MEDICAL IMAGE COMPUTING (CAP 5937)

LECTURE 17: Medical Image Registration III (Advanced): FFD with B-Splines, Diffeomorphic Image Registration, and Regularizations

Dr. Ulas Bagci

HEC 221, Center for Research in Computer Vision (CRCV), University of Central Florida (UCF), Orlando, FL 32814.

bagci@ucf.edu or bagci@crcv.ucf.edu

Outline

- Deformable Image Registration
  – B-spline parametrization and Free Form Deformation
- Optimization
- Diffeomorphic Image Registration

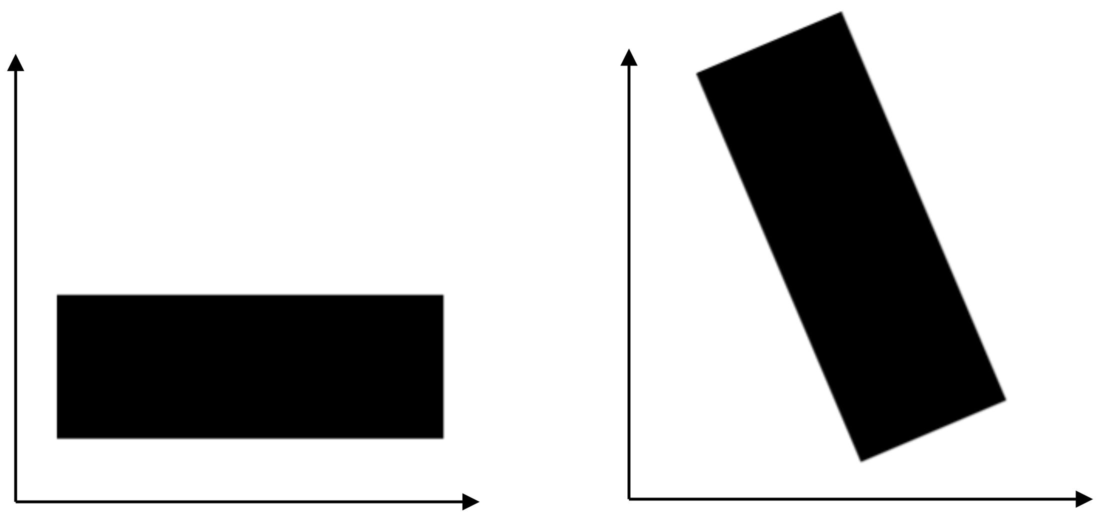
Rigid Transformation

- Rotation
- Translation
- Scale

$$\vec{p}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \vec{p}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad \vec{s}_1 = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad \vec{t}_1 = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\vec{p}_2 = \vec{t} + \vec{s} R \vec{p}_1$$

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



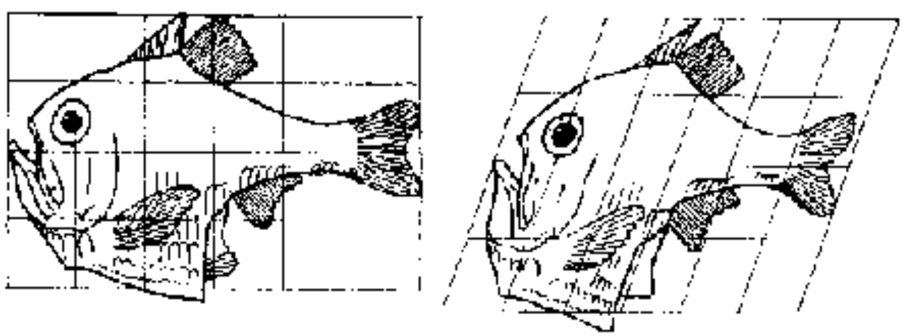
Rigid Transformation

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \mathbf{R} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \mathbf{t} \quad \text{with } \mathbf{R} = \begin{pmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \text{ and } \mathbf{t} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix},$$

Affine Transformation

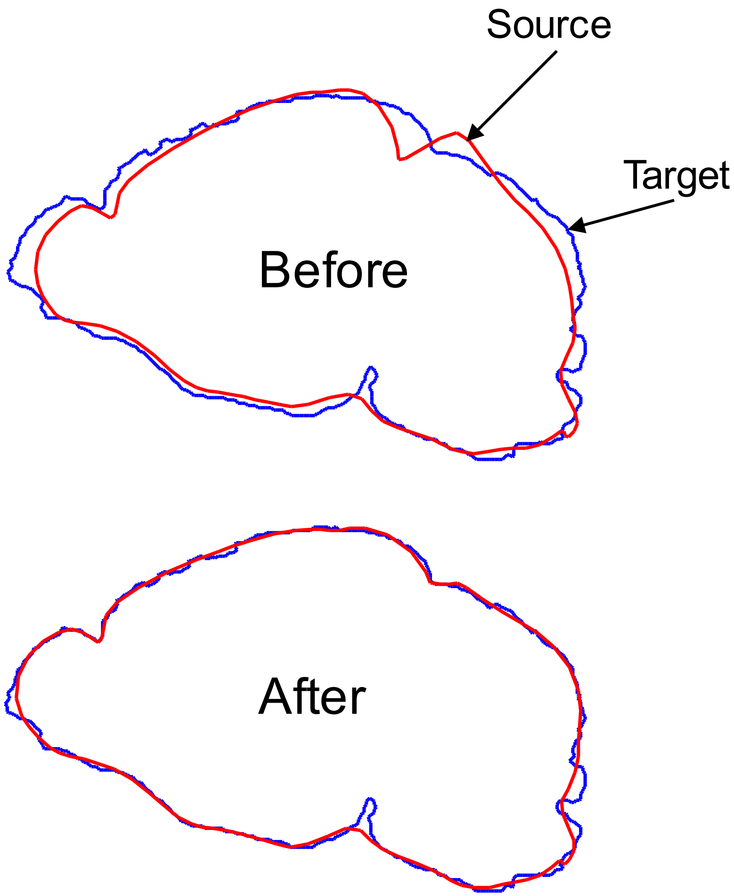
- Rotation
- Translation
- Scale
- Shear
  - No more preservation of lengths and angles
  - Parallel lines are preserved

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} + \begin{bmatrix} a_{11} + a_{12} \\ a_{21} + a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

