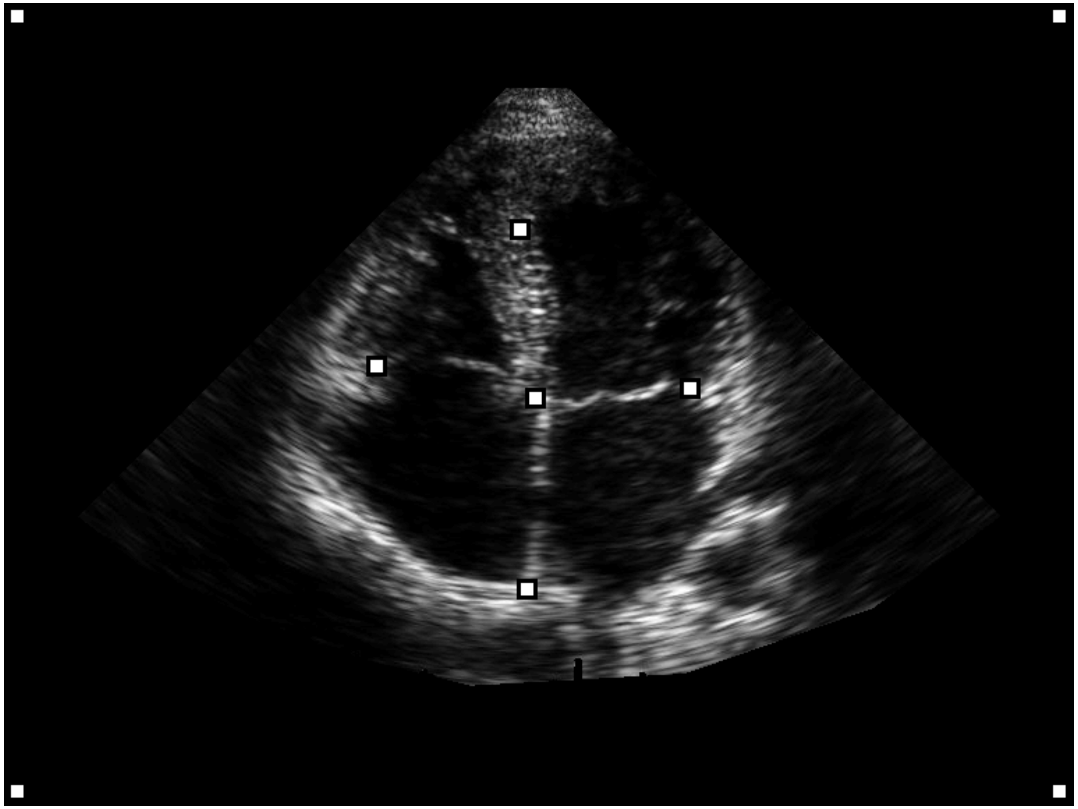
Tunable 2D interpolation



reference

Tunable 2D interpolation

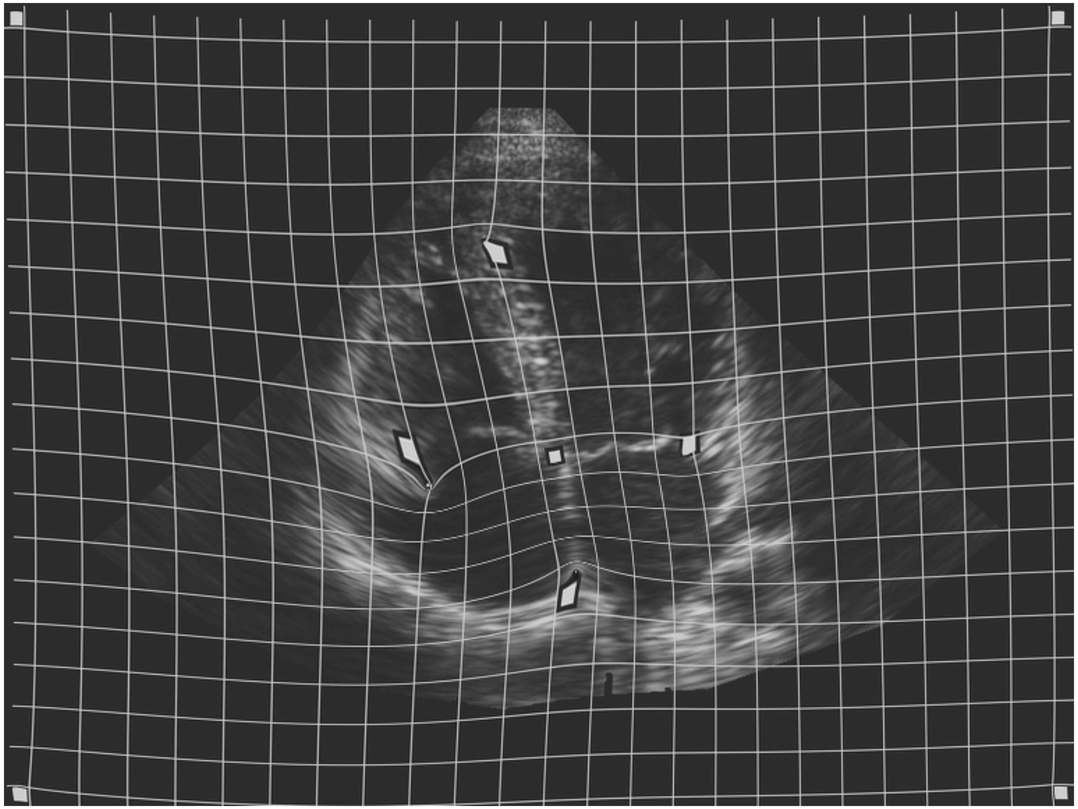
Test



test

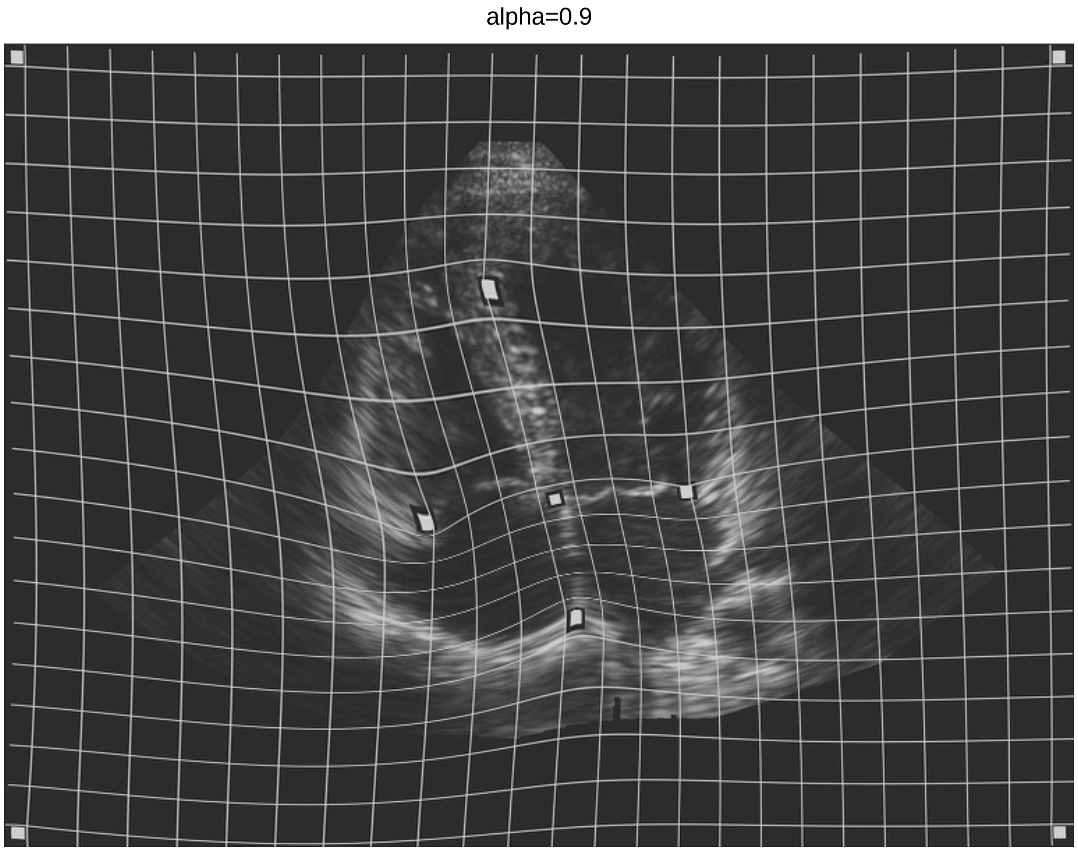
Tunable 2D interpolation

alpha=0.5



$$\int \|\nabla^{0.5} g(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}$$

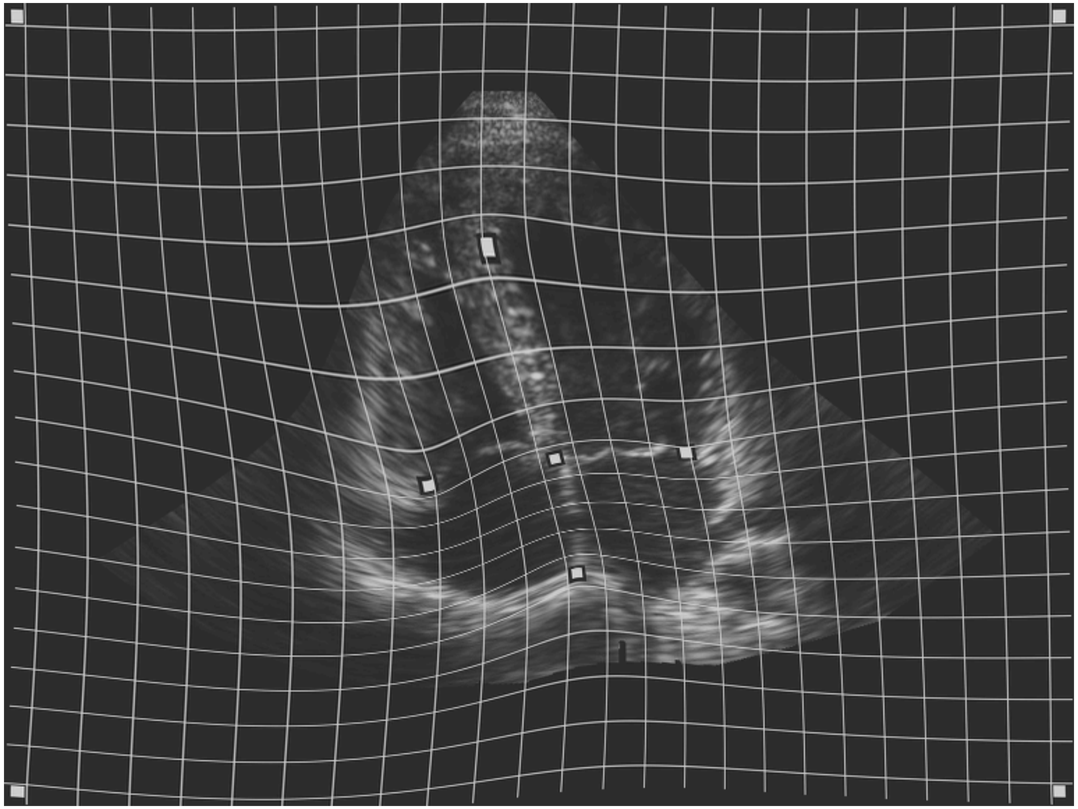
Tunable 2D interpolation



