

Basic Mathematical Tools for Imaging and Visualization

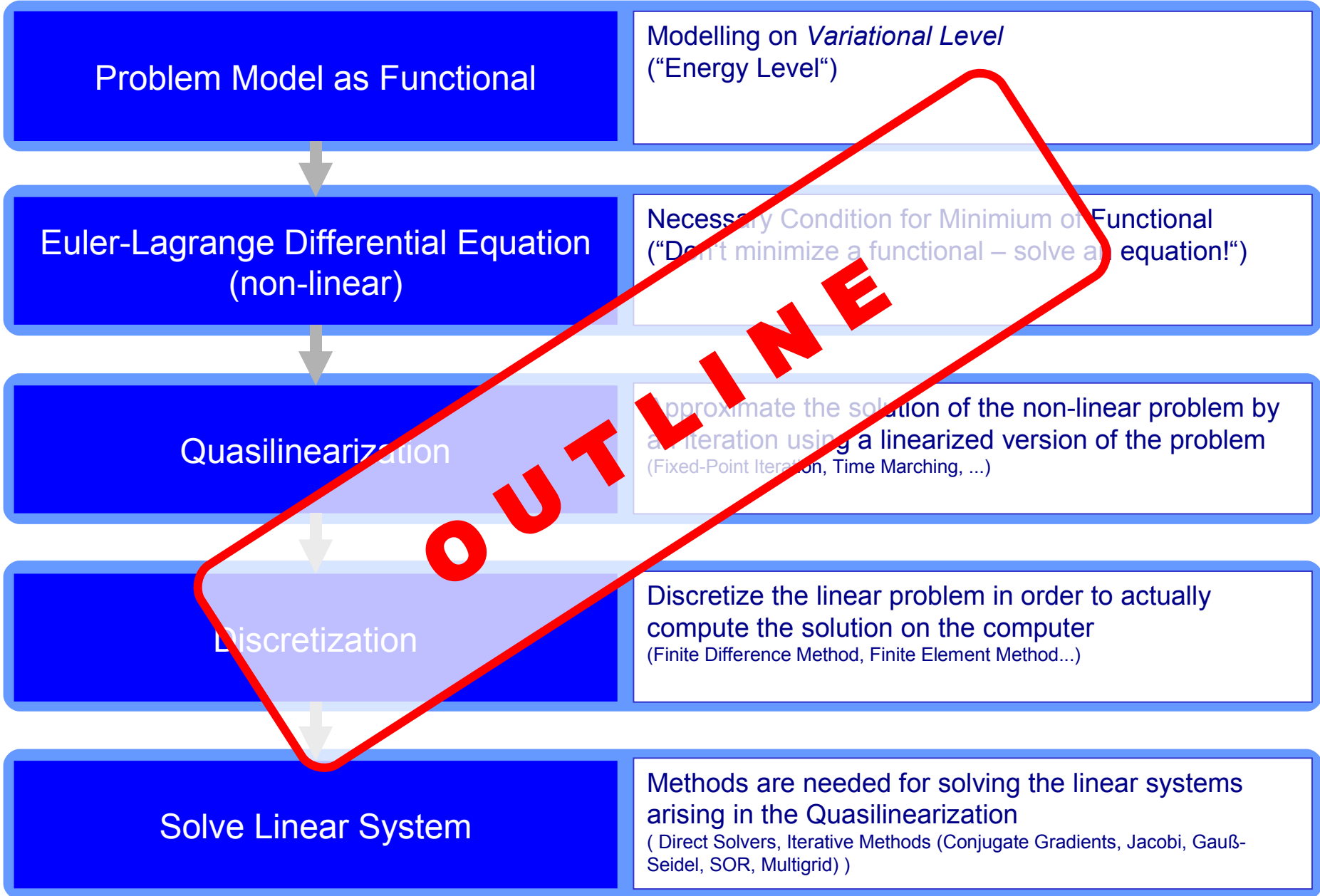
Minimization of Functionals Case Study: Deformable Registration