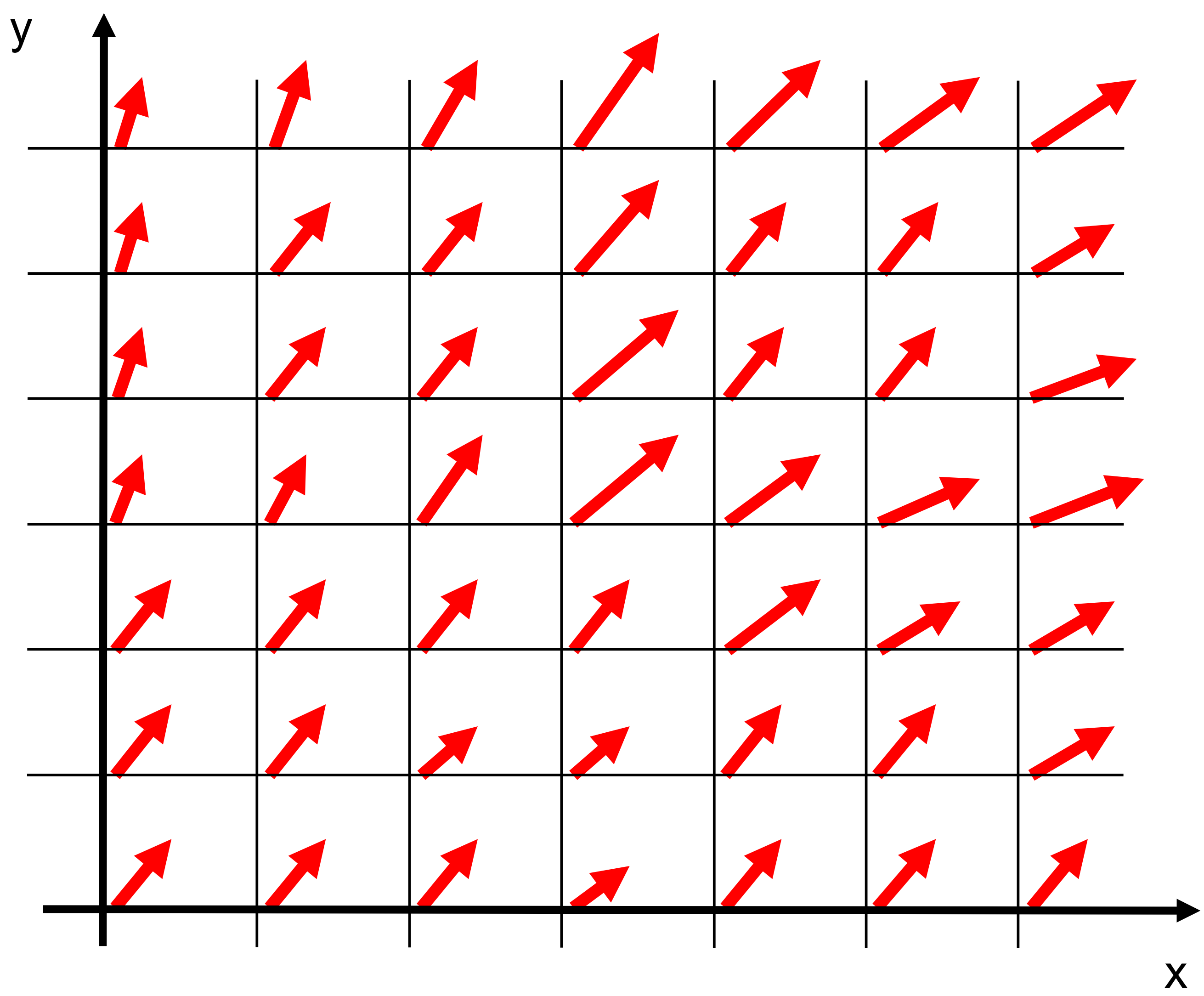


Argyropelecus olfersi. *Sternoptyx diaphana.*

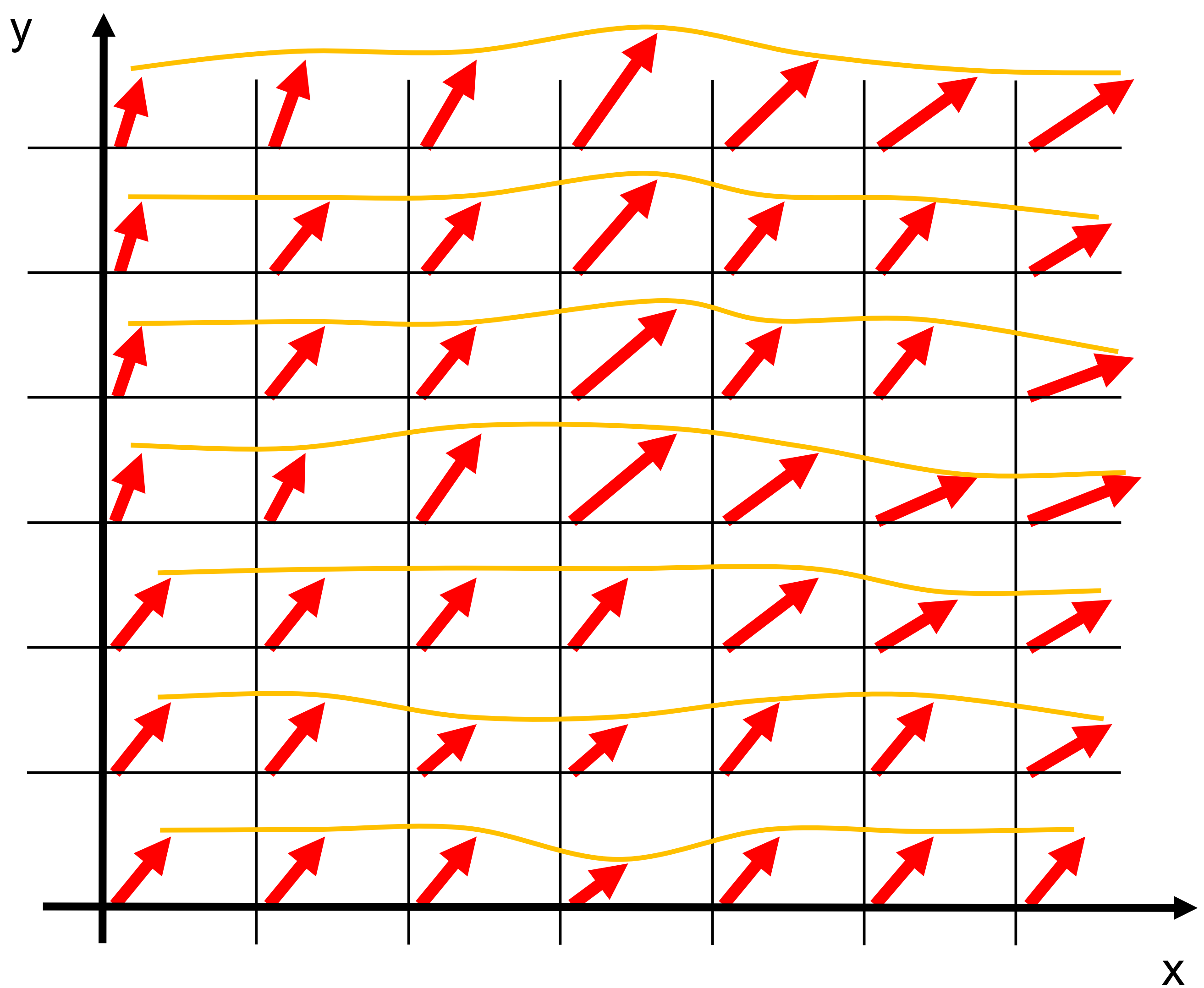
Non-Rigid Deformation



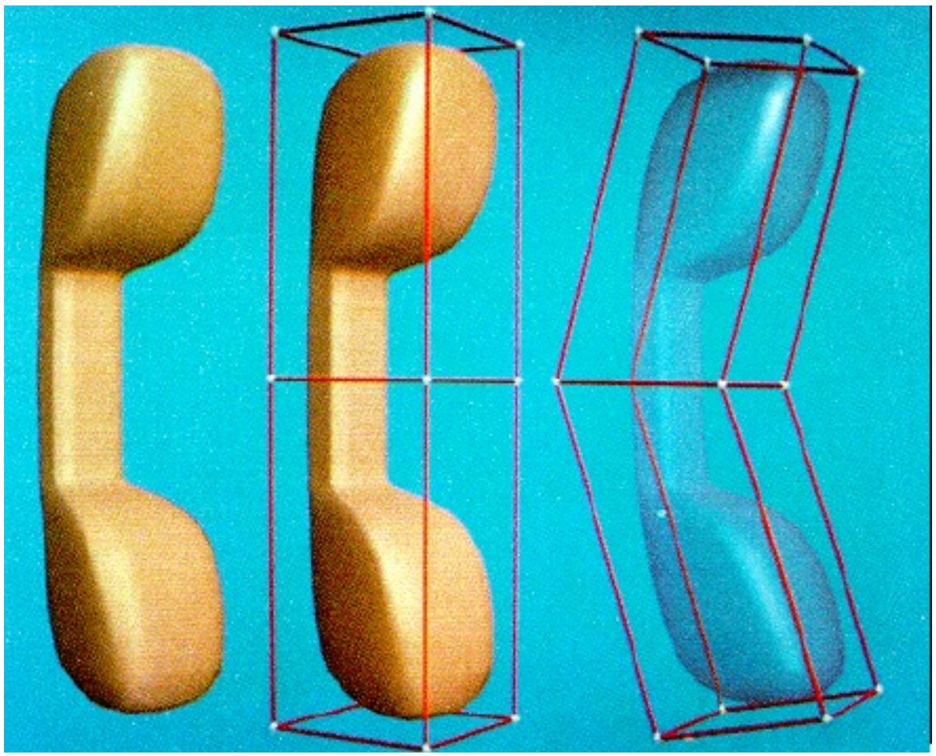
Deformation Fields

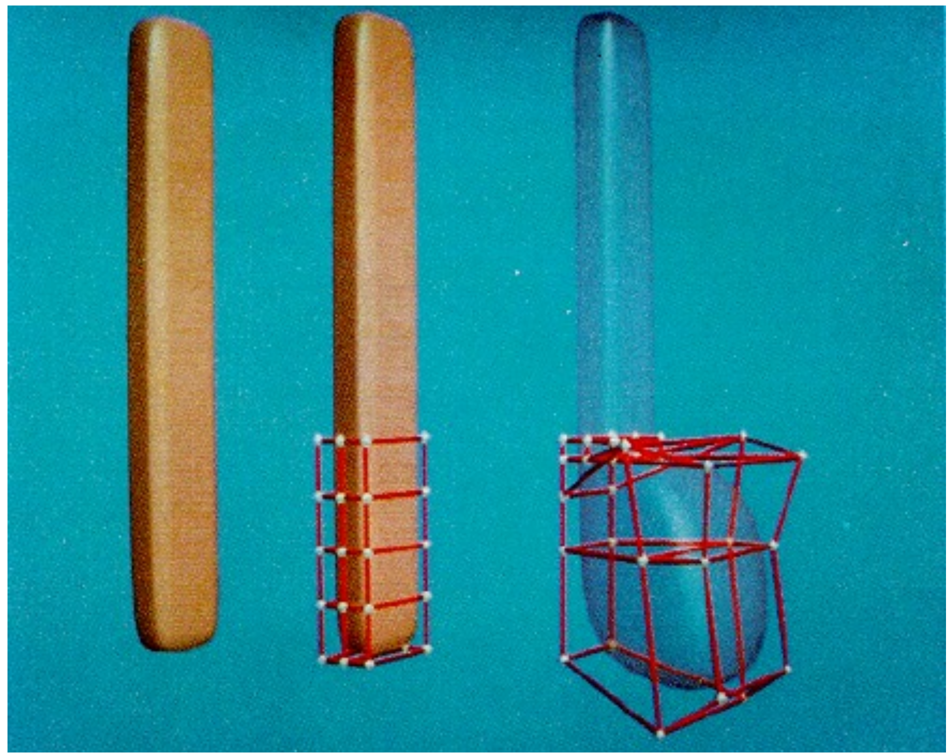


Deformation Fields



Free Form Deformation





Credits: Sederberg and Parry, SIGGRAPH (1986)

$$\mathbf{T}(x, y, z) = \mathbf{T}_{\text{global}}(x, y, z) + \mathbf{T}_{\text{local}}(x, y, z).$$

Global Motion Model

$$\mathbf{T}_{\mathrm{global}}(x, y, z) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \theta_{14} \\ \theta_{24} \\ \theta_{34} \end{pmatrix}$$

(12 degrees of freedom)

Local Motion Model

- The affine transformation captures only the global motion.
- An additional transformation is required, which models the local deformation

$$\mathbf{T}_{\mathrm{local}}(x, y, z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \phi_{i+l,\, j+m,\, k+n}$$

where $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$ and where B_l represents the lth basis function of the B-spline

Local Motion Model

- The affine transformation captures only the global motion.
- An additional transformation is required, which models the local deformation

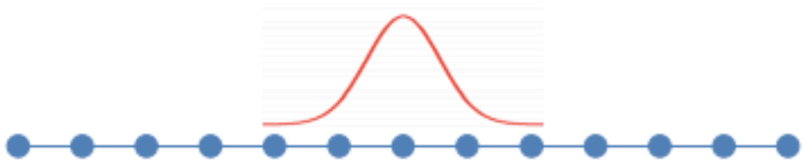
$$\boxed{\begin{aligned}&\mathbf{T}_{\text{local}}(x, y, z)\\&\quad=\sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\phi_{i+l,\,j+m,\,k+n}\end{aligned}}$$

$$\begin{aligned}B_0(u) &= (1-u)^3/6\\B_1(u) &= (3u^3-6u^2+4)/6\\B_2(u) &= (-3u^3+3u^2+3u+1)/6\\B_3(u) &= u^3/6.\end{aligned}$$