$$\int \|\nabla^{0.9} g(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}$$

Tunable 2D interpolation

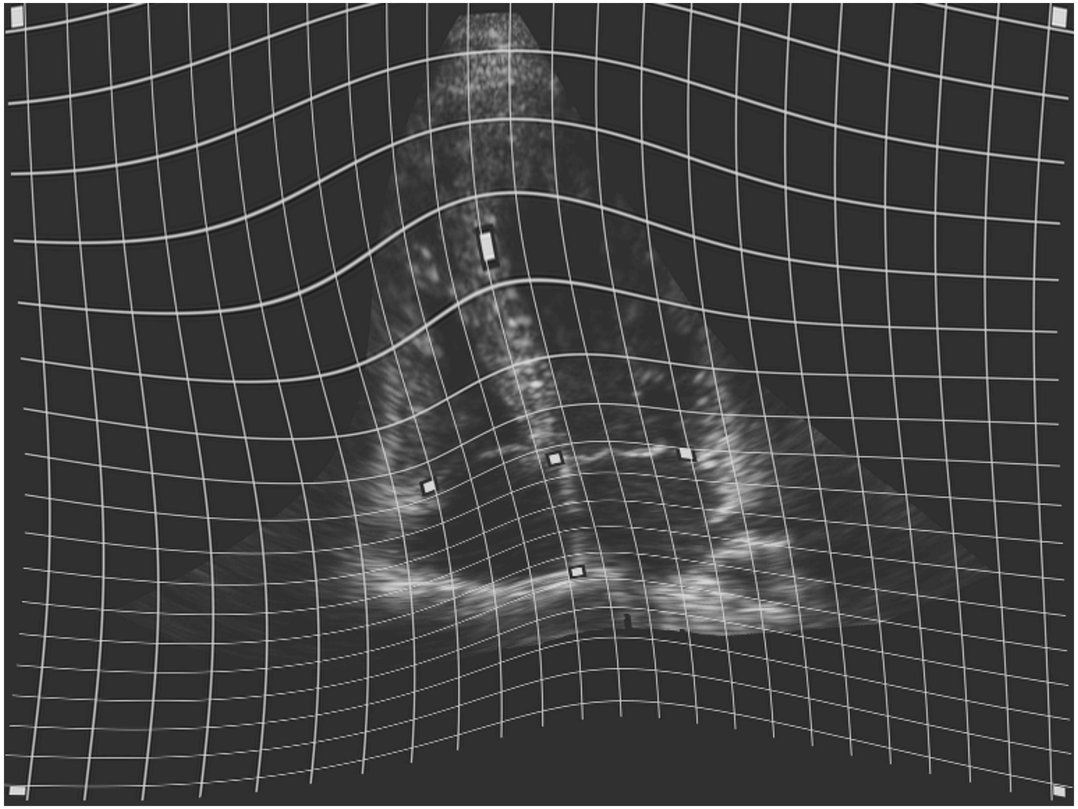
alpha=1.3



$$\int \|\nabla^{1.3} g(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}$$

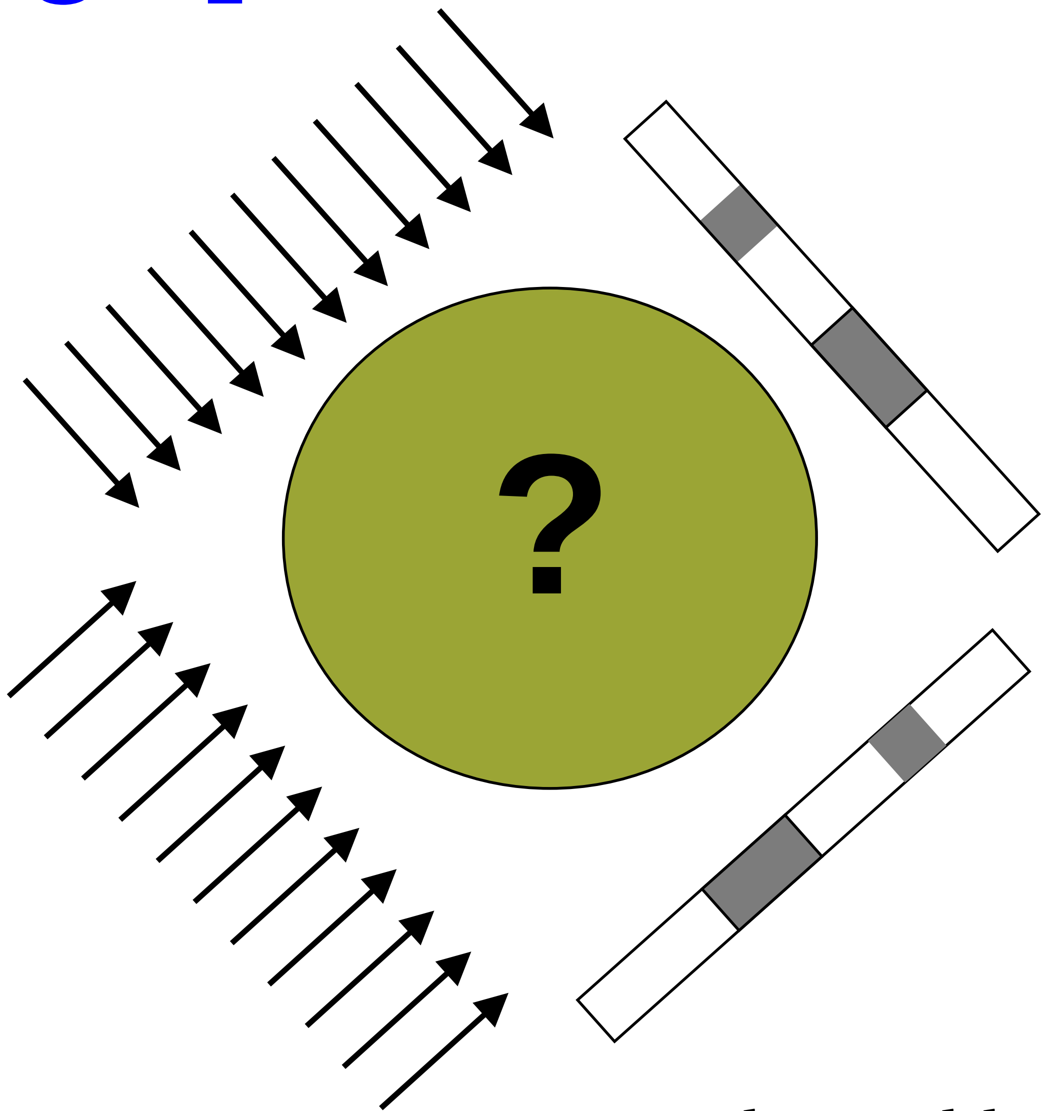
Tunable 2D interpolation

alpha=2.5



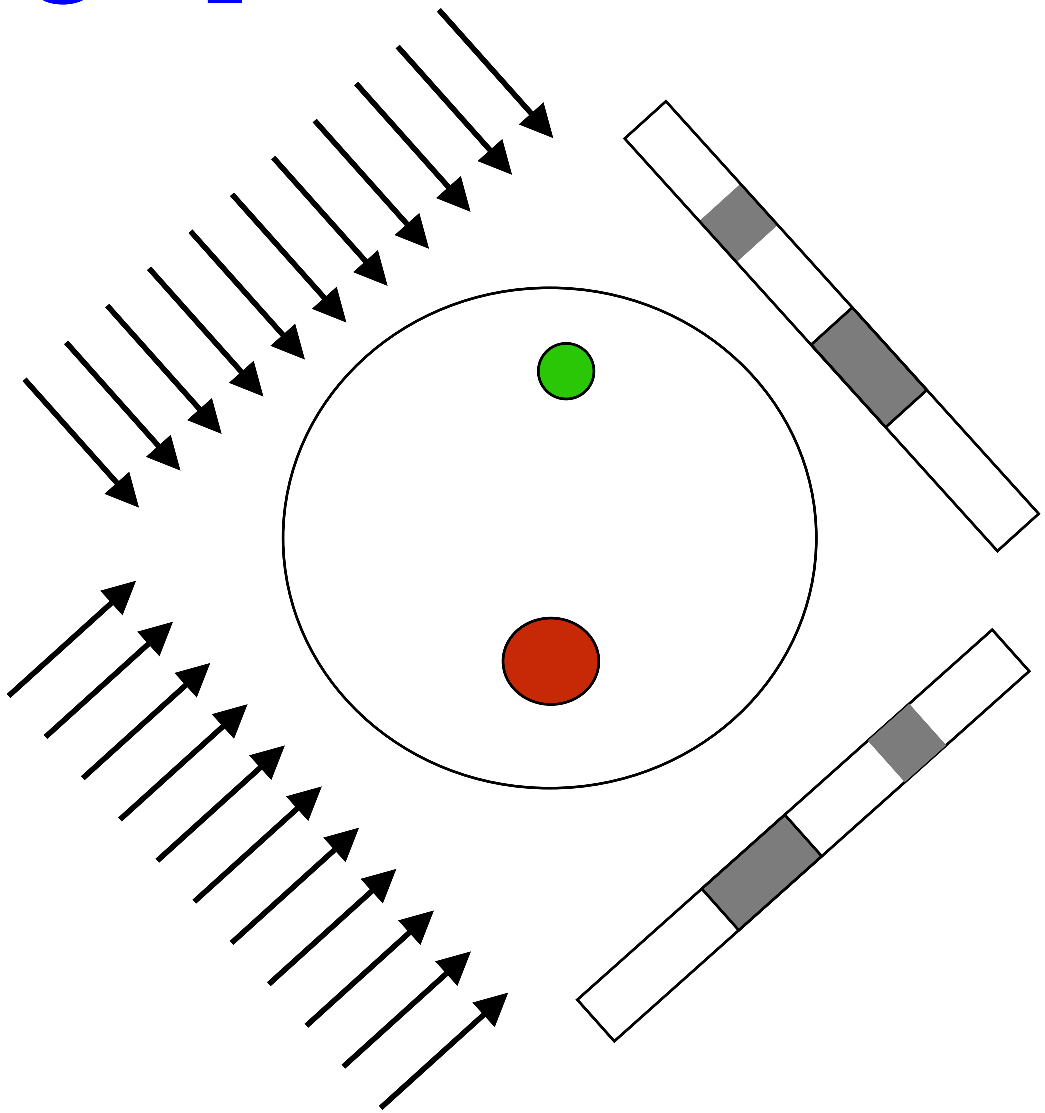
$$\int \|\nabla^{2.5} g(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}$$