Nassir Navab

2 February 2007

chair for computer aided medical procedures and augmented reality

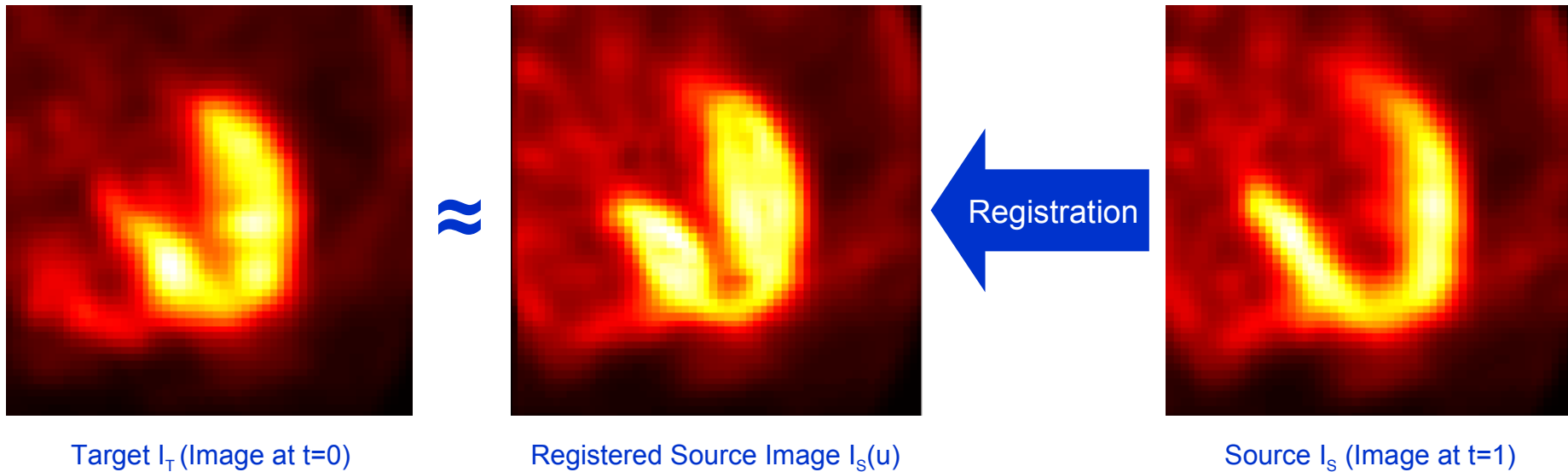
department of computer science | technische universität münchen