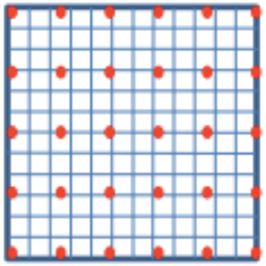
FFD with B-Splines

- **Rueckert 1999**
- **Cubic B-splines** (degree $D = 3$), basis functions all have same shape and are translated versions of each other

- **Compact support** of D+1 control points



- **Regular grid** of control points



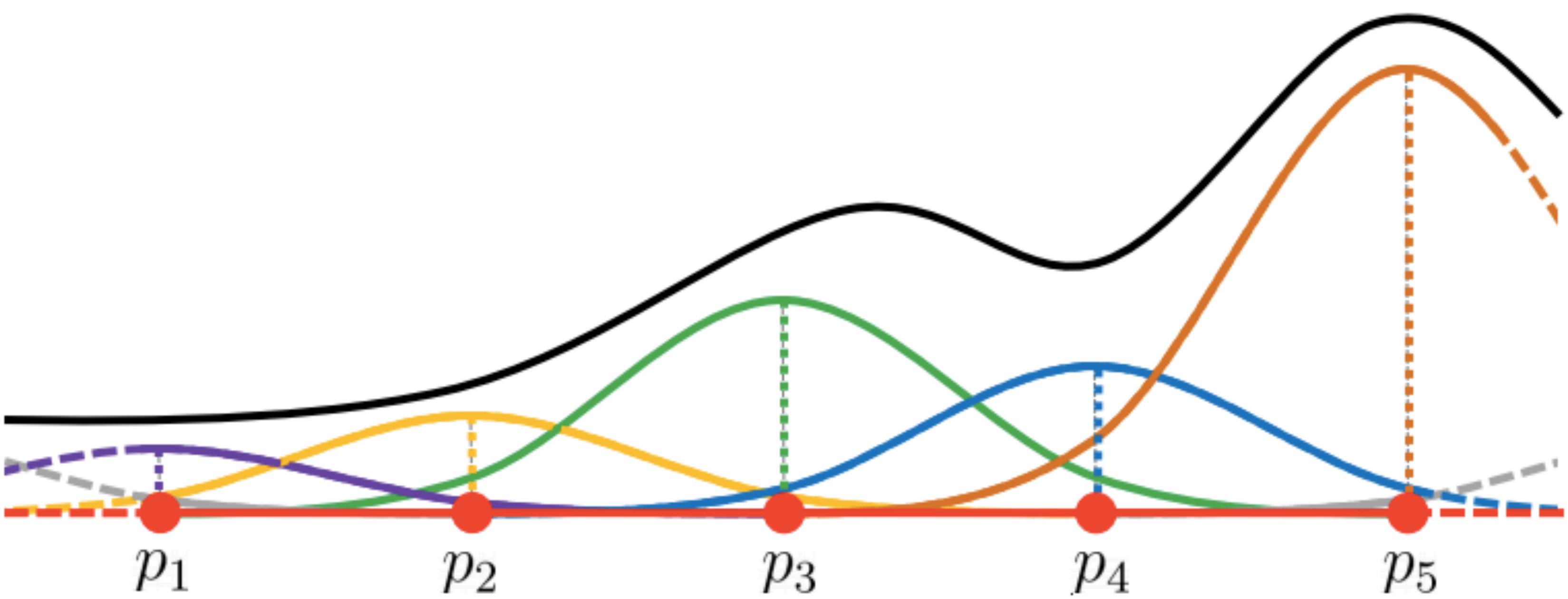
- **More:** piecewise polynomial, inherent smoothness, differentiability, hierarchical

B-Spline / Math

$$u_p(x) = \sum_k p_k B_k(x)$$

Parameters: $p_k \in \mathbb{R}^d$

Basis functions: $B_k : \Omega \to \mathbb{R}$

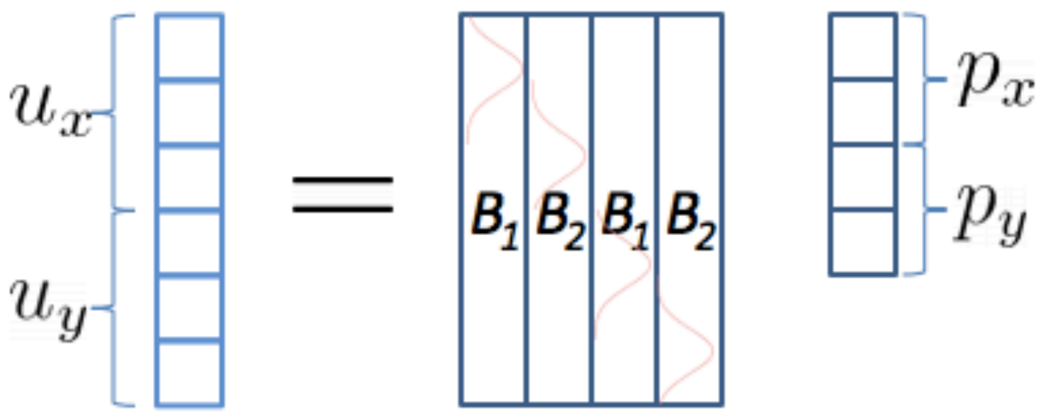


- Parametrization (linear model):

$$p_k \in \mathbb{R}^d \qquad B_k : \Omega \to \mathbb{R}$$

$$u_p(x) = \sum_k p_k B_k(x)$$

$$u = B\ p$$



Deformation with B-Splines

Original Lena

Deformation with B-Splines

Deformed with B-Spline - Lena

B-Spline Parametrization

- *Deformation* is modeled by B-splines

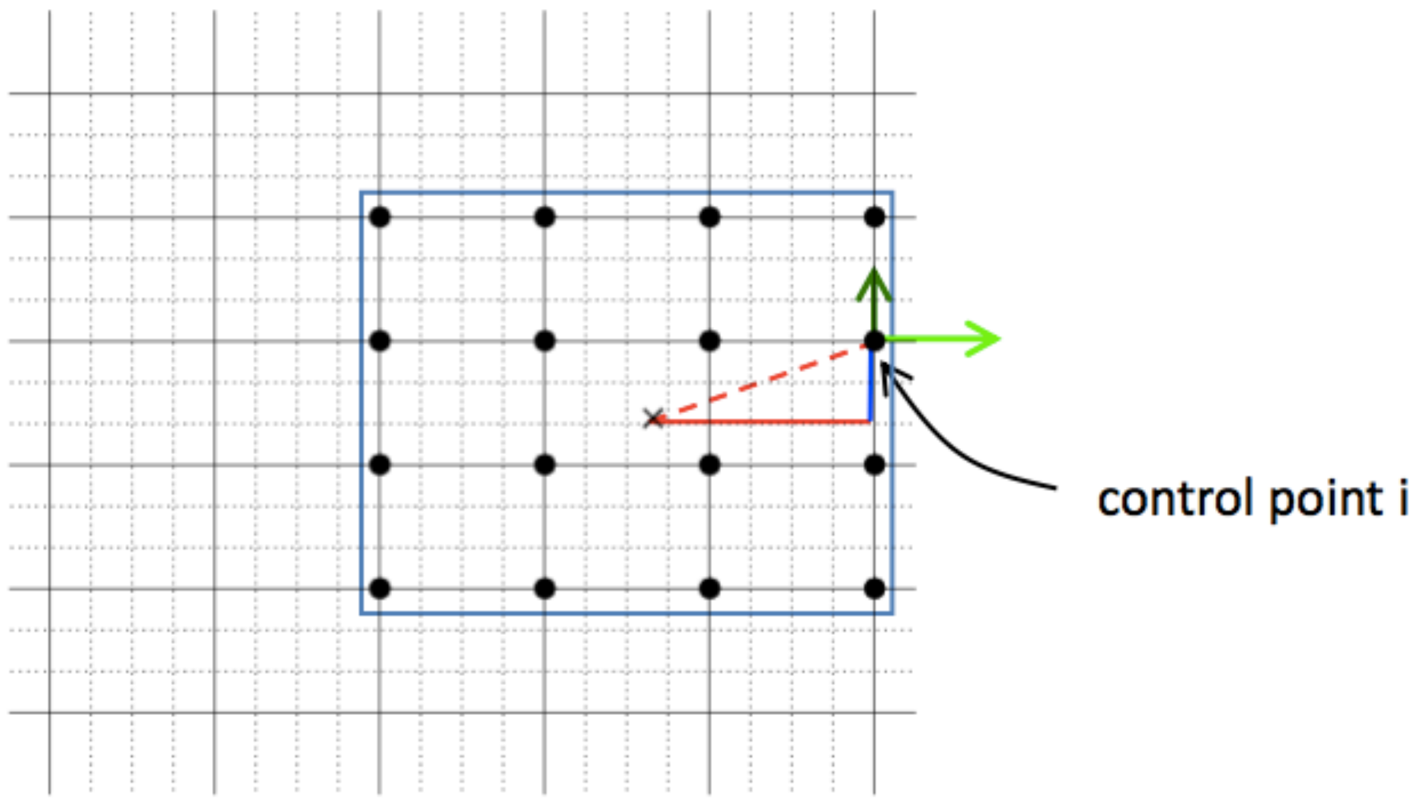
$$\boldsymbol{T}_{\boldsymbol{\mu}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1(\boldsymbol{x}) \\ u_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \sum_i \mu_{i1} \beta^3 (x_1 - y_{i1}) \, \beta^3 (x_2 - y_{i2}) \\ \sum_i \mu_{i2} \beta^3 (x_1 - y_{i1}) \, \beta^3 (x_2 - y_{i2}) \end{bmatrix}$$

- $\boldsymbol{T} \Rightarrow \boldsymbol{T}_{\boldsymbol{\mu}}$
- $\arg\min_{\boldsymbol{T}} \mathcal{C}(I_F, I_M, \boldsymbol{T}) \Rightarrow \arg\min_{\boldsymbol{\mu}} \mathcal{C}(I_F, I_M, \boldsymbol{T}_{\boldsymbol{\mu}})$