Tomographic reconstruction



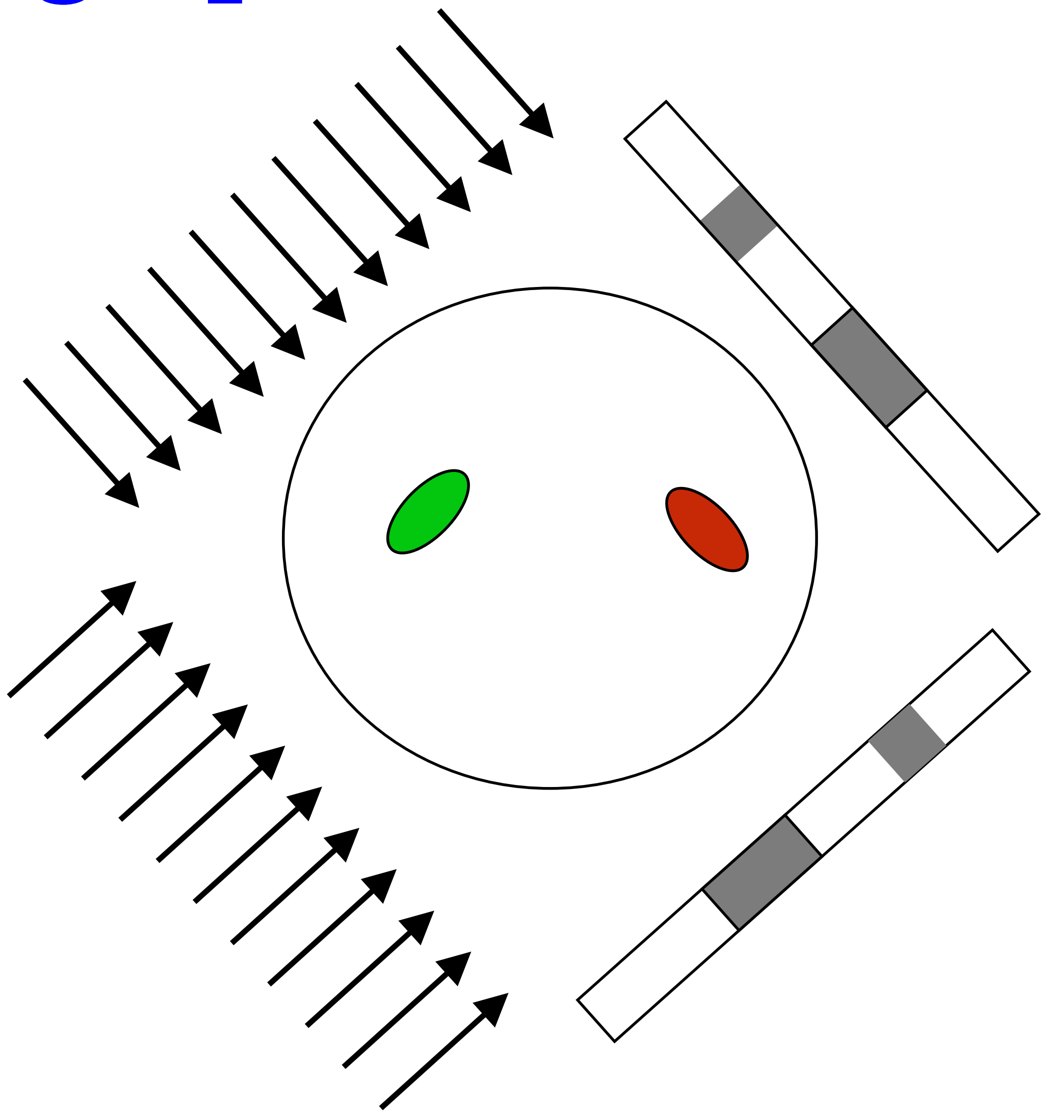
The problem

Tomographic reconstruction



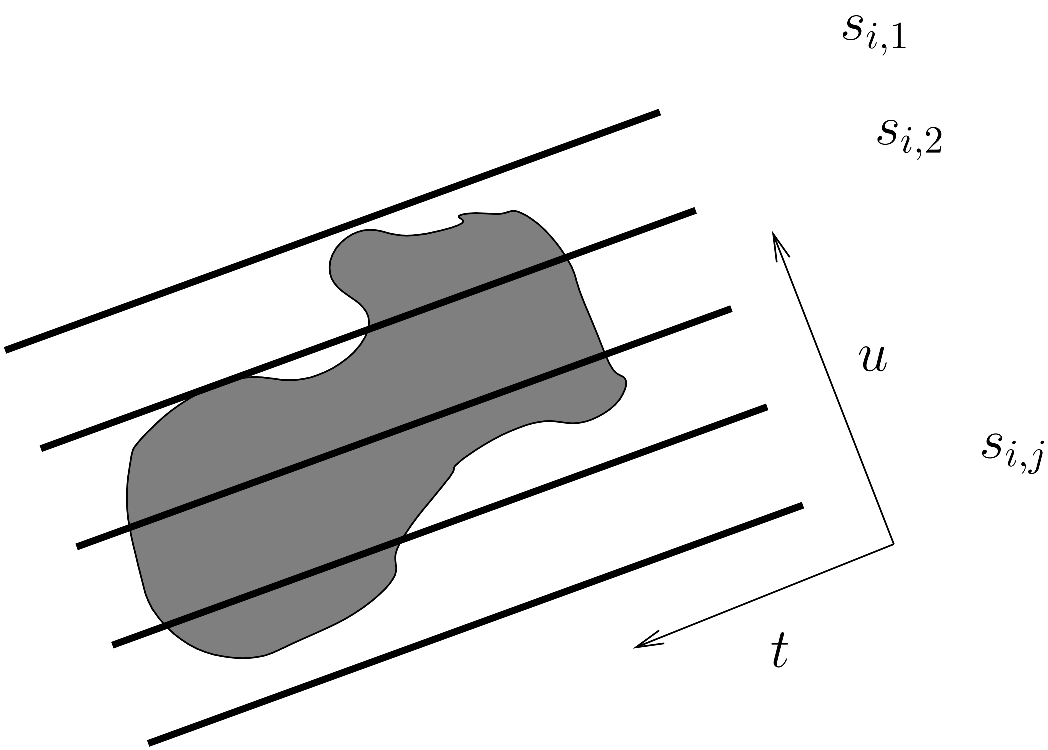
Solution

Tomographic reconstruction



Another solution

Radon transform



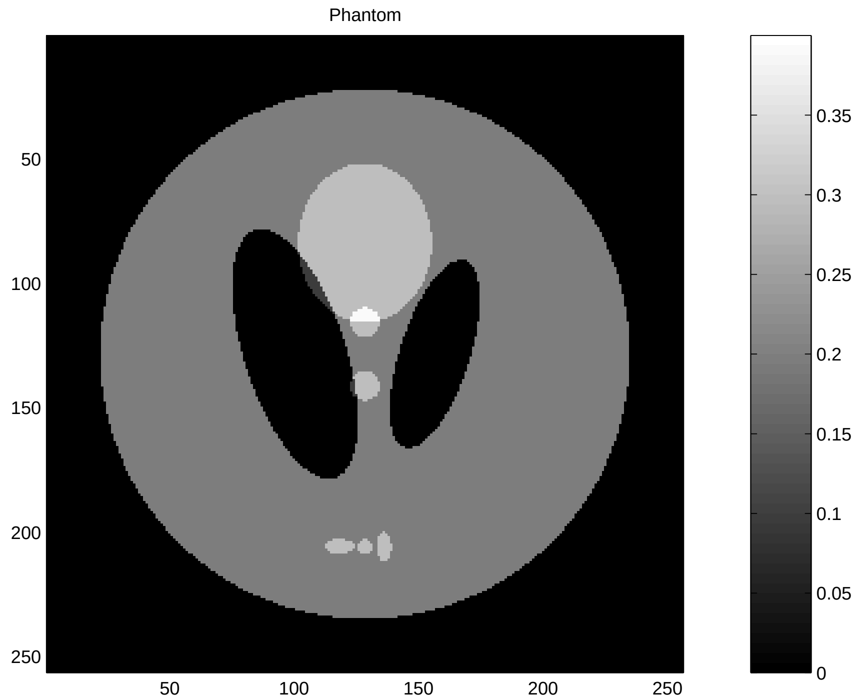
angle θ_i

Radon transform

$$(\mathscr{R} f)(\theta_i, u_j) = \int f\big(t\cos\theta_i - u_j \sin\theta_i, t\sin\theta_i + u_j\cos\theta_i\big)\ \mathrm{d}t$$

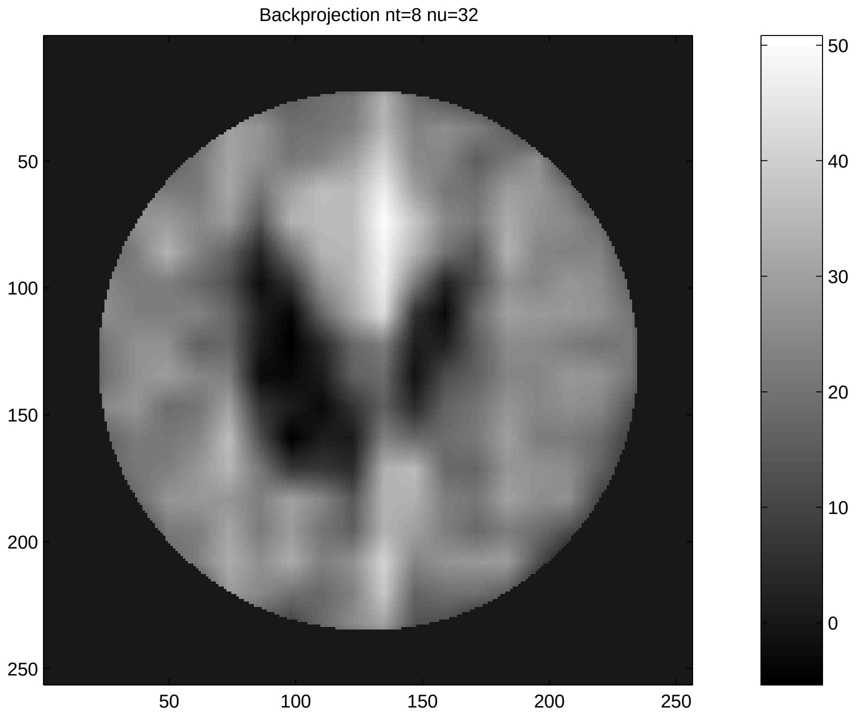
Tomographic experiments

Phantom

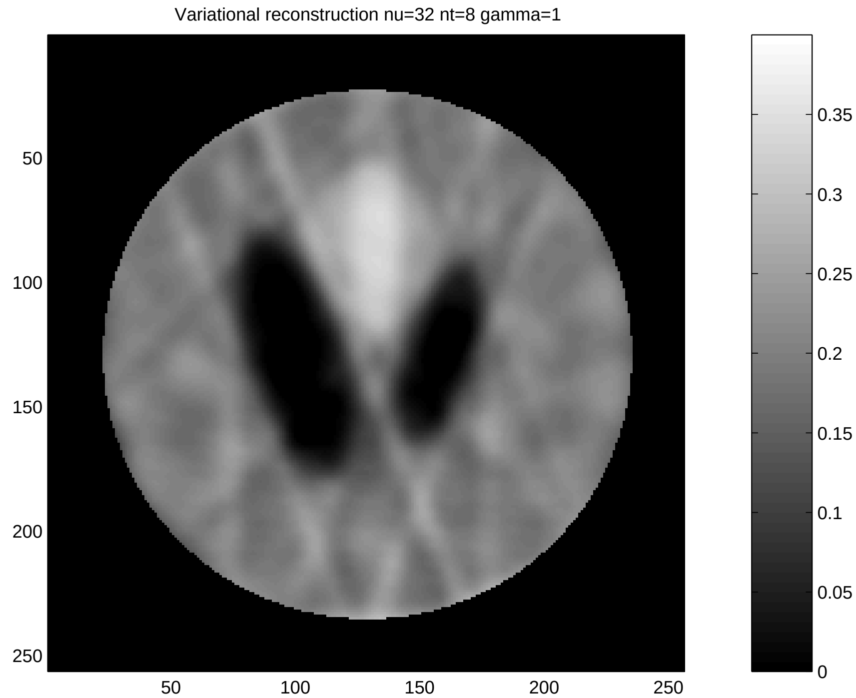


Tomographic experiments

Backprojection



Variational reconstruction



8 angles (each 22°), 32 samples per angle

Overview (2)

- Registration and its applications
- Manual registration
  - Interpolation
  - Variational reconstruction
- Splines
- Automatic registration
  - Algorithm
  - Semi-automatic registration
  - Applications
- Conclusions
- Party

The splines

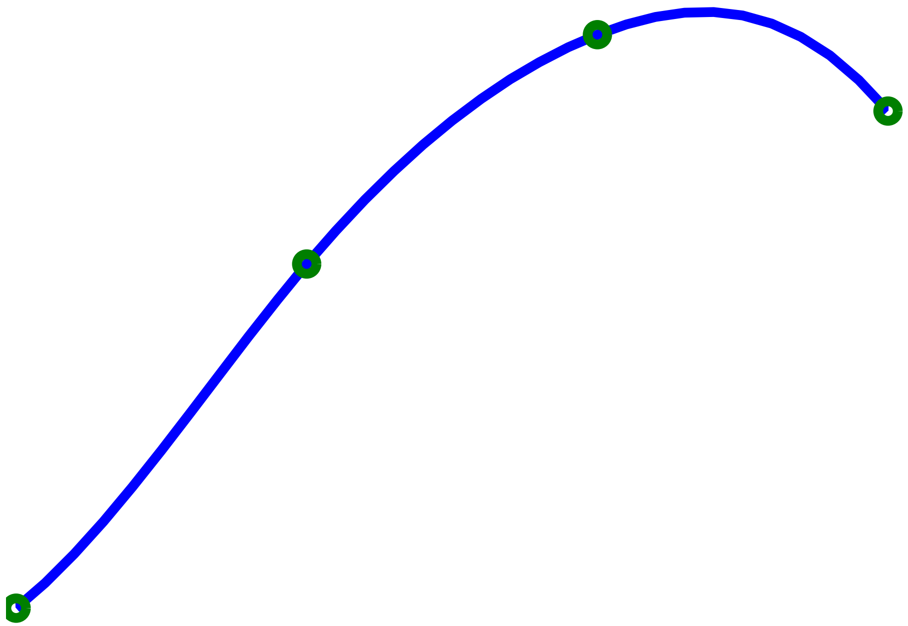
What are splines, anyway?

The splines

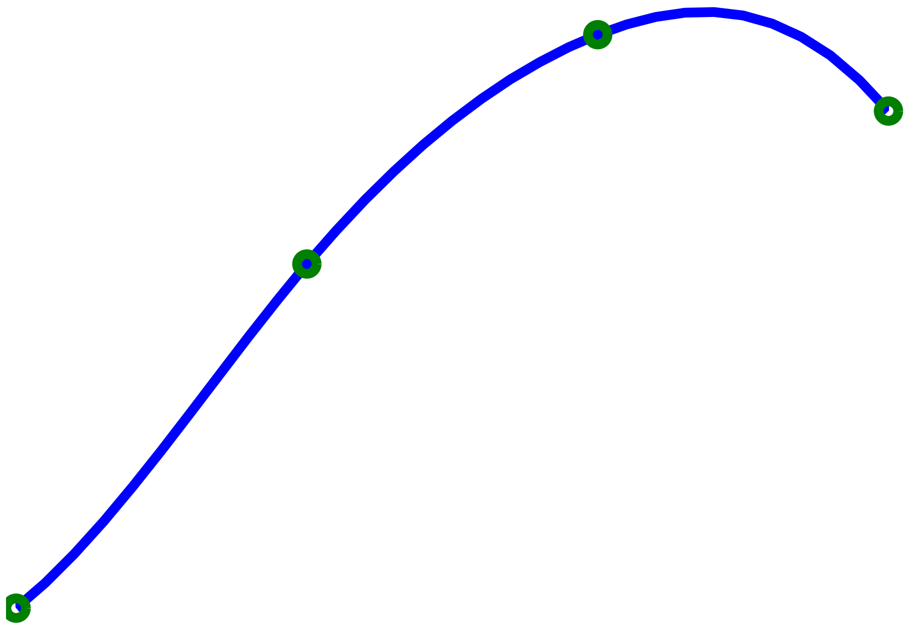
What are splines, anyway?

The best functions in the world!