OUTLINE

intro

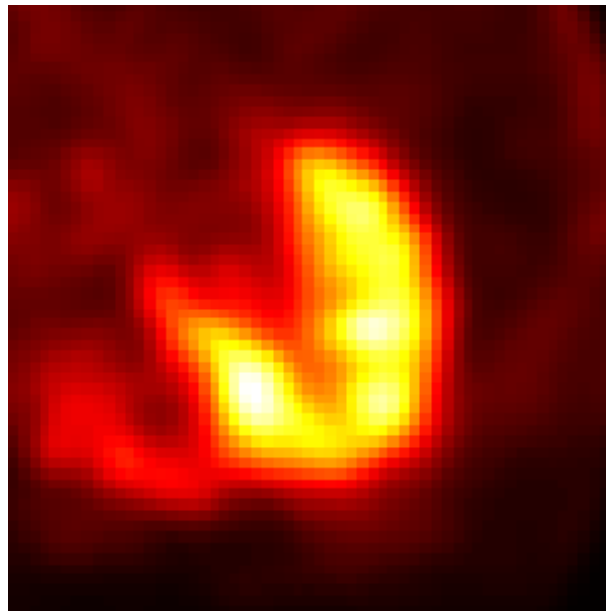
Deformable Registration



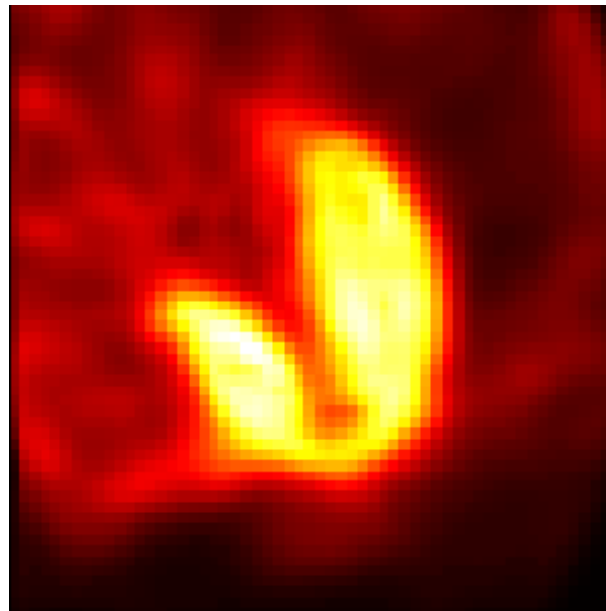
Deformable Registration

Undo deformation of an image, such that it resembles the previous image

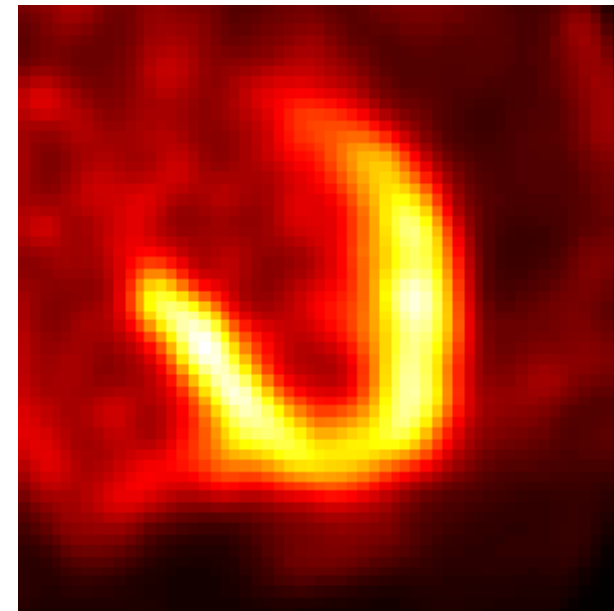
Example: Deformable Registration of Gated PET Data