B-Splines Practically

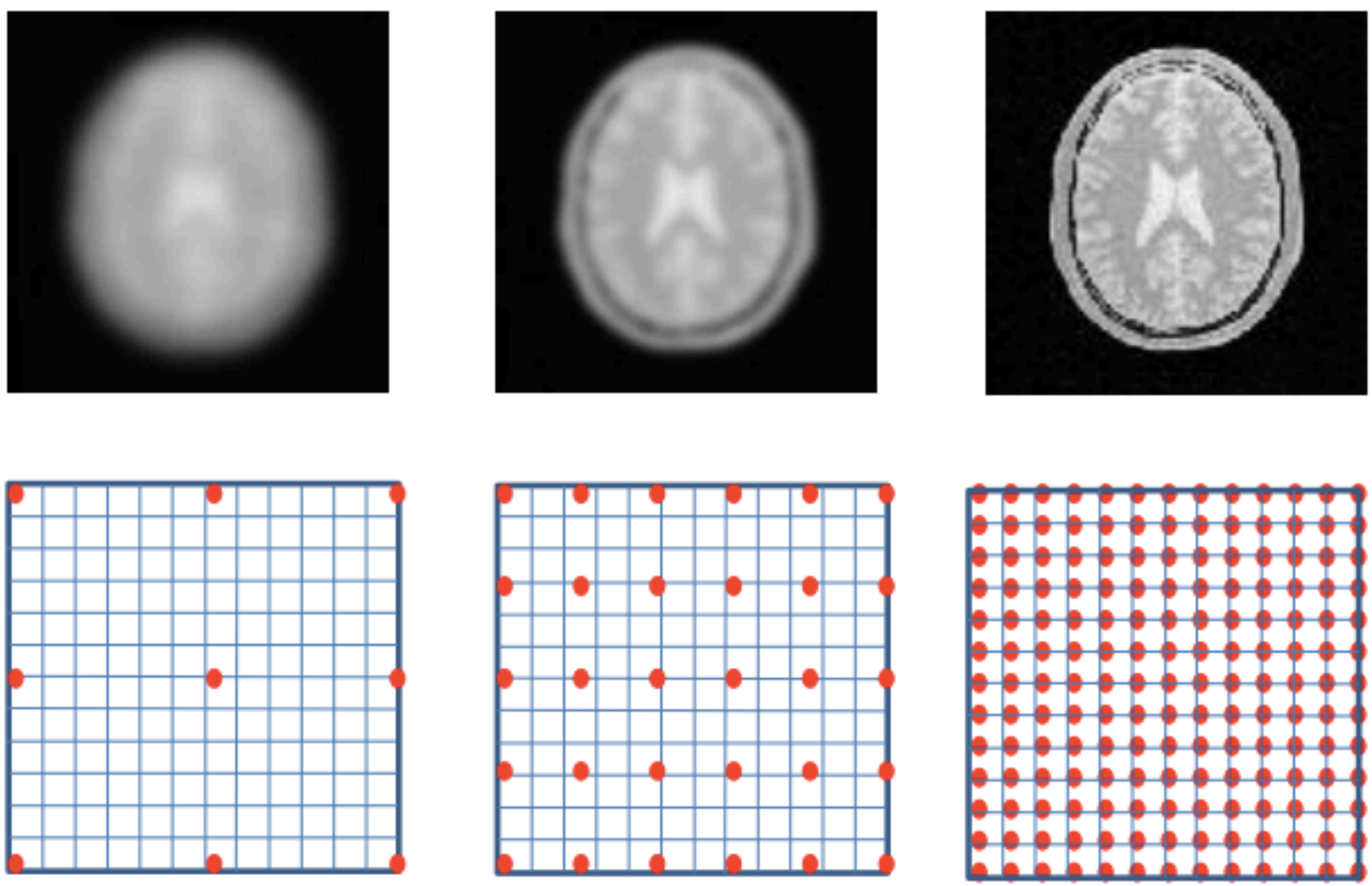
- • control point
- × world coordinate x
- □ support region S(x)

$$\boldsymbol{T}_{\boldsymbol{\mu}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \sum_i \underline{\mu_{i1}} \beta^3 \left(\underline{x_1 - y_{i1}}\right) \beta^3 \left(\underline{x_2 - y_{i2}}\right) \\ \sum_i \underline{\mu_{i2}} \beta^3 \left(\underline{x_1 - y_{i1}}\right) \beta^3 \left(\underline{x_2 - y_{i2}}\right) \end{bmatrix}$$



B-Splines Practically

- Multi-resolution: coarse to fine



Optimization

- $\arg\min_{\boldsymbol{\mu}} \mathcal{C}(I_F, I_M, \boldsymbol{T}_{\boldsymbol{\mu}}),\qquad \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - a_k \frac{\partial \mathcal{C}}{\partial \boldsymbol{\mu}_k}$
- $\frac{\partial \mathcal{C}}{\partial \boldsymbol{\mu}} = f\left(\cdots, \frac{\partial I_M}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mu}}\right)$
- $\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}) = f(\boldsymbol{\mu}, \beta^3) \quad \Rightarrow \quad \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mu}} = f(\beta^3)$
- Penalty terms: $P = f\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}, \frac{\partial^2 \boldsymbol{T}}{\partial \boldsymbol{x} \partial \boldsymbol{x}'}\right)$

$$\frac{\partial T_1}{\partial x_2}(\boldsymbol{x}) = \sum_{y_i \in \mathcal{S}(\boldsymbol{x})} \mu_{i1} \beta^3 (x_1 - y_{i1}) \boxed{\frac{\partial}{\partial x_2} \beta^3 (x_2 - y_{i2})}$$

Math-Defn.

- $F(\mathbf{x})$ = fixed image, $M(\mathbf{x})$ = moving image

  $\mathbf{x}$ = voxel coordinate

- Transformation function: $T(\mathbf{x}\,;\,\mathbf{p})$

  $\mathbf{p}$ = vector of transformation parameters

- Cost function: $C(\,\mathbf{p}\,)$

  measures similarity of fixed image $F(\mathbf{x})$ and deformed moving image $M(\,T(\mathbf{x};\,\mathbf{p})\,)$

- Find $\mathbf{p}$ that minimises C

Iterative Optimization

$$\mathbf{p}_{k+1} = \mathbf{p}_k + a_k \cdot \mathbf{d}_k$$

$\mathbf{d}_k$ = search direction

a_k = step size

gradient descent: $\mathbf{d}_k = -\frac{\partial C}{\partial \mathbf{p}}(\mathbf{p}_k) \equiv -\mathbf{g}_k$

Gradient Descent

$$p_{k+1} = p_k - a_k \cdot g_k$$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \end{bmatrix}_{k+1} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \end{bmatrix}_k - a_k \cdot \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \end{bmatrix}_k$$

$$g_1 = \left[\frac{\partial C}{\partial p_1}\right]_k$$

Cost Function Derivative

Example for mean of squared differences:

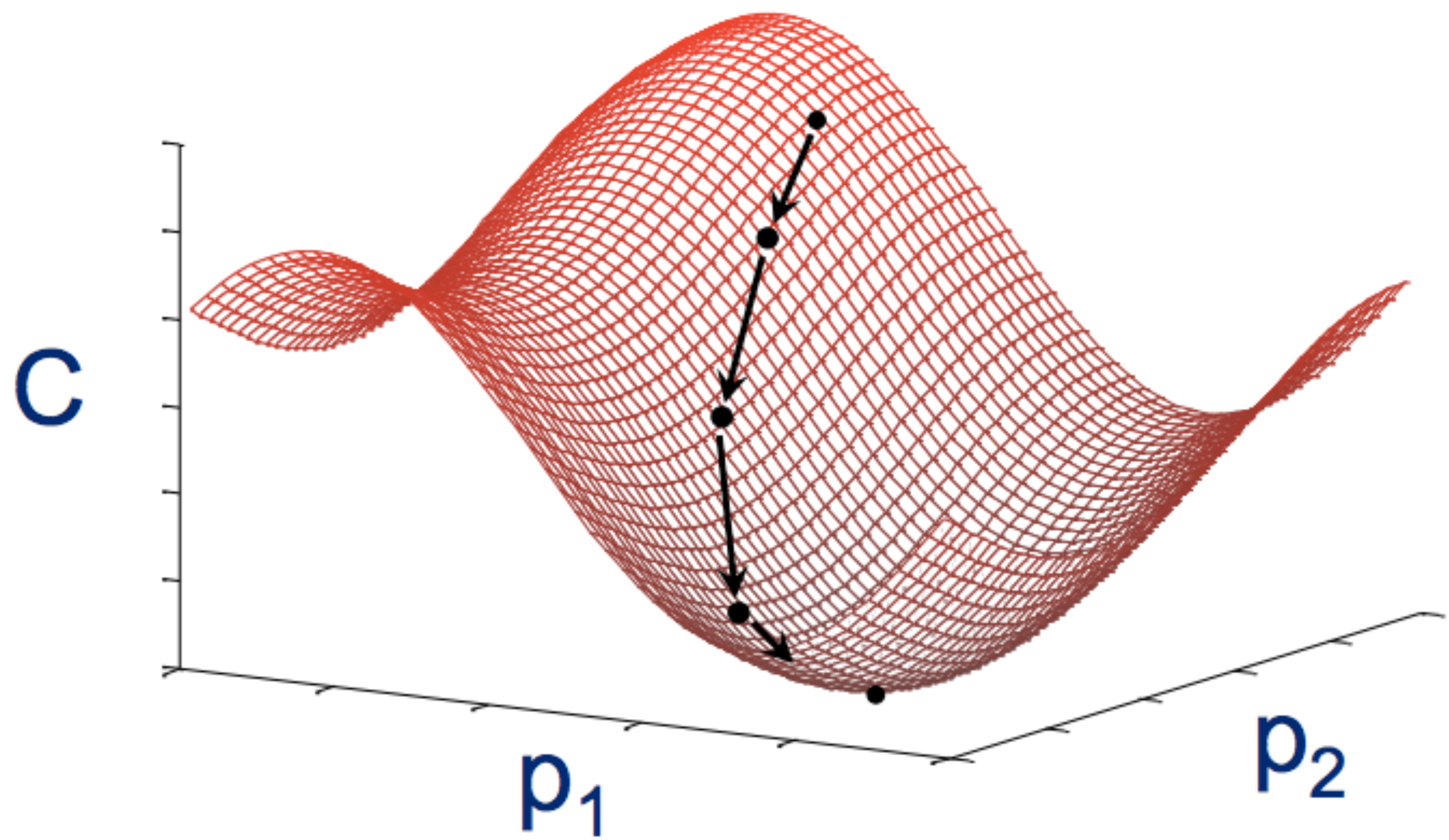
$$C(\mathbf{p}) = \frac{1}{N}\sum_{\mathbf{x}}\left(F(\mathbf{x}) - M(\mathbf{T}(\mathbf{x};\mathbf{p}))\right)^2$$

$$\frac{\partial C}{\partial \mathbf{p}} = -\frac{2}{N}\sum_{\mathbf{x}}\left(F(\mathbf{x}) - M(\mathbf{T}(\mathbf{x};\mathbf{p}))\right)\frac{\partial M}{\partial \mathbf{p}}$$

$$= -\frac{2}{N}\sum_{\mathbf{x}}\left(F(\mathbf{x}) - M(\mathbf{T}(\mathbf{x};\mathbf{p}))\right)\left(\frac{\partial \mathbf{T}}{\partial \mathbf{p}}\right)^t\frac{\partial M}{\partial \mathbf{x}}$$