(Uniform) splines



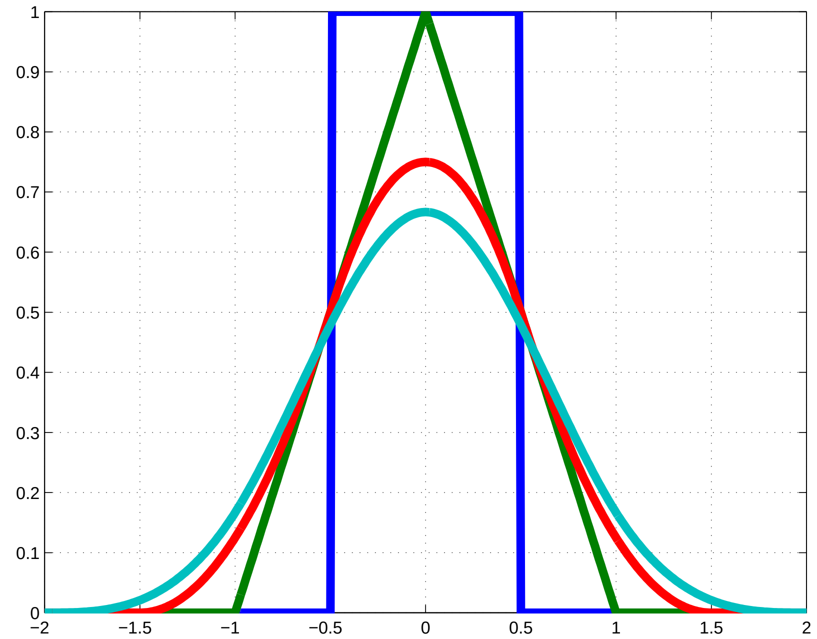
(Uniform) splines



- Piecewise polynomial of degree n
- Continuous $(n-1)^{\text{th}}$ derivative
- (Uniform) knots

Uniform B-splines

Haar	β_0
linear	β_1
quadratic	β_2
cubic	β_3



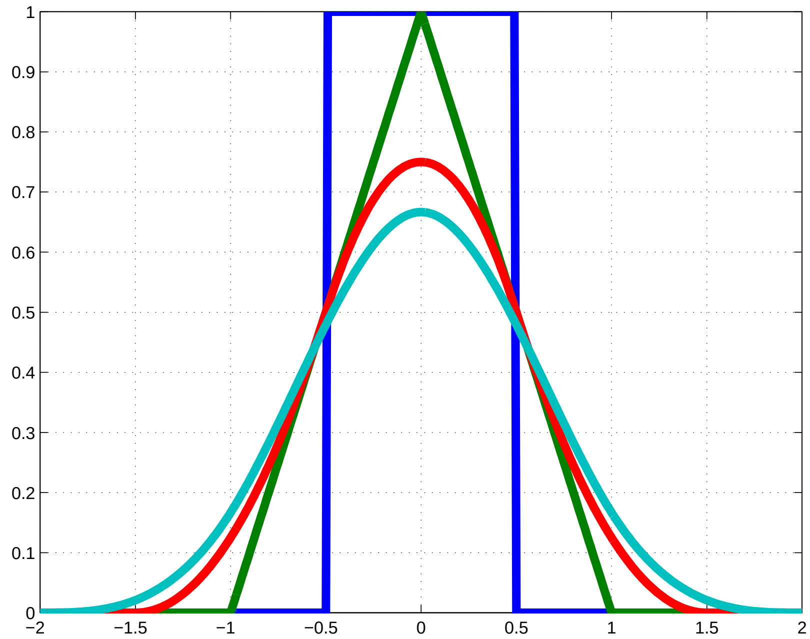
Uniform B-splines

Haar β_0

linear β_1

quadratic β_2

cubic β_3

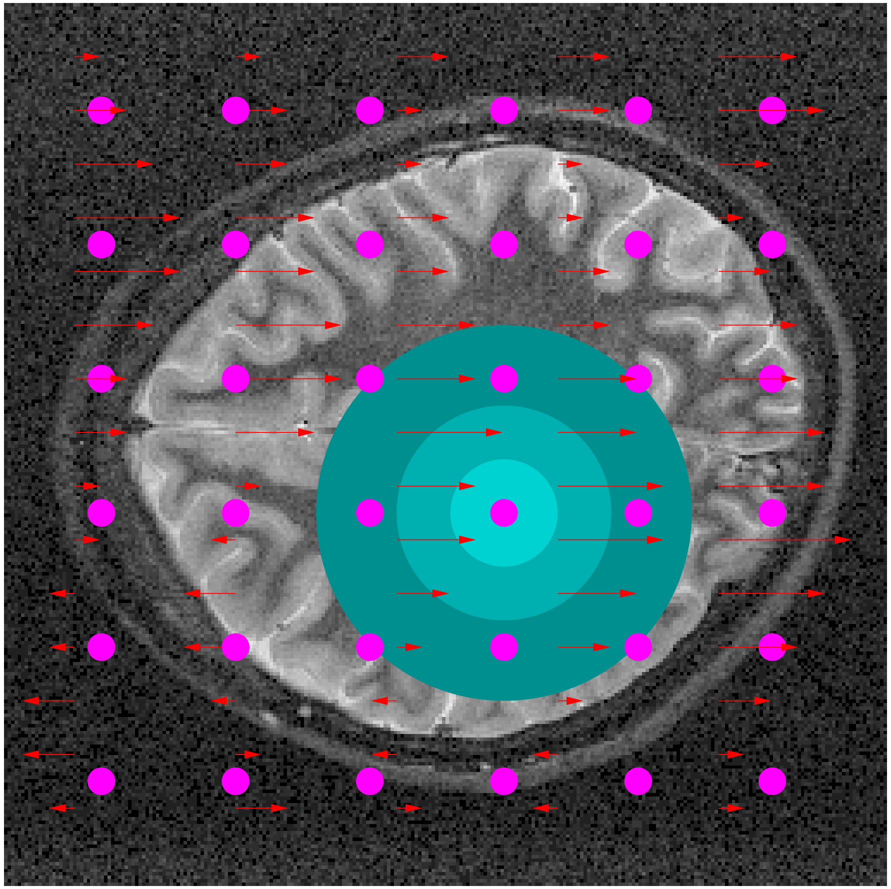


- Generation: $\beta_{n+1} = \beta_n * \beta_0$
- Basis for splines: $s(x) = \sum_i c_i \, \beta(x - i)$

Automatic registration

How does it work?

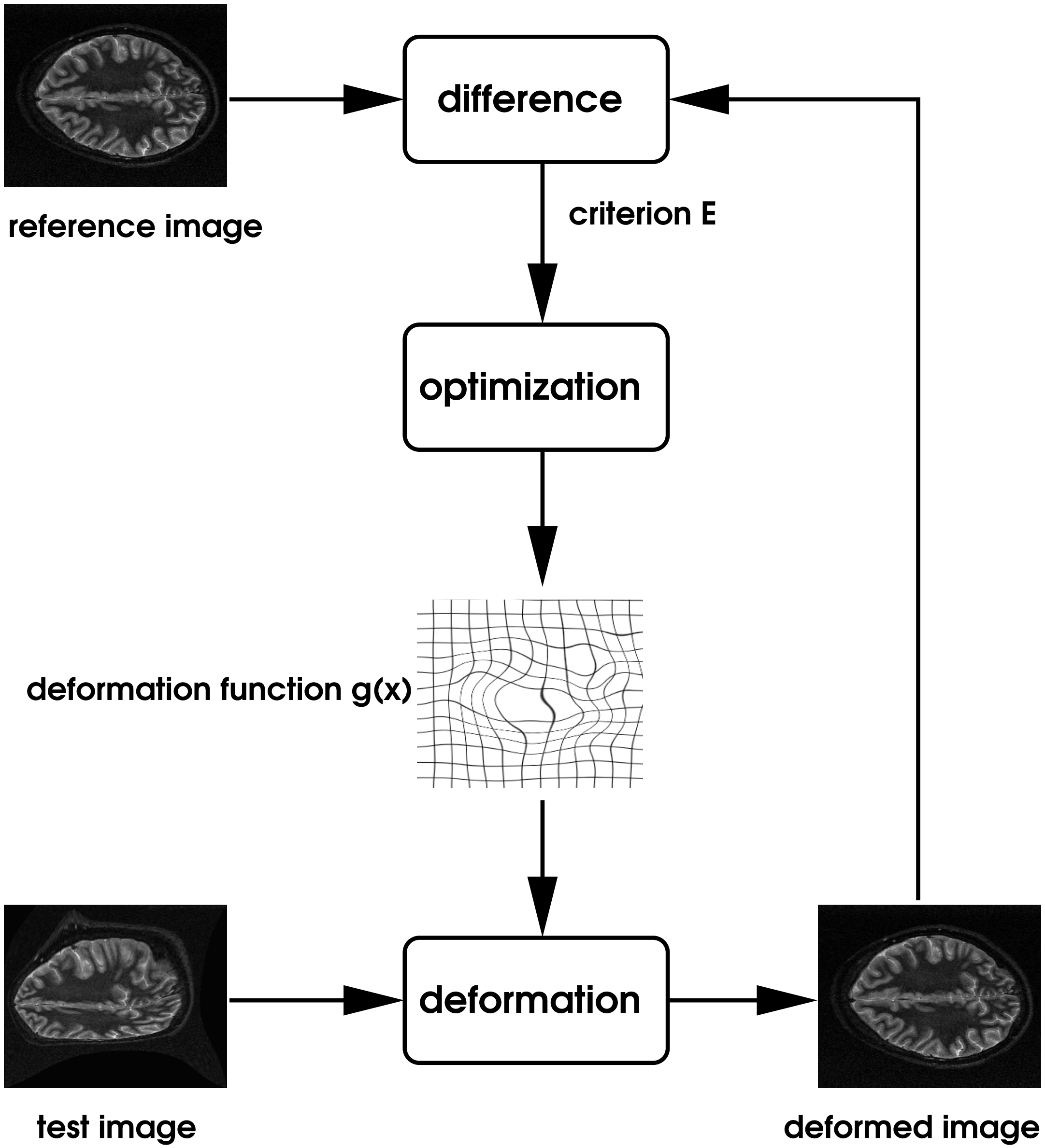
Spline based warping



- Approximation properties → precision
- Short support → speed
- Scalability
- Representability of linear transforms

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{i} \in \mathbb{Z}^2} \mathbf{c}(\mathbf{i})\, \beta(\mathbf{x}/\mathbf{h} + \mathbf{d} - \mathbf{i})$$

Registration as minimization



Registration as minimization

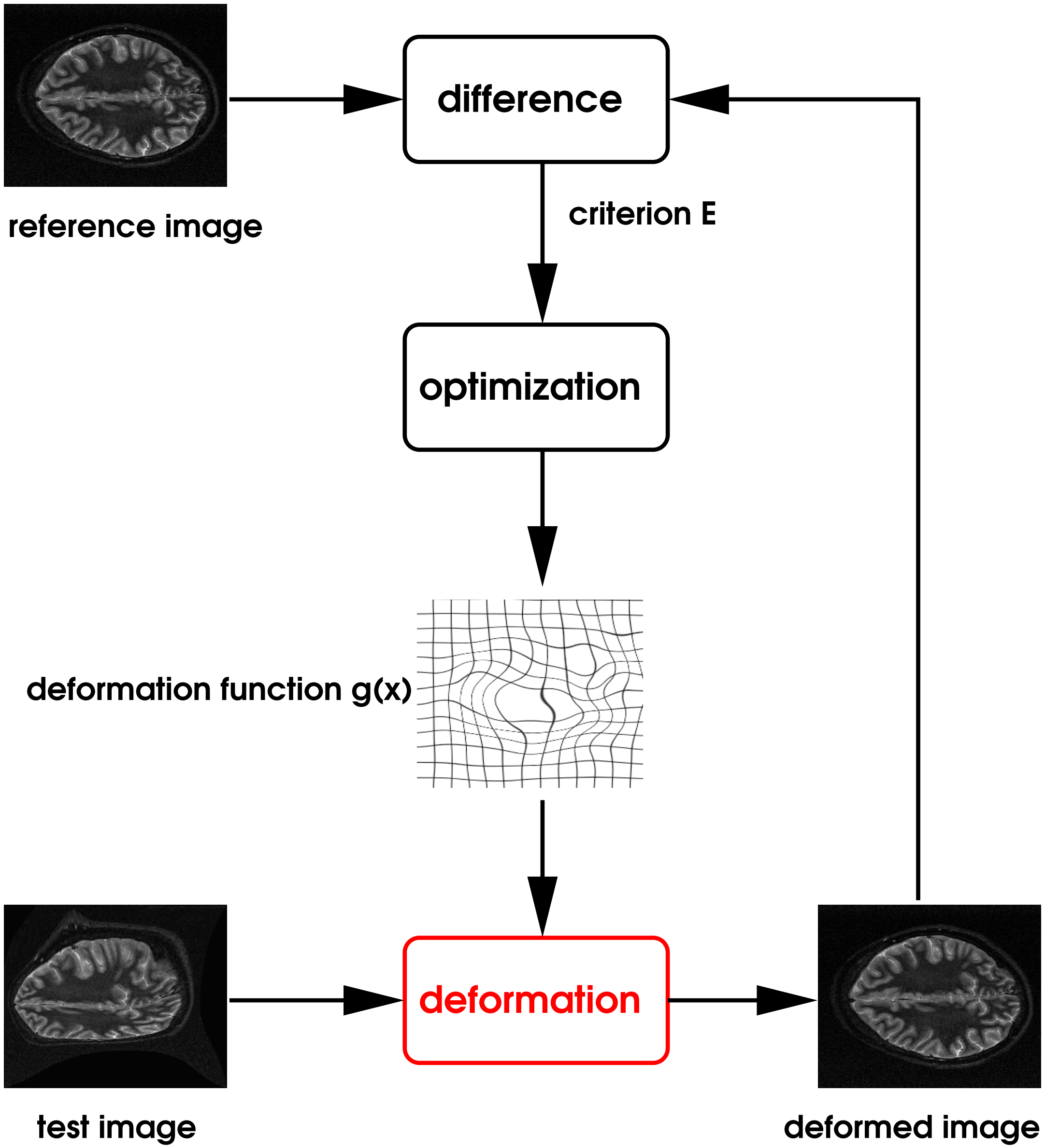
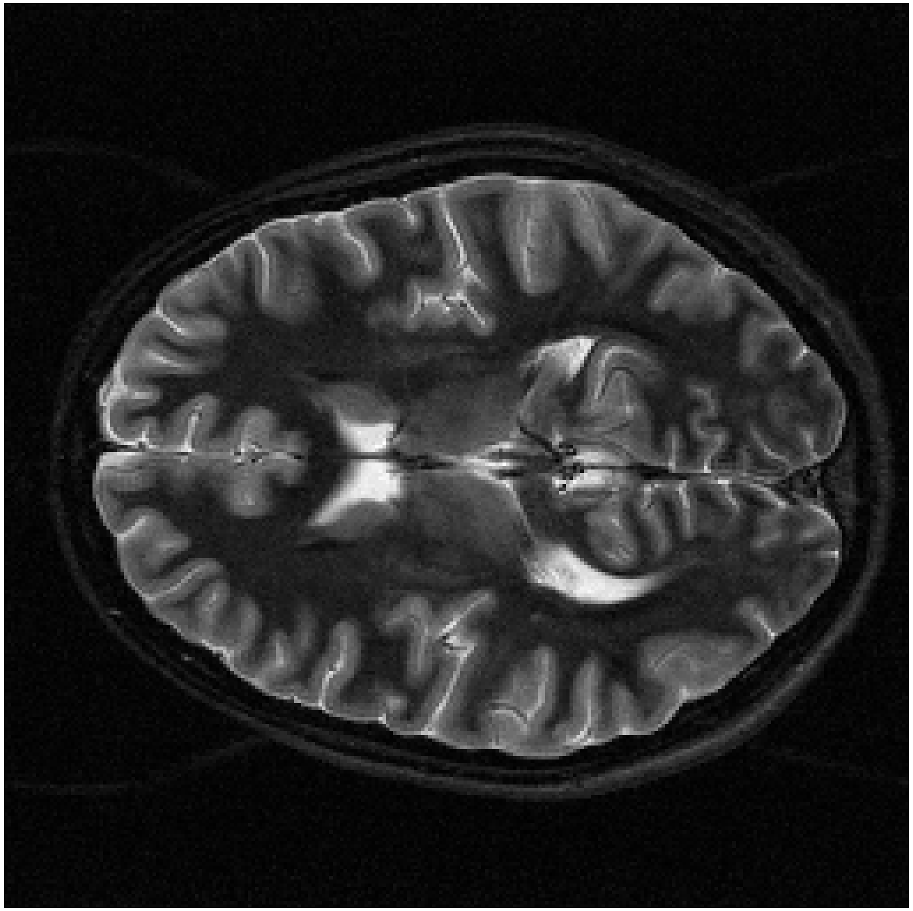
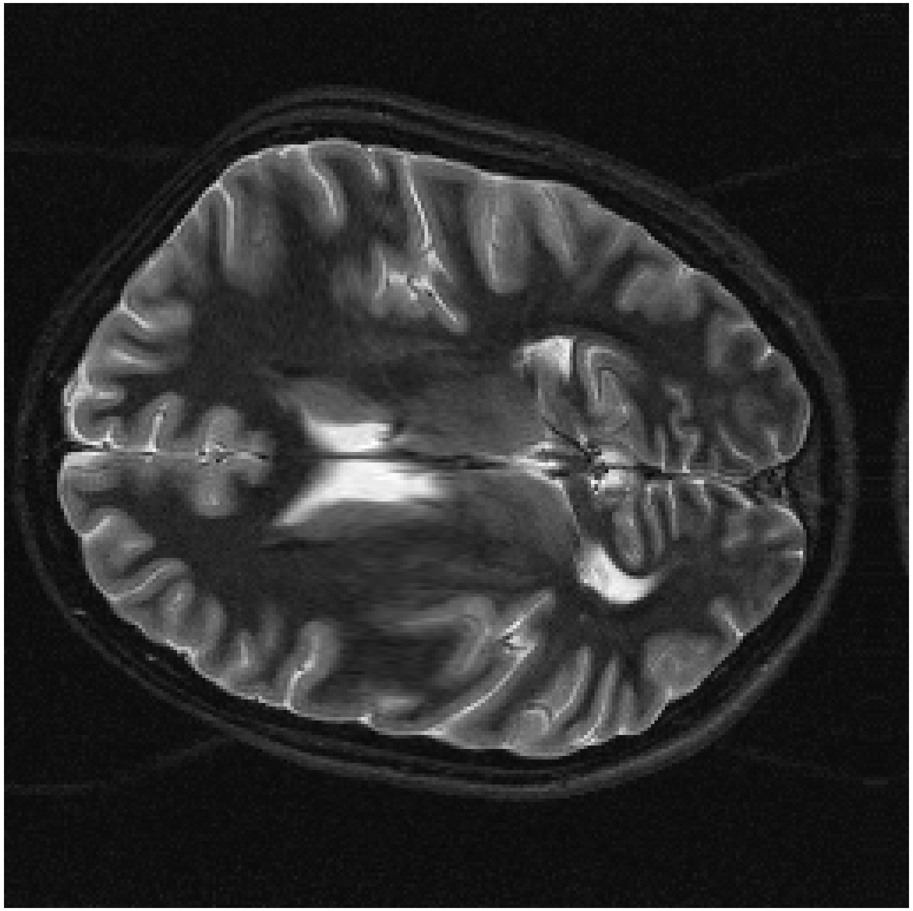