Target Image



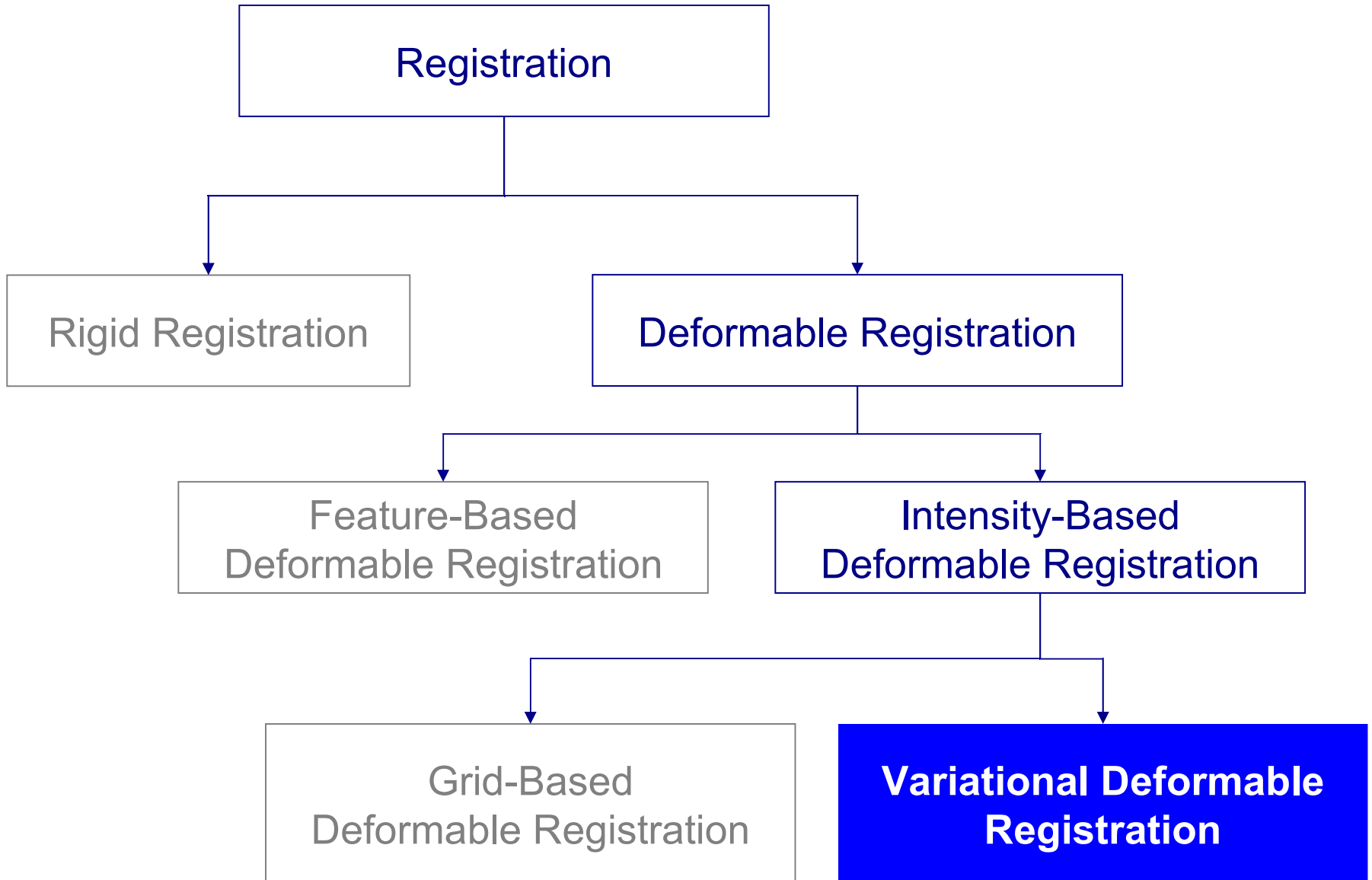
Deformed Source Image



Source Image

Registration





main

Problem Model as Functional

Modelling on *Variational Level*
("Energy Level")

Euler-Lagrange Differential Equation
(non-linear)

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

Approximate the solution of the non-linear problem by
an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually
compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems
arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-
Seidel, SOR, Multigrid))

'General' Optimization Approach

- Assign certain value/cost to every possible configuration such that this cost is minimal for the 'true' configuration
- Value = Cost = "Error"

- However, measuring only the Error gives bad algorithms

→ ILL-POSEDNESS

- Remedy: **Regularization**

Well-Posedness (Hadamard):

- a solution exists
- solution is unique
- problem is well-conditioned:
 - = errors in output depend continuously on errors in input
 - = small input errors do not cause large errors in result

Regularization

- Making the problem mathematically well-posed
- Encoding a-priori knowledge about the problem