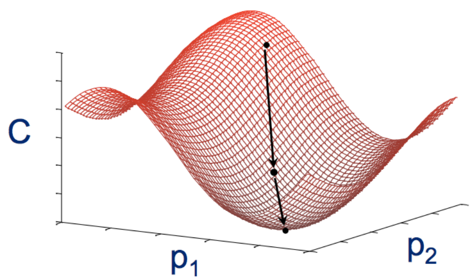
Choice of d_k

gradient descent



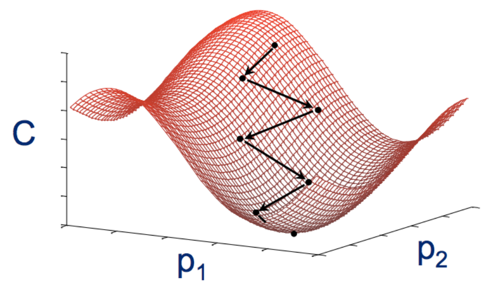
Choice of d_k

smarter steps



Choice of d_k

cheaper steps

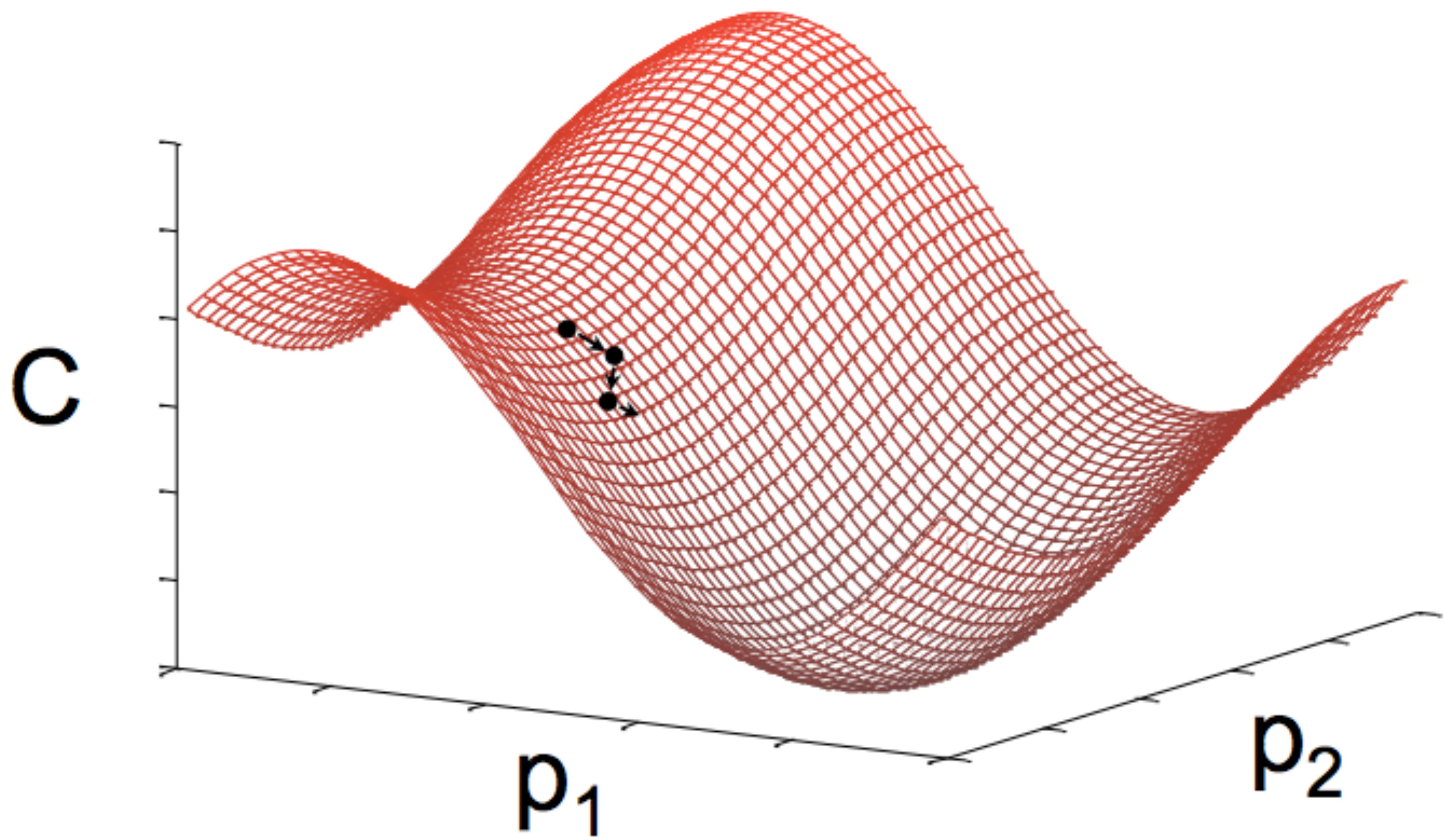


$$p_{k+1} = p_k + a_k \cdot d_k$$

gradient descent:	$d_k = -g_k$	
Newton:	$d_k = -[H_k]^{-1} g_k$	smarter steps
quasi-Newton:	$d_k = -B_k g_k$	smarter steps
conjugate gradient:	$d_k = -g_k + \beta_k d_{k-1}$	smarter steps
stochastic gradient:	$d_k \approx -g_k$	$\rightarrow$ cheaper steps

Choice of a_k

Too small steps



Choice of a_k

$$p_{k+1} = p_k + a_k \cdot d_k$$

constant: $a_k = a$

slowly decaying: $a_k = f(k) = a / (A + k)^\alpha$

exact line search: $a_k = \operatorname{argmin}_a C(p_k + a\, d_k)$

inexact line search: $a_k \approx \operatorname{argmin}_a C(p_k + a\, d_k)$ *[Wolfe conditions]*

adaptive: $a_k = F(\text{progress in previous iterations})$

Diffeomorphic Image Registration

- At its simplest, image registration involves estimating a smooth, continuous mapping between the points in one image and those in another.

Diffeomorphic Image Registration

- At its simplest, image registration involves estimating a smooth, continuous mapping between the points in one image and those in another.
- The relative shapes of the images can then be determined from the parameters that encode the mapping.

Diffeomorphic Image Registration

- At its simplest, image registration involves estimating a smooth, continuous mapping between the points in one image and those in another.
- The relative shapes of the images can then be determined from the parameters that encode the mapping.
- **The objective is** usually to determine the single "best" set of values for these parameters

Diffeomorphic Image Registration

- At its simplest, image registration involves estimating a smooth, continuous mapping between the points in one image and those in another.
- The relative shapes of the images can then be determined from the parameters that encode the mapping.
- **The objective is** usually to determine the single "best" set of values for these parameters
  - The small-deformation framework does not necessarily preserve topology—although if the deformations are relatively small, then it may still be preserved.
  - The large-deformation framework generates deformations (diffeomorphisms) that have a number of elegant mathematical properties, such as enforcing the preservation of topology.

Deformation Model

- Many registration algorithm still use small displacement models (u), which is simply added into identity transform (x):