Image warping (2)



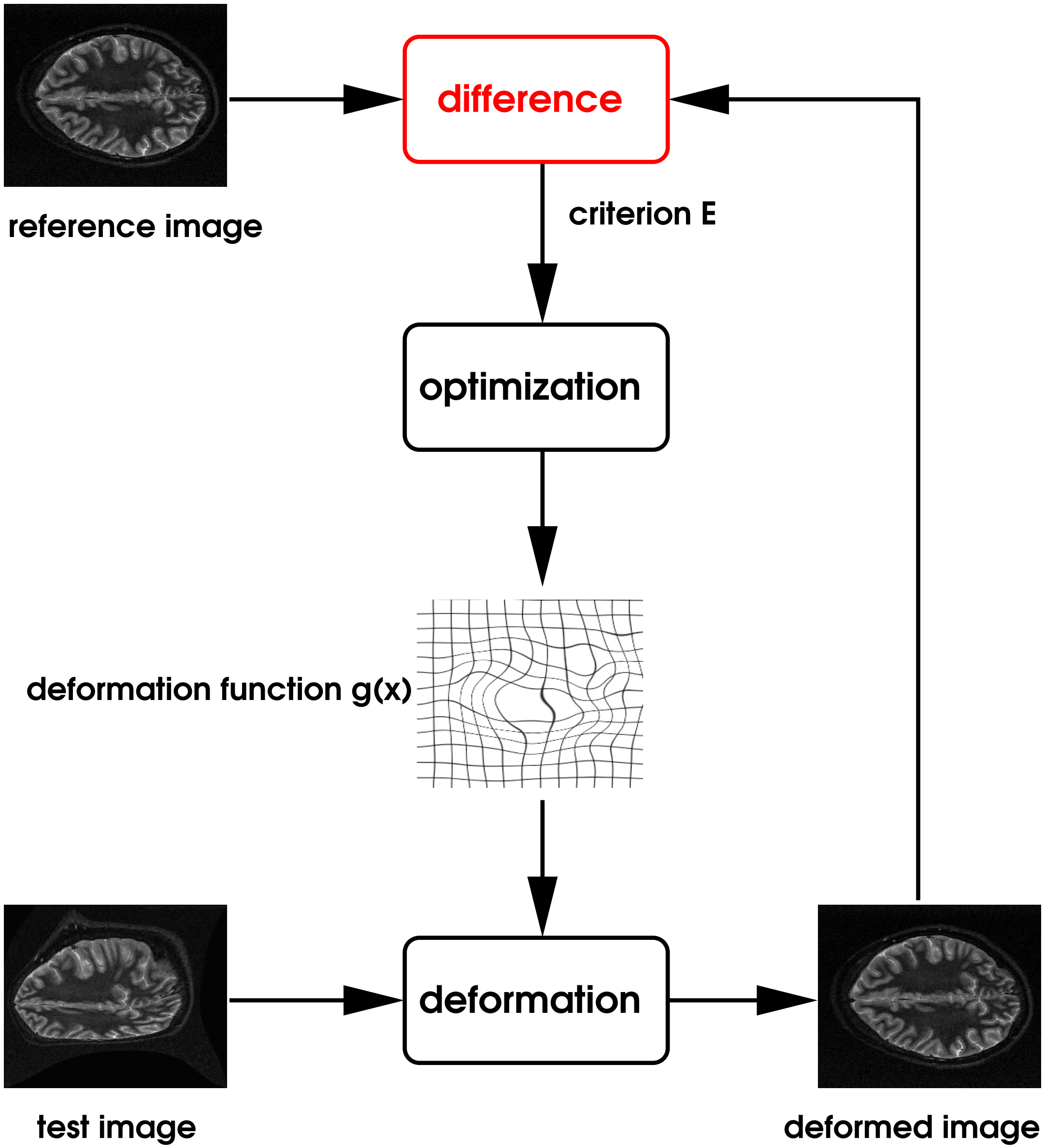
0 % deformation

Image warping (2)

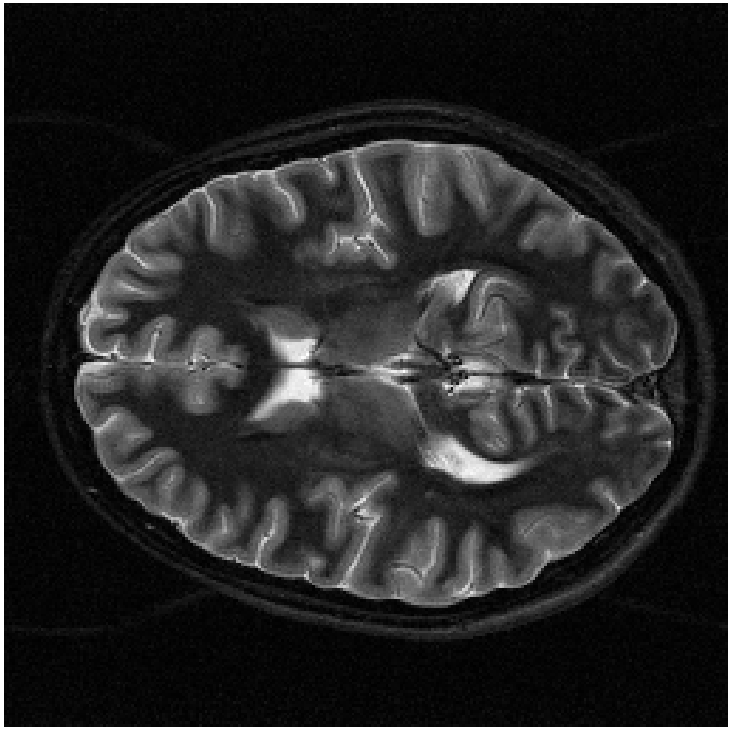


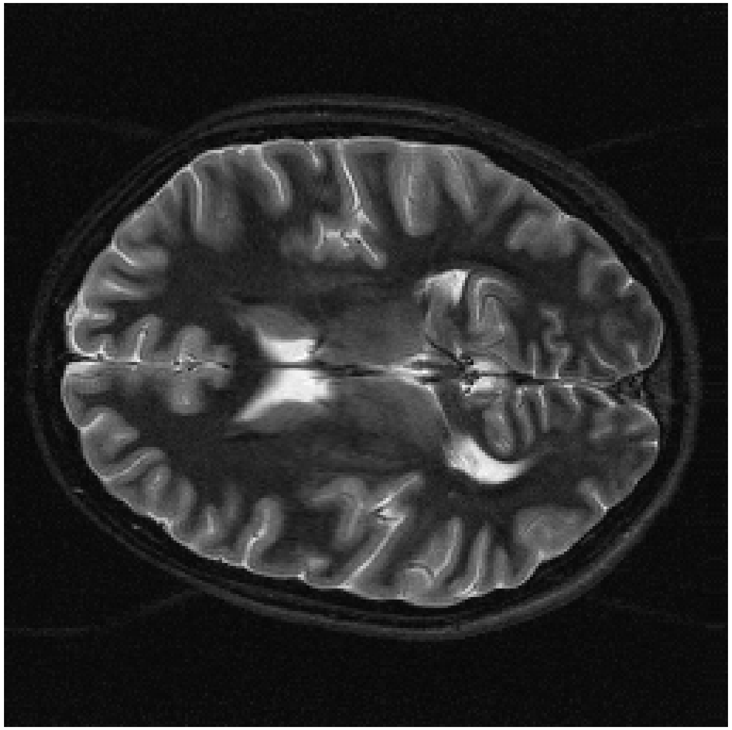
100 % deformation

Registration as minimization (2)