Deformable Registration is Ill-Posed

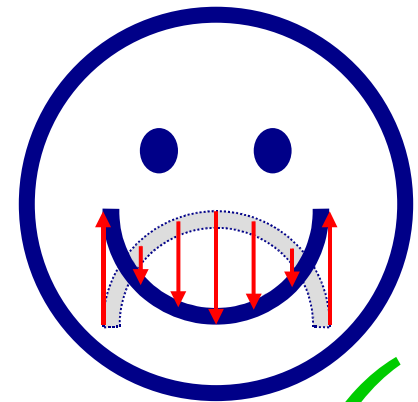
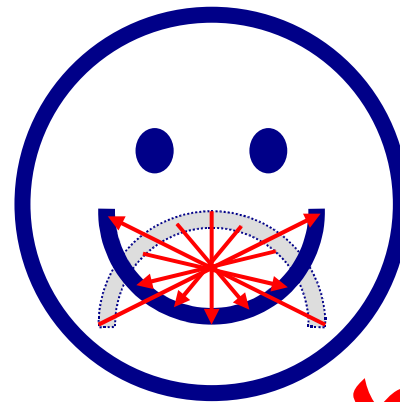
- Deformable registration is ill-posed in the sense of Hadamard
 - no unique solution
 - bad condition: small errors/changes in measurements can lead to large changes in the results (\rightarrow noise)



Source

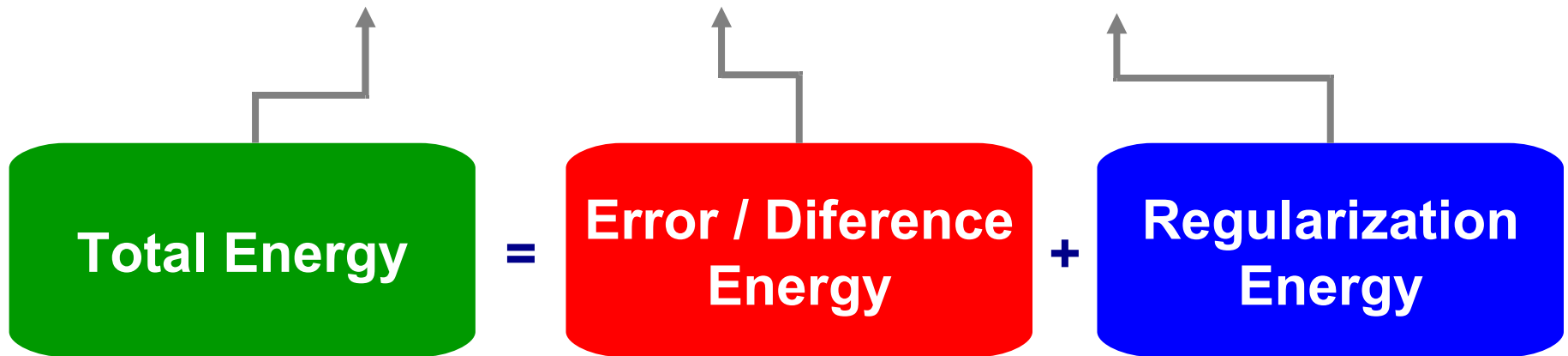


Target



Typical Image Processing Problem

$$J = D + \alpha S$$



Model for Deformable Registration

$$\mathcal{J}[I_T, I_S, \varphi] = \mathcal{D}[I_T, I_S, \varphi] + \alpha \mathcal{S}[\varphi] \rightarrow \min!$$

input images

$$I_T, I_S : \Omega \rightarrow \mathbb{R}, \\ \Omega = [0, 1]^d \subset \mathbb{R}^d$$

deformation function

$$\varphi : \Omega \rightarrow \Omega$$

Regularizing Term, for example:

- Diffusion Regularization
- Curvature Regularization
- Linear-Elasticity

Difference Measure - here we can use:

- Sum of Squared Differences
- Sum of Absolute Differences
- Cross-Correlation
- Correlation Ratio
- Mutual Information

Definitions and Notation

- Domain $\Omega = [0, 1]^d \subset \mathbb{R}^d$
- Images $I = \Omega \rightarrow \mathbb{R}$
- Deformation Function $\varphi = \Omega \rightarrow \Omega$, $\varphi = \text{Id} + u$
- Displacement $u = \Omega \rightarrow \Omega$

Model for Deformable Registration

$$\mathcal{J}[I_T, I_S, \varphi] = \mathcal{D}[\varphi] + \alpha \mathcal{S}[\varphi] \rightarrow \min!$$

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx$$