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$$

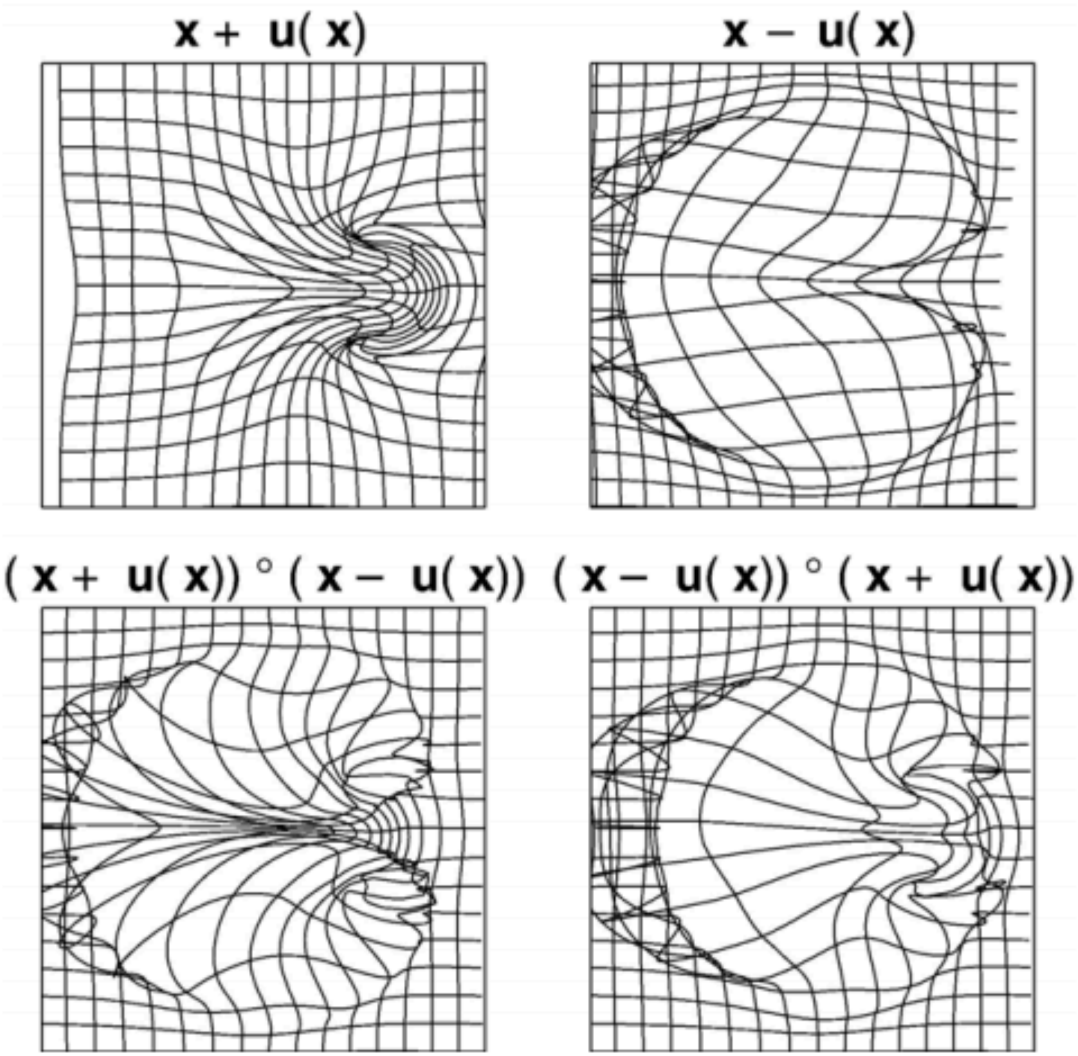
Deformation Model

- Many registration algorithm still use small displacement models (u), which is simply added into identity transform (x):

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$$

- In such parameterizations, the inverse transformation is sometimes approximated by subtracting the displacement. It is worth noting that this is only a very approximate inverse, which fails badly for larger deformations.

Deformation Model



Small deformation models do not necessarily enforce a one-to-one mapping.

If the inverse transformation is correct, these two should be identity!

Diffeomorphic Registration

- **The large-deformation** or **diffeomorphic setting** is a much more elegant framework. A diffeomorphism is a globally one-to-one (objective) smooth and continuous mapping with derivatives that are invertible.

Diffeomorphic Registration

- **The large-deformation** or **diffeomorphic setting** is a much more elegant framework. A diffeomorphism is a globally one-to-one (objective) smooth and continuous mapping with derivatives that are invertible.
- If the mapping is not diffeomorphic, then topology is not necessarily preserved.

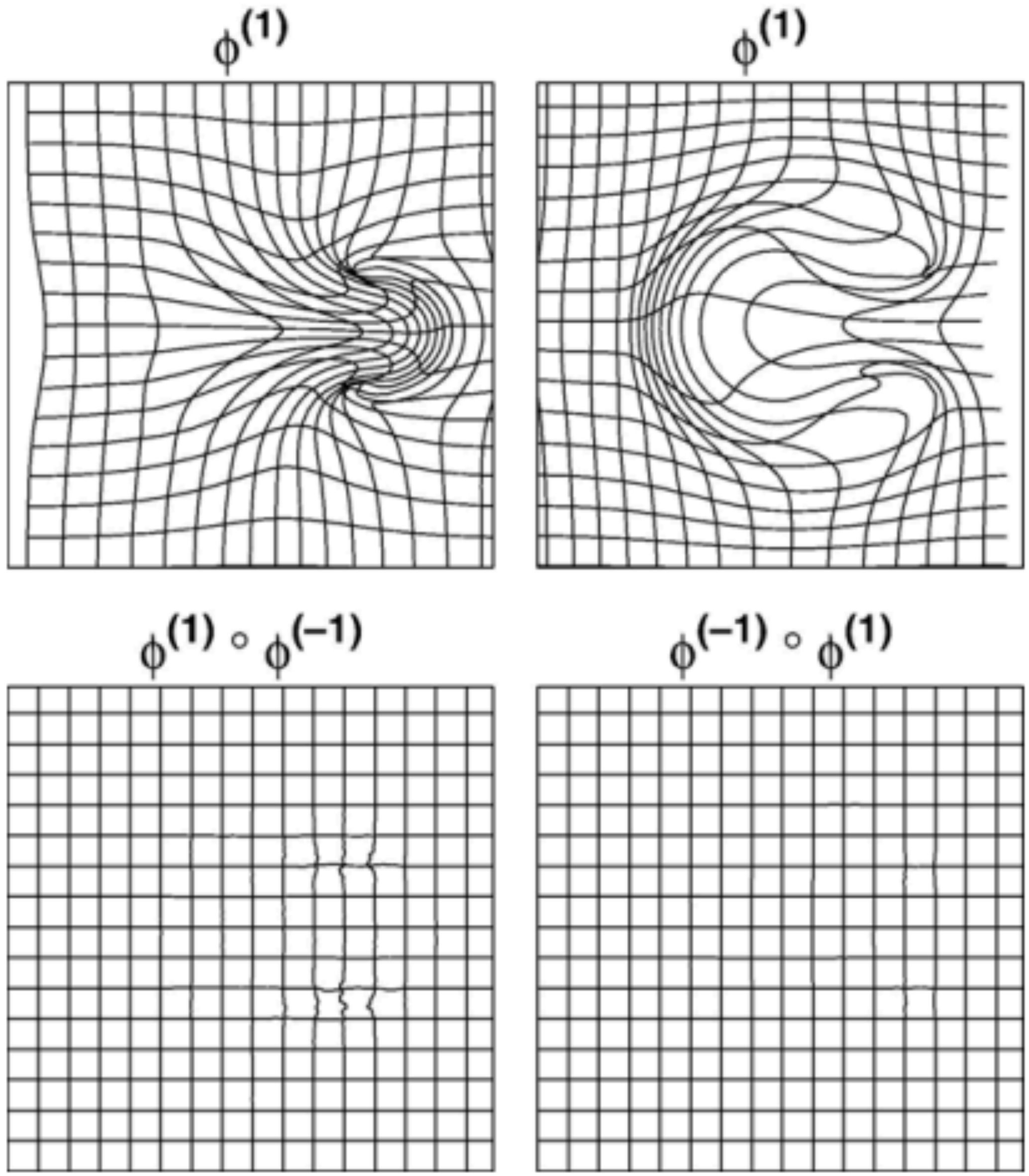
Diffeomorphic Registration

- **The large-deformation** or **diffeomorphic setting** is a much more elegant framework. A diffeomorphism is a globally one-to-one (objective) smooth and continuous mapping with derivatives that are invertible.
- If the mapping is not diffeomorphic, then topology is not necessarily preserved.
- $$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}t} = \boldsymbol{u}^{(t)}\left(\boldsymbol{\Phi}^{(t)}\right)$$ it is easier to parameterize using a number corresponding to different time periods over the course of the evolution of the diffeomorphism (consider $u^{(t)}$ as a velocity field at time t

Diffeomorphic Registration

- **The large-deformation** or **diffeomorphic setting** is a much more elegant framework. A diffeomorphism is a globally one-to-one (objective) smooth and continuous mapping with derivatives that are invertible.
- If the mapping is not diffeomorphic, then topology is not necessarily preserved.
- $\frac{d\Phi}{dt} = u^{(t)}\left(\Phi^{(t)}\right)$ it is easier to parameterize using a number corresponding to different time periods over the course of the evolution of the diffeomorphism (consider $u^{(t)}$ as a velocity field at time t
- Diffeomorphisms are generated by initializing with an identity transform $(\Phi^{(0)} = x)$ and integrating over unit time to obtain $\Phi^{(1)}$.

Forward & Inverse Transform



DARTEL: A Fast Diffeomorphic Method (Ashburner, NeuroImage 2007)

Diffeomorphic

Anatomical

Registration

Through

Exponentiated

Lie Algebra

DARTEL: A Fast Diffeomorphic Method (Ashburner, NeuroImage 2007)

- The DARTEL model assumes a flow field (u) that remains constant over time. With this model, the differential equation describing the evolution of a deformation is

$$\frac{d\boldsymbol{\Phi}}{dt} = \boldsymbol{u}\left(\boldsymbol{\Phi}^{(t)}\right)$$

DARTEL: A Fast Diffeomorphic Method (Ashburner, NeuroImage 2007)

- The DARTEL model assumes a flow field (u) that remains constant over time. With this model, the differential equation describing the evolution of a deformation is

$$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}t} = \boldsymbol{u}\left(\boldsymbol{\Phi}^{(t)}\right)$$

- The Euler method is a simple integration approach to extend from identity (initial) transform, which involves computing new solutions after many successive small time-steps (h).

$$\boldsymbol{\Phi}^{(t+h)} = \boldsymbol{\Phi}^{(t)} + h\boldsymbol{u}(\boldsymbol{\Phi}^{(t)})$$

DARTEL: A Fast Diffeomorphic Method (Ashburner, NeuroImage 2007)

- The DARTEL model assumes a flow field (u) that remains constant over time. With this model, the differential equation describing the evolution of a deformation is

$$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}t} = \boldsymbol{u}\left(\boldsymbol{\Phi}^{(t)}\right)$$

- The Euler method is a simple integration approach to extend from identity (initial) transform, which involves computing new solutions after many successive small time-steps (h).

$$\boldsymbol{\Phi}^{(t+h)} = \boldsymbol{\Phi}^{(t)} + h\boldsymbol{u}(\boldsymbol{\Phi}^{(t)})$$

- Each of these Euler steps is equivalent to

$$\boldsymbol{\Phi}^{(t+h)} = (\boldsymbol{x} + h\boldsymbol{u})\circ\boldsymbol{\Phi}^{(t)}$$

DARTEL: A Fast Diffeomorphic Method (Ashburner, NeuroImage 2007)

- The use of a large number of small time steps will produce a more accurate solution, for instance (8 steps)

$$
\begin{aligned}
\boldsymbol{\Phi}^{(1/8)} &= \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})/8 \\
\boldsymbol{\Phi}^{(2/8)} &= \boldsymbol{\Phi}^{(1/8)} \circ \boldsymbol{\Phi}^{(1/8)} \\
\boldsymbol{\Phi}^{(3/8)} &= \boldsymbol{\Phi}^{(1/8)} \circ \boldsymbol{\Phi}^{(2/8)} \\
&\vdots \\
\boldsymbol{\Phi}^{(8/8)} &= \boldsymbol{\Phi}^{(1/8)} \circ \boldsymbol{\Phi}^{(7/8)}
\end{aligned}
$$

Optimization of DARTEL

- Image registration procedures use a mathematical model to explain the data. Such a model will contain a number of unknown parameters that describe how an image is deformed.