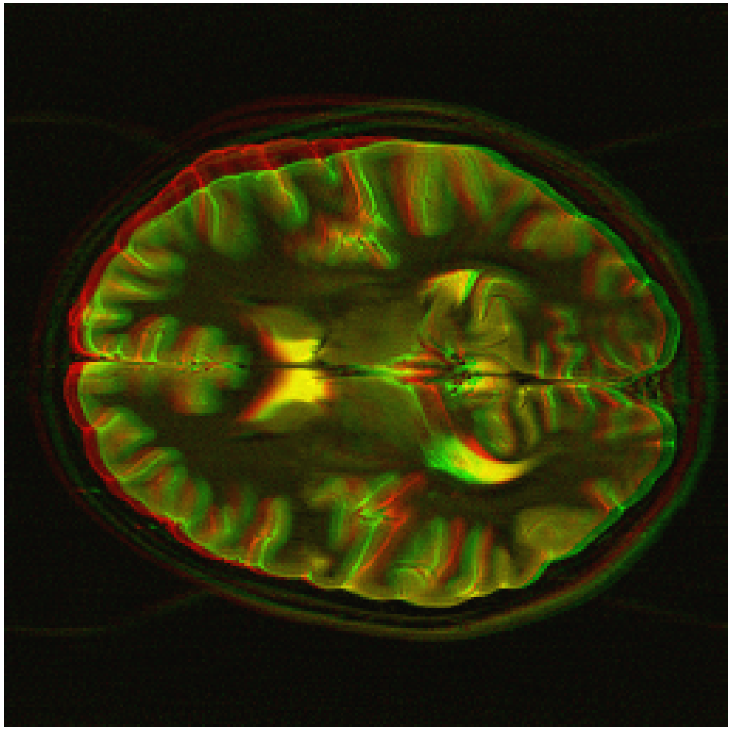


Evaluating the difference

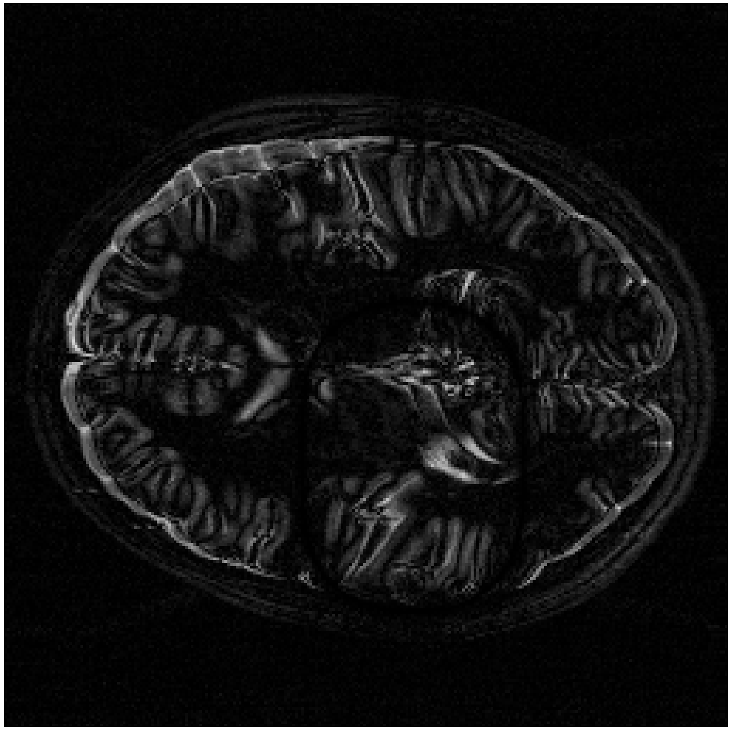




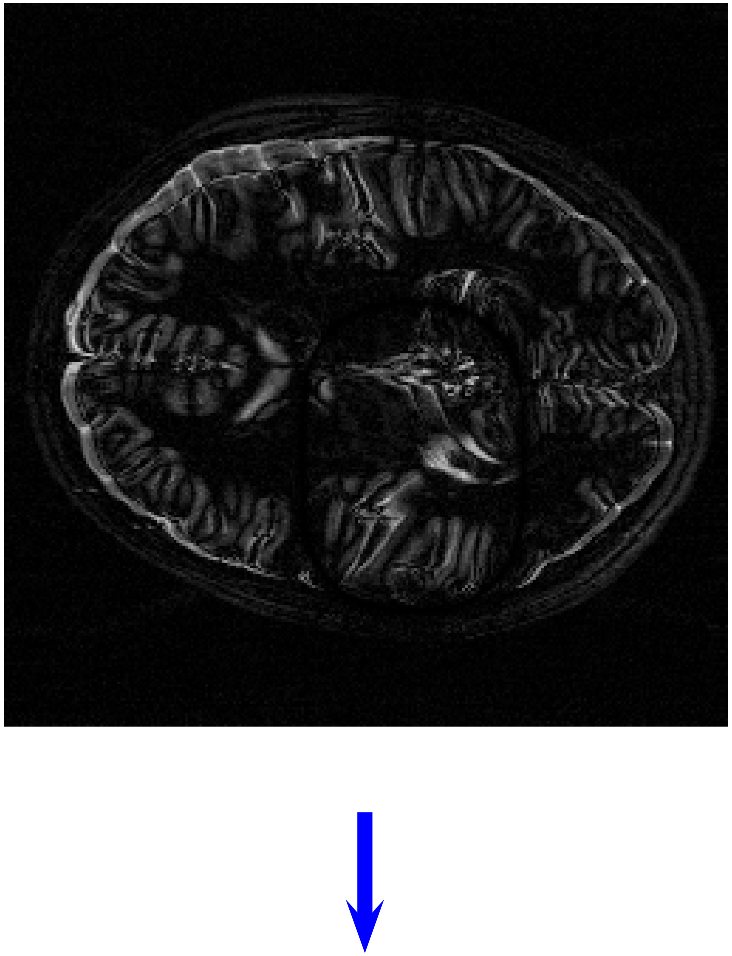
Evaluating the difference



Evaluating the difference

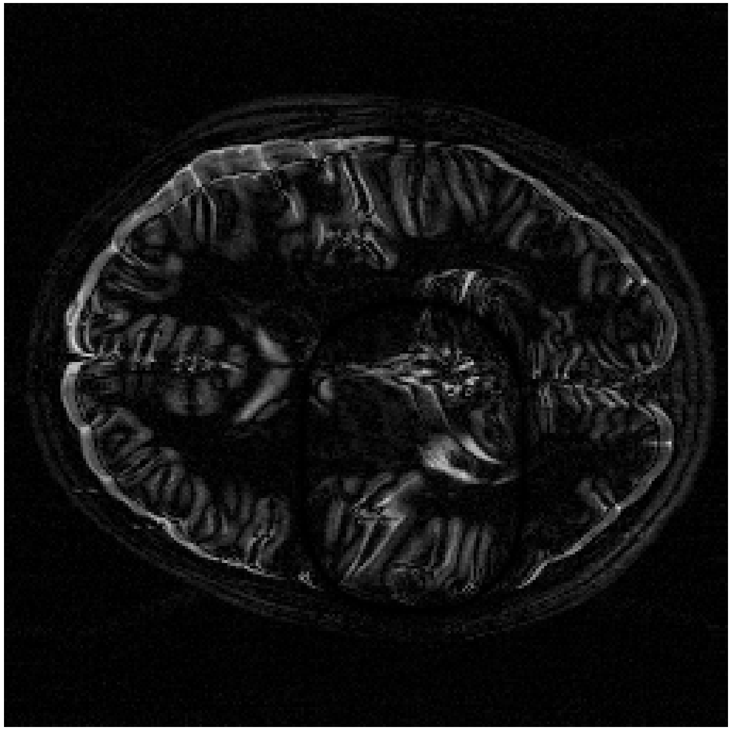


Evaluating the difference



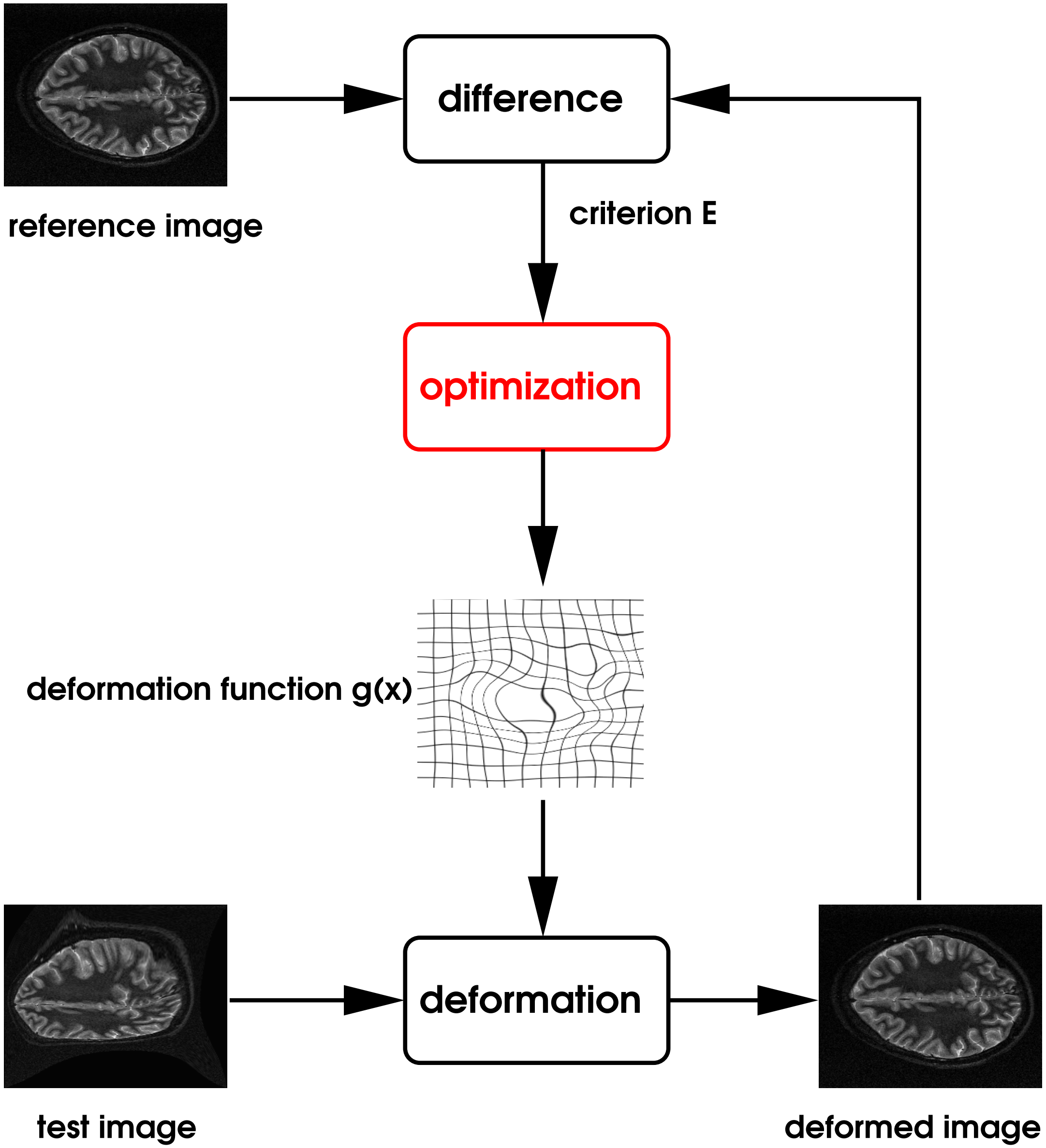
$$E = 435.7$$

Evaluating the difference



$$E = (1/N) \sum_{\mathbf{i}} \left(f_t^c(\mathbf{g}(\mathbf{i})) - f_r(\mathbf{i}) \right)^2$$

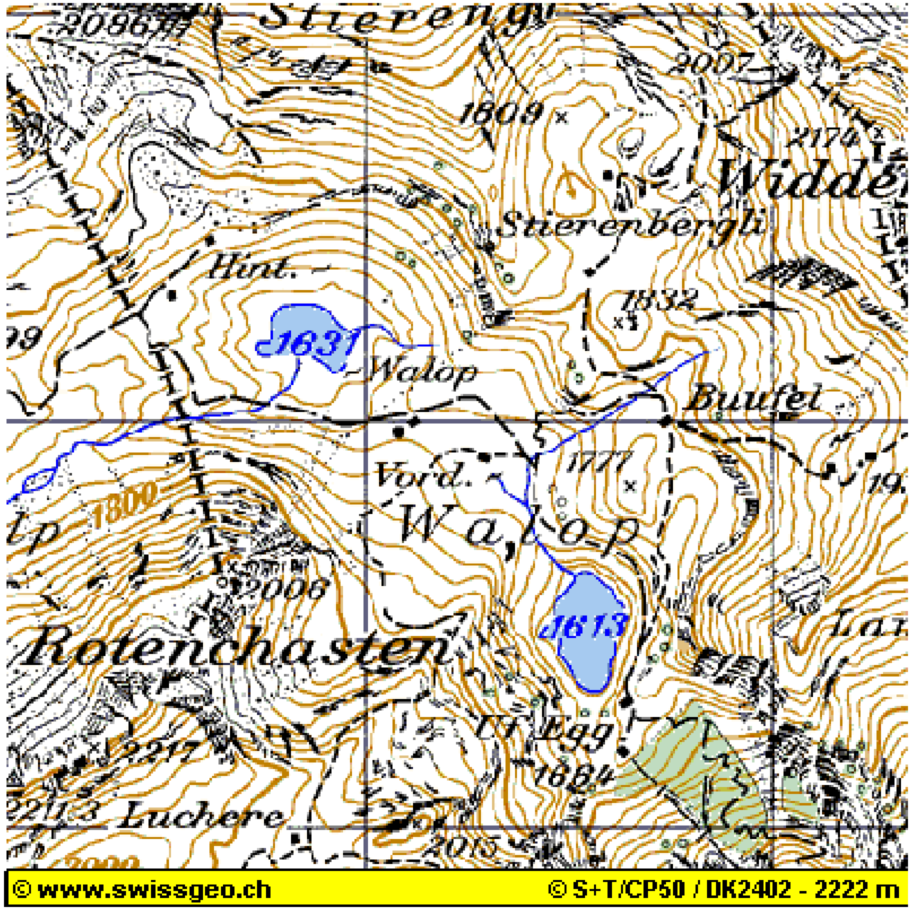
Registration as minimization (3)



Optimization

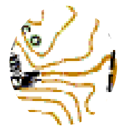
Minimize criterion E with respect to parameters c.

Optimization



Find the lowest point

Optimization



Frog search in the mist

Optimization

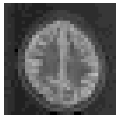
Next move

Acceleration

It works. Now let's make it run fast.

Multiresolution

32×32



Multiresolution

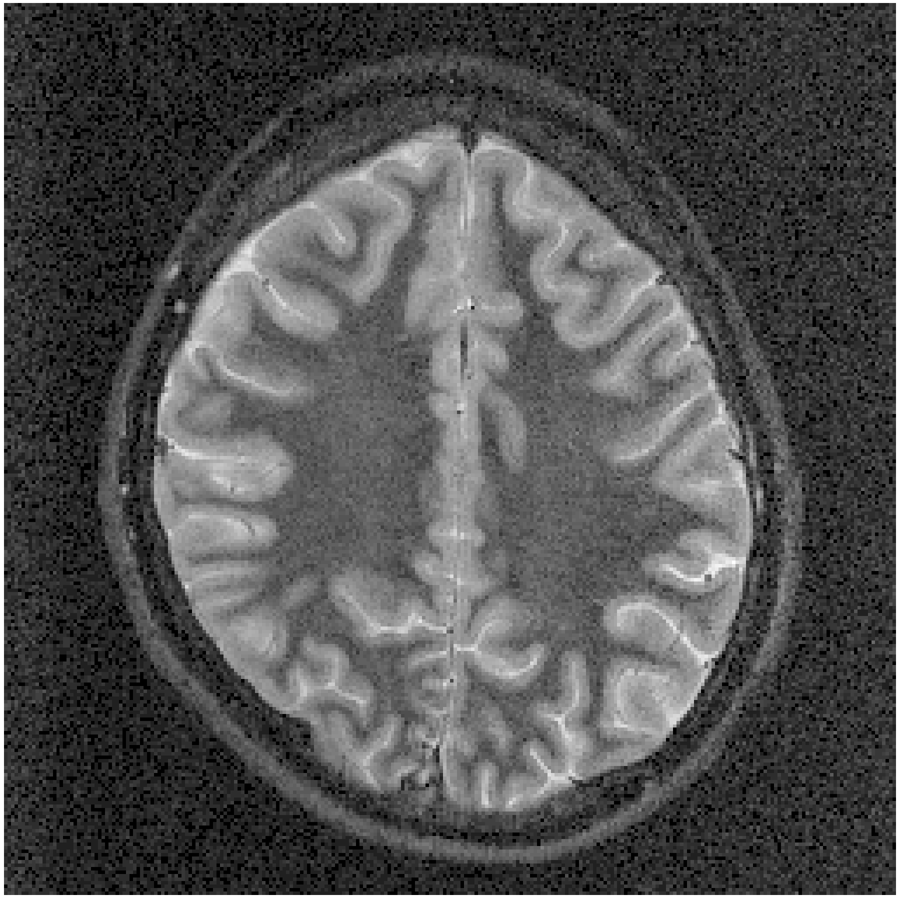
$$64 \times 64$$

Multiresolution

$$128 \times 128$$

Multiresolution

256×256



Overview (3)

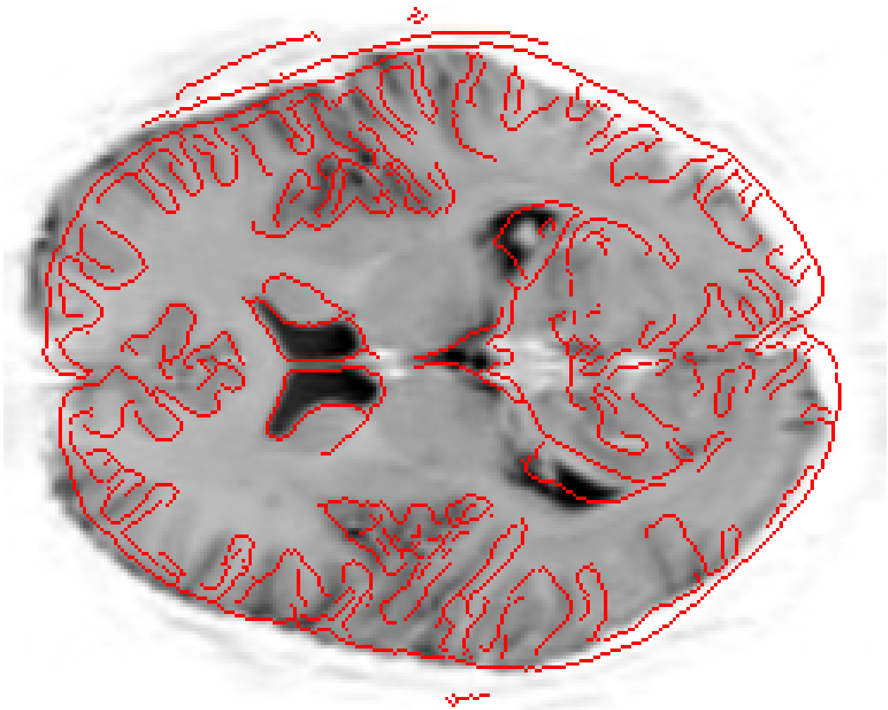
- Registration and its applications
- Manual registration
  - Interpolation
  - Variational reconstruction
- Splines
- Automatic registration
  - Algorithm
  - Semi-automatic registration
  - Applications
- Conclusions
- Party

Movie

click here

Applications

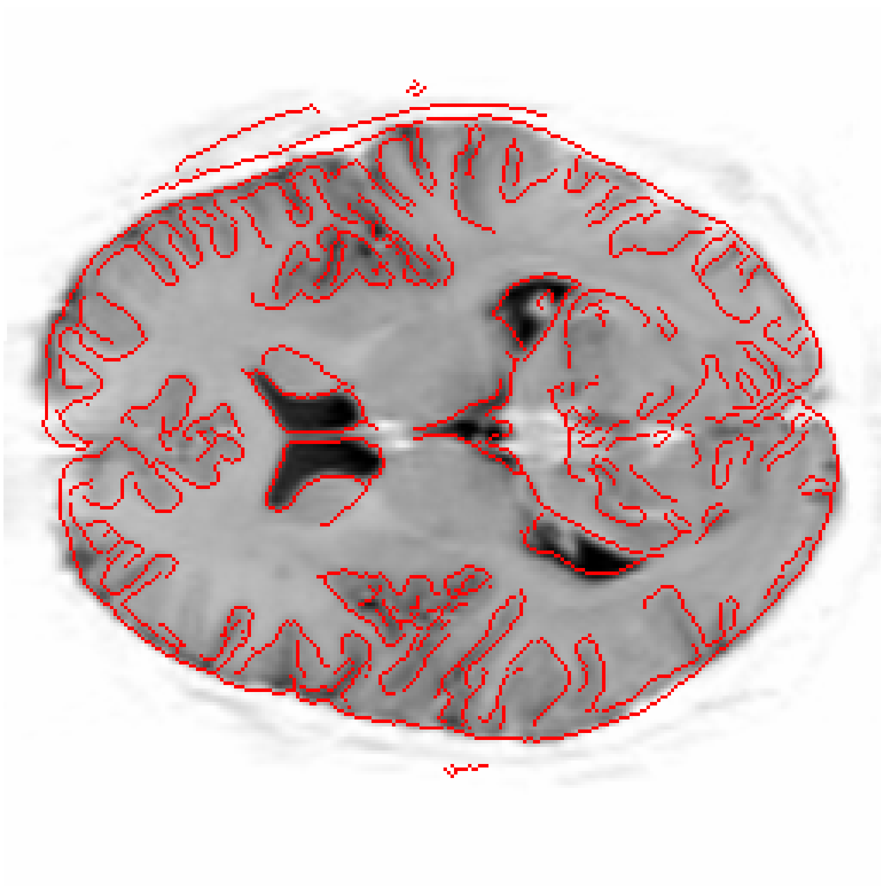
- EPI distortion



Before (with Arto Nirkko)

Applications

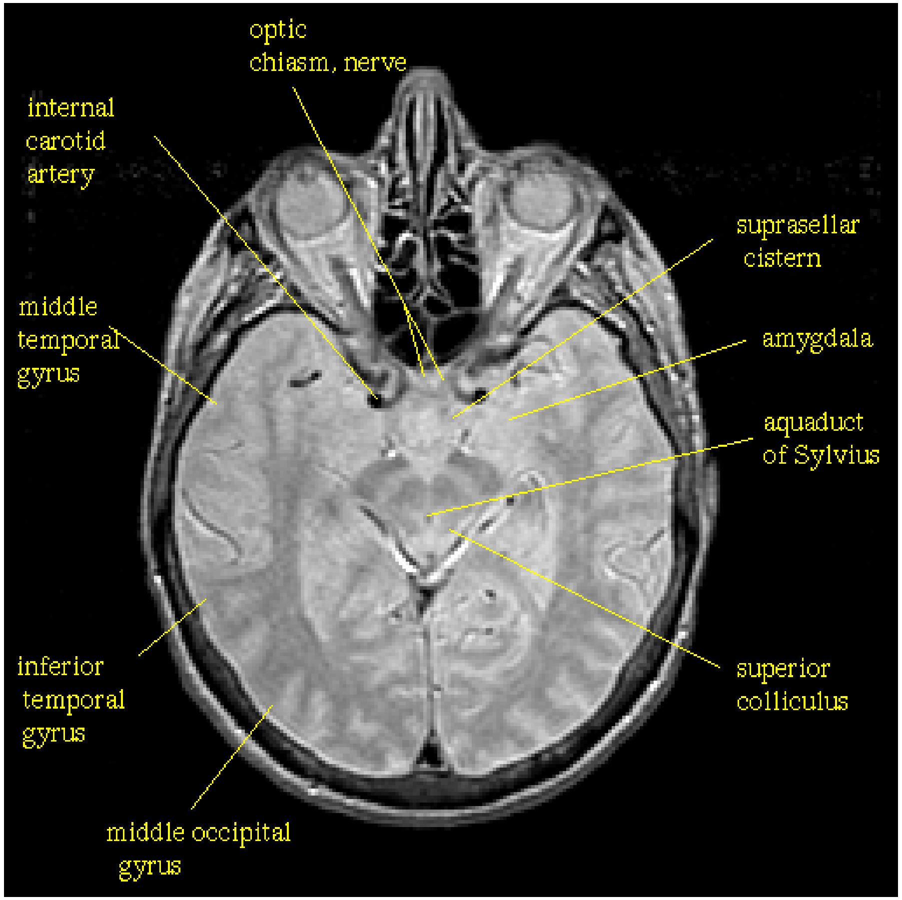
- EPI distortion



After

Applications

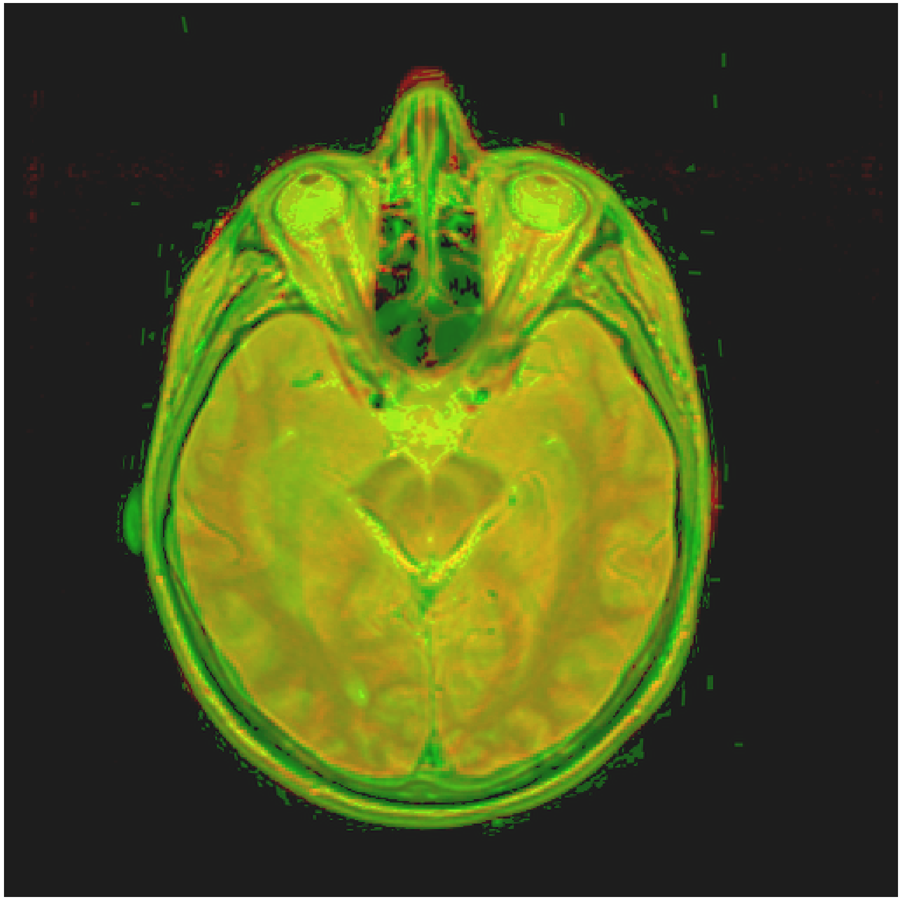
- EPI distortion
- MRI atlas



Atlas

Applications

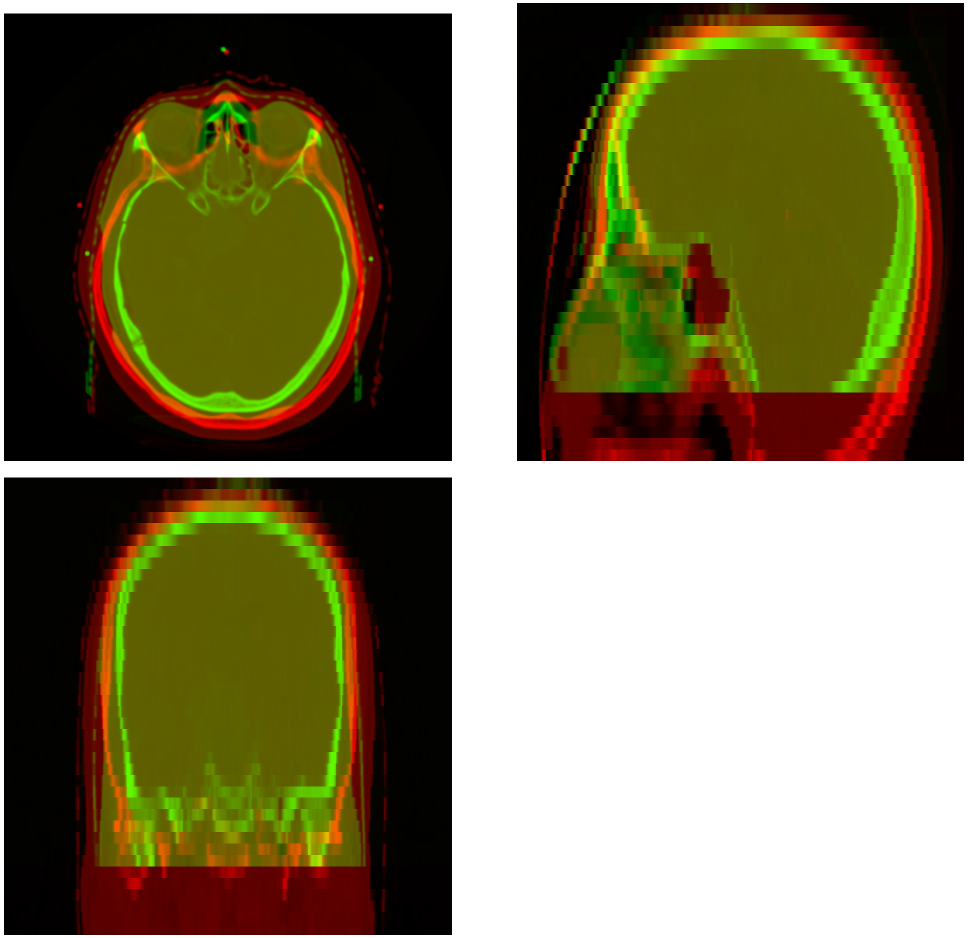
- EPI distortion
- MRI atlas



Aligned

Applications

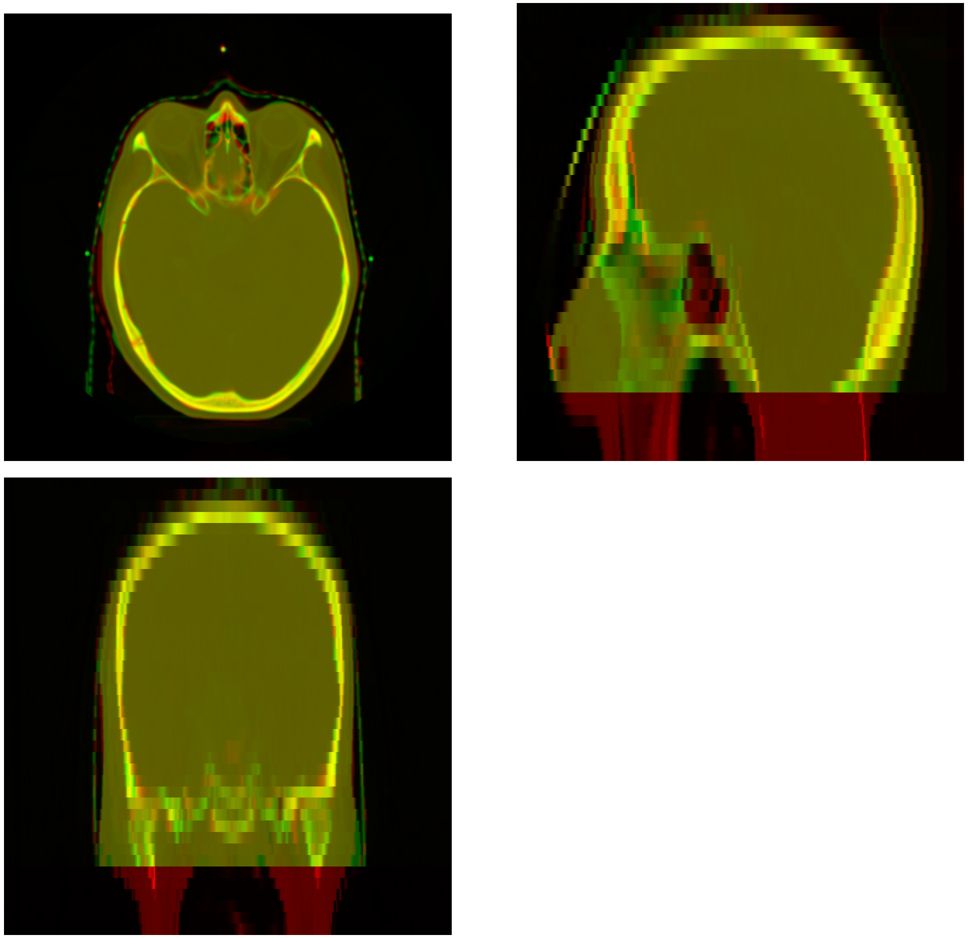
- EPI distortion
- MRI atlas
- CT alignment



Before

Applications

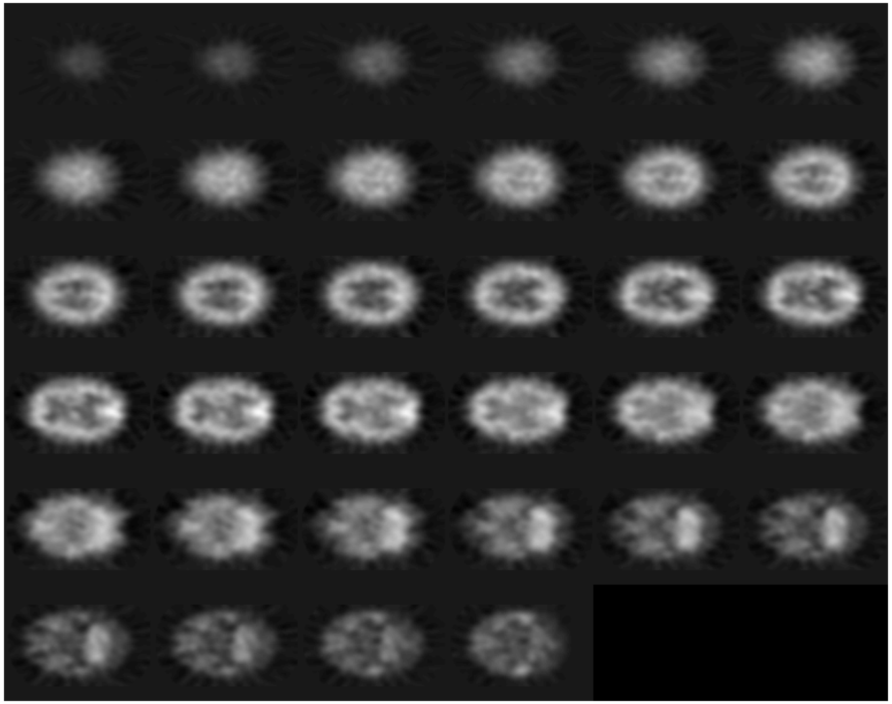
- EPI distortion
- MRI atlas
- CT alignment



After

Applications

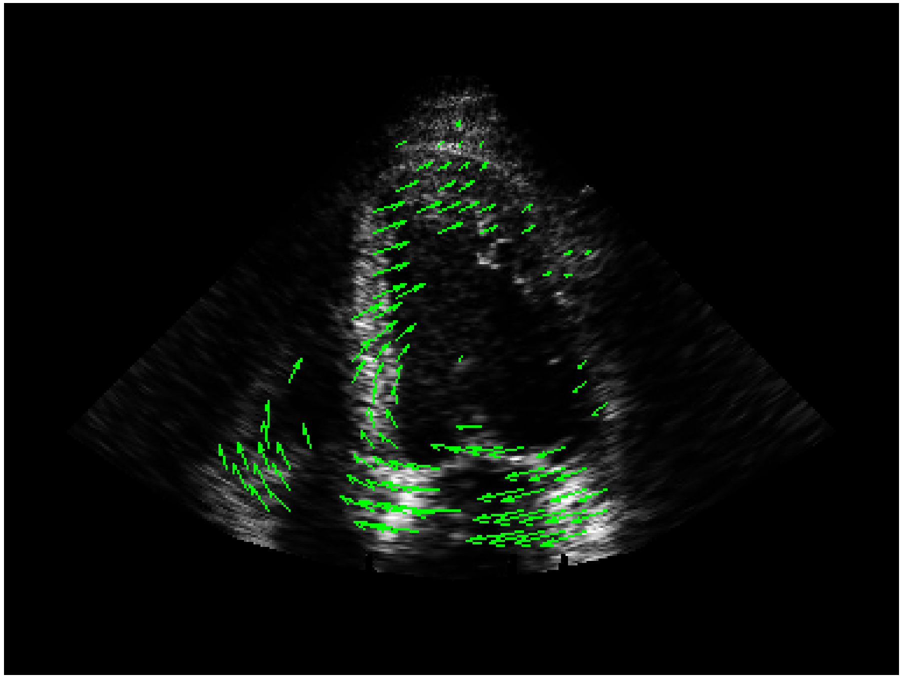
- EPI distortion
- MRI atlas
- CT alignment
- SPECT atlas



(with University Hospital in Geneva)

Applications

- EPI distortion
- MRI atlas
- CT alignment
- SPECT atlas
- Ultrasound



velocity (with María J. Ledesma-Carbayo)

Conclusions

- Image registration is useful
- Our registration algorithm works
- Interpolation is interesting
- Variational is elegant
- Splines are great

The End

Díky!

Merci!

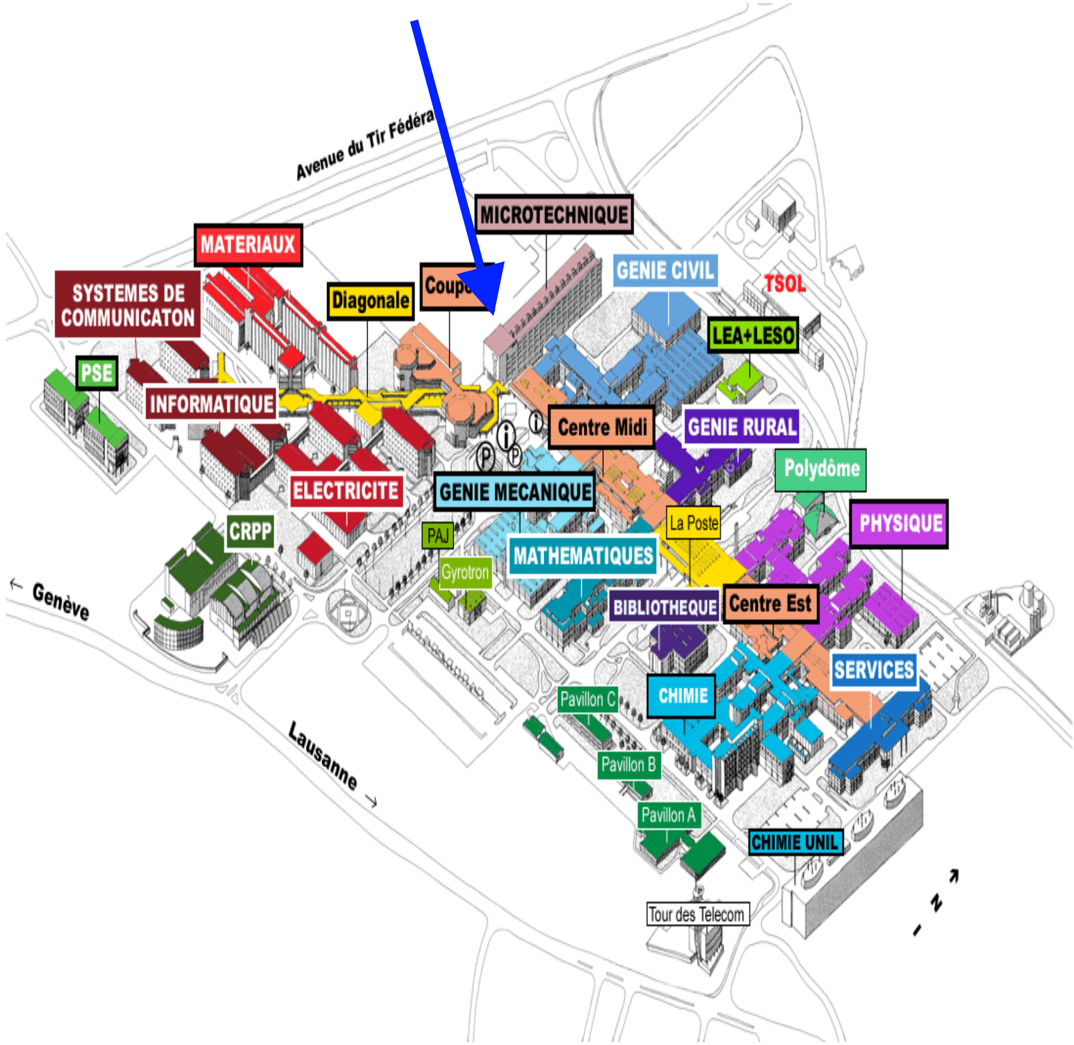
Thank you!

¡Gracias!

Grazie!

Danke!

Party



Bâtiment de microtechnique, 4th floor