Optimization of DARTEL

- Image registration procedures use a mathematical model to explain the data. Such a model will contain a number of unknown parameters that describe how an image is deformed.
- A true diffeomorphism has an infinite number of dimensions and is infinitely differential.

Optimization of DARTEL

- Image registration procedures use a mathematical model to explain the data. Such a model will contain a number of unknown parameters that describe how an image is deformed.
- A true diffeomorphism has an infinite number of dimensions and is infinitely differential.
- The discrete parameterization of the velocity field, u(x), can be considered as a linear combination of basis functions.

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_i v_i \boldsymbol{\rho}_i(\boldsymbol{x})$$

v is a vector of coefficients

ϱi(x) is the ith first degree B-spline basis function at position x

Optimization of DARTEL

- The aim is to estimate the single “best” set of values for these parameters (v). The objective function, which is the measure of “goodness”, is formulated as the most probable deformation, given the data (D).

$$p(\mathbf{v}|D) = \frac{p(D|\mathbf{v})p(\mathbf{v})}{p(D)}$$

Optimization of DARTEL

- The aim is to estimate the single “best” set of values for these parameters (v). The objective function, which is the measure of “goodness”, is formulated as the most probable deformation, given the data (D).

$$p(\boldsymbol{v}|D) = \frac{p(D|\boldsymbol{v})p(\boldsymbol{v})}{p(D)}$$

- The objective is to find the most probable parameter values and not the actual probability density, so this factor is ignored. The single most probable estimate of the parameters is known as the maximum a posteriori (MAP) estimate.

$$-\log p(\boldsymbol{v}, D) = -\log p(\boldsymbol{v}) - \log p(D|\boldsymbol{v})$$

Optimization of DARTEL

- The aim is to estimate the single "best" set of values for these parameters (v). The objective function, which is the measure of "goodness", is formulated as the most probable deformation, given the data (D).

$$p(\mathbf{v}|D) = \frac{p(D|\mathbf{v})p(\mathbf{v})}{p(D)}$$

- The objective is to find the most probable parameter values and not the actual probability density, so this factor is ignored. The single most probable estimate of the parameters is known as the maximum a posteriori (MAP) estimate.

$$-\log p(\mathbf{v}, D) = -\log p(\mathbf{v}) - \log p(D|\mathbf{v})$$

Or

$$\mathcal{E}(\mathbf{v}) = \mathcal{E}_1(\mathbf{v}) + \mathcal{E}_2(\mathbf{v})$$

Optimization of DARTEL

- Many nonlinear registration approaches search for a maximum a posteriori (MAP) estimate of the parameters defining the warps, which corresponds to the mode of the probability density

Optimization of DARTEL

- Many nonlinear registration approaches search for a maximum a posteriori (MAP) estimate of the parameters defining the warps, which corresponds to the mode of the probability density
- The Levenberg–Marquardt (LM) algorithm is a very good general purpose optimization strategy

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} - \left(\left.\frac{\partial^2 \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}^2}\right|_{\boldsymbol{v}^{(n)}} + \zeta \mathbf{I} \right)^{-1} \left.\frac{\partial \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}^{(n)}}$$

Optimization of DARTEL

- Many nonlinear registration approaches search for a maximum a posteriori (MAP) estimate of the parameters defining the warps, which corresponds to the mode of the probability density
- The Levenberg–Marquardt (LM) algorithm is a very good general purpose optimization strategy

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} - \left(\left.\frac{\partial^2 \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}^2}\right|_{\boldsymbol{v}^{(n)}} + \zeta \mathbf{I} \right)^{-1} \left.\frac{\partial \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}^{(n)}}$$

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} - (\mathbf{A} + \mathbf{H} + \zeta \mathbf{I})^{-1} (\boldsymbol{b} + \mathbf{H}\boldsymbol{v}^{(n)})$$

A: second order tensor field, H: concentration matrix, b: first derivative of likelihood func.

Optimization of DARTEL

Simultaneously minimize the sum of

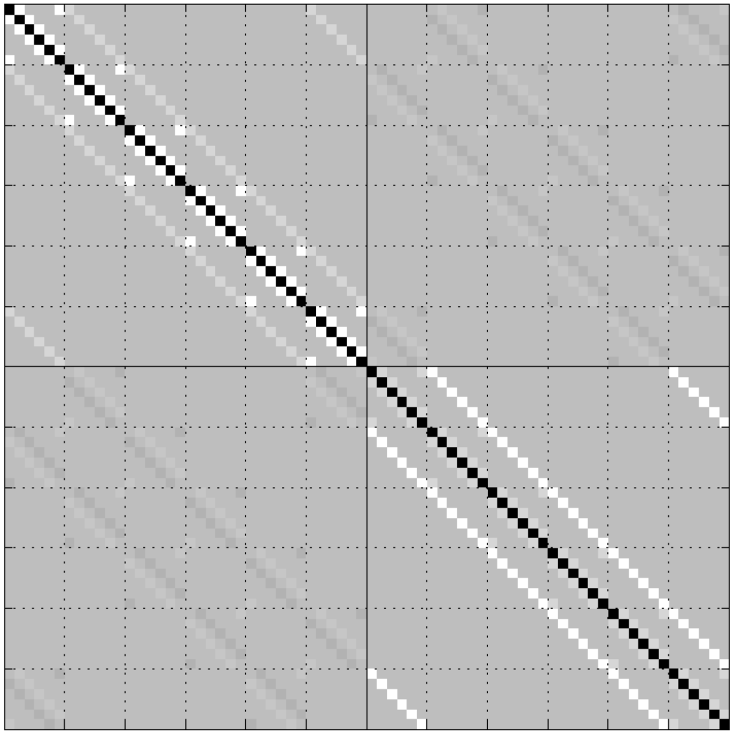
- Likelihood component
  - Sum of squares difference
  - $\frac{1}{2} \sum_i \sum_k \left(t_k(\mathbf{x}_i) - \mu_k(\boldsymbol{\phi}^{(1)}(\mathbf{x}_i))\right)^2$
  - $\boldsymbol{\phi}^{(1)}$ parameterized by $\mathbf{u}$
- Prior component
  - A measure of deformation roughness
  - $\frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u}$

Optimization of DARTEL

PRIOR TERM

- $\frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u}$
- DARTEL has three different models for **H**
  - Membrane energy
  - Linear elasticity
  - Bending energy
- **H** is very sparse
- H: deformation roughness

An example **H** for 2D registration of 6x6 images (linear elasticity)



Optimization of DARTEL

LIKELIHOOD TERM

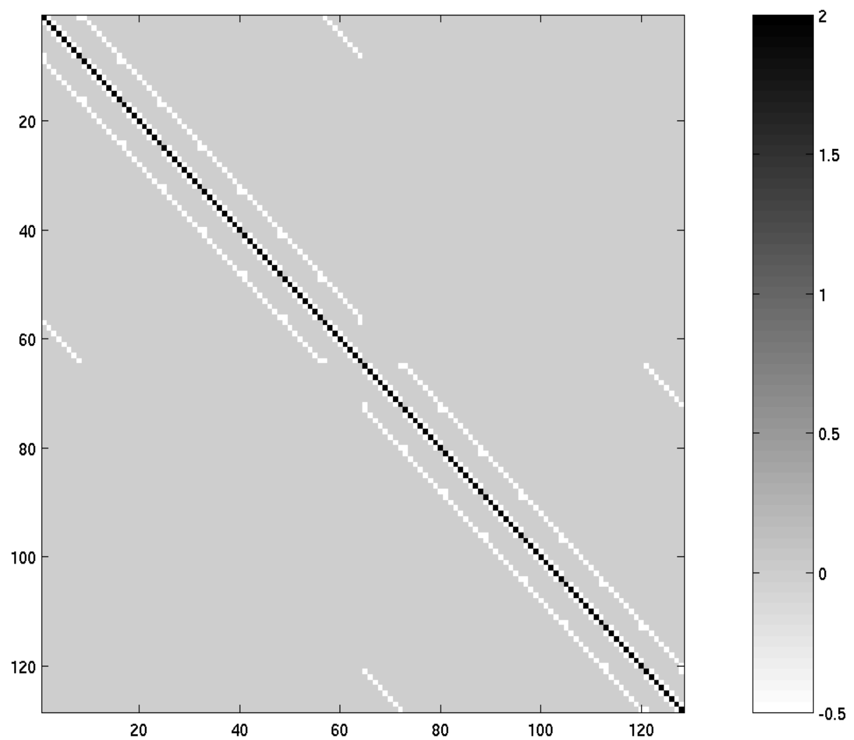
- Images assumed to be partitioned into different tissue classes.
  - E.g., a 3 class registration simultaneously matches:
    - Grey matter with grey matter
    - White matter wit white matter
    - Background (1 – GM – WM) with background

"Membrane Energy"

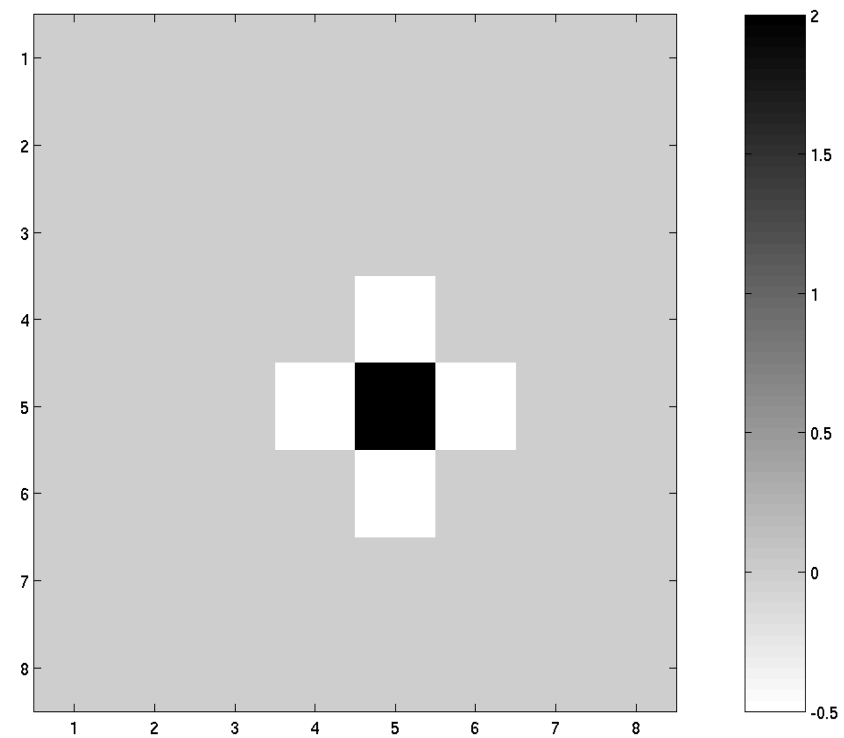
Penalizes first derivatives.

Sum of squares of the elements of the Jacobian (matrices) of the flow field.

Sparse Matrix Representation



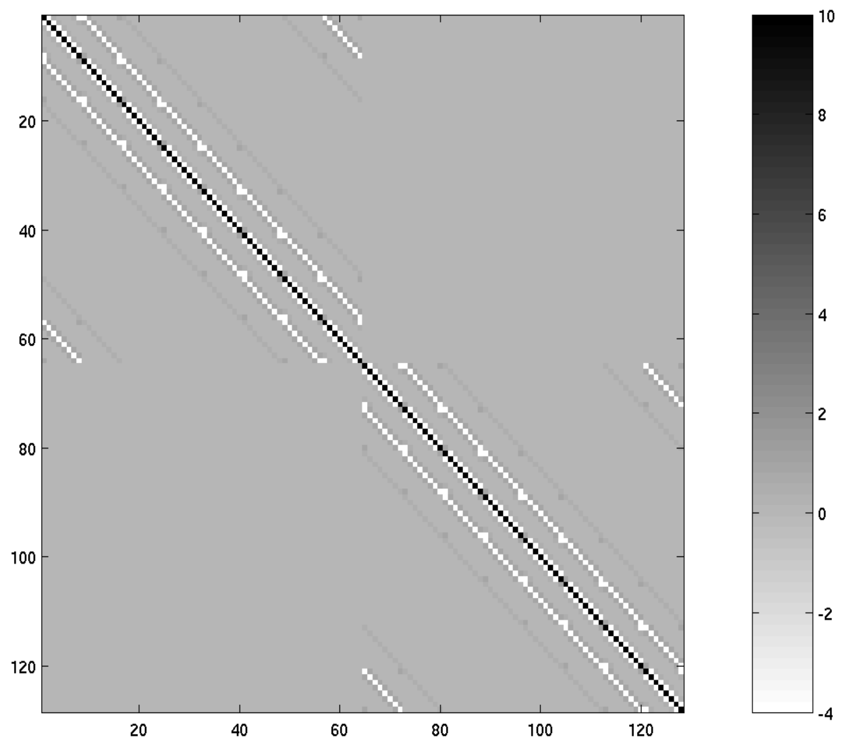
Convolution Kernel



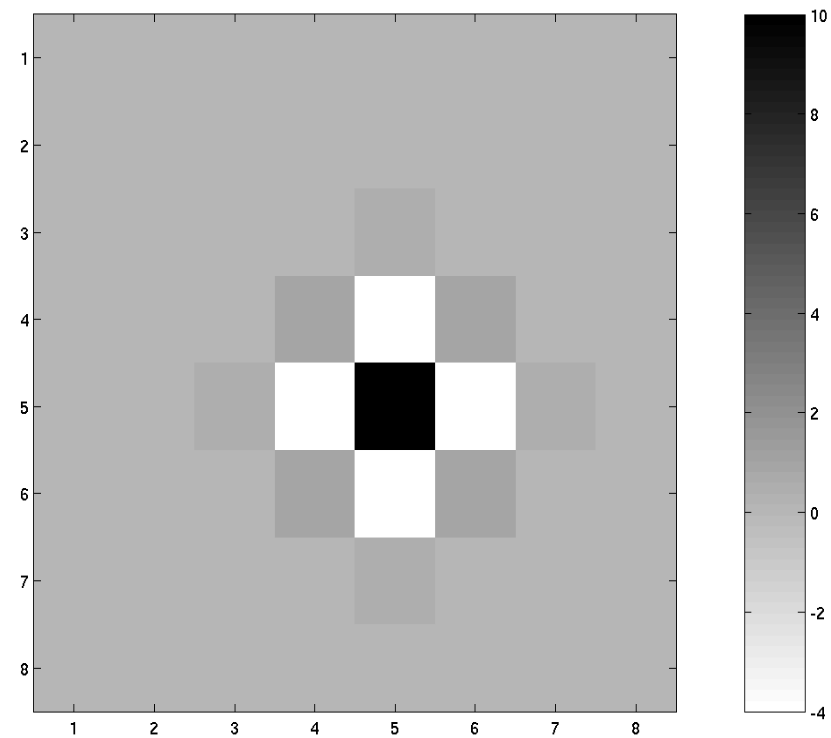
"Bending Energy"

Penalizes second derivatives.

Sparse Matrix Representation

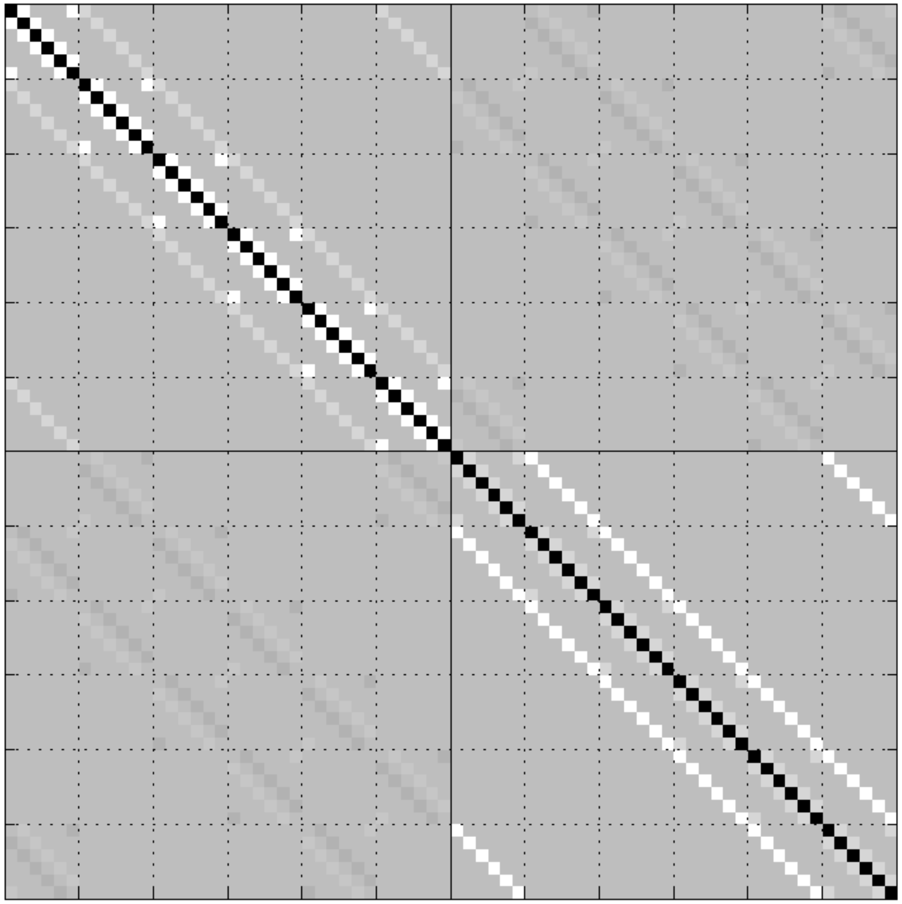


Convolution Kernel



"Linear Elasticity"

- Decompose the Jacobian of the flow field into
  - Symmetric component
    - ½(**J**+**J**^{T})
    - Encodes non-rigid part.
  - Anti-symmetric component
    - ½(**J**-**J**^{T})
    - Encodes rigid-body part.
- Penalise sum of squares of symmetric part.
- Trace of Jacobian encodes volume changes. Also penalized.



Gauss-Newton Optimization

- Uses Gauss-Newton
  - Requires a matrix solution to a very large set of equations at each iteration

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - (\mathbf{H}+\mathbf{A})^{-1}\,\mathbf{b}$$

  - **b** are the first derivatives of objective function
  - **A** is a sparse matrix of second derivatives
  - Computed efficiently, making use of scaling and squaring

Summary

- Deformable Image Registration
  - B-spline parametrization and Free Form Deformation
- Optimization
- Diffeomorphic Image Registration

Slide Credits and References

- *Darko Zikic, MICCAI 2010 Tutorial*
- *Stefan Klein, MICCAI 2010 Tutorial*
- *Marius Staring, MICCAI 2010 Tutorial*
- *J. Ashburner, NeuroImage 2007.*
- M. F. Beg, M. I. Miller, A. Trouvé and L. Younes. "*Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms*". International Journal of Computer Vision 61(2):139–157 (2005).
- M. Vaillant, M. I. Miller, L. Younes and A. Trouvé. "*Statistics on diffeomorphisms via tangent space representations*". NeuroImage 23:S161–S169 (2004).
- L. Younes, "*Jacobi fields in groups of diffeomorphisms and applications*". Quart. Appl. Math. 65:113–134 (2007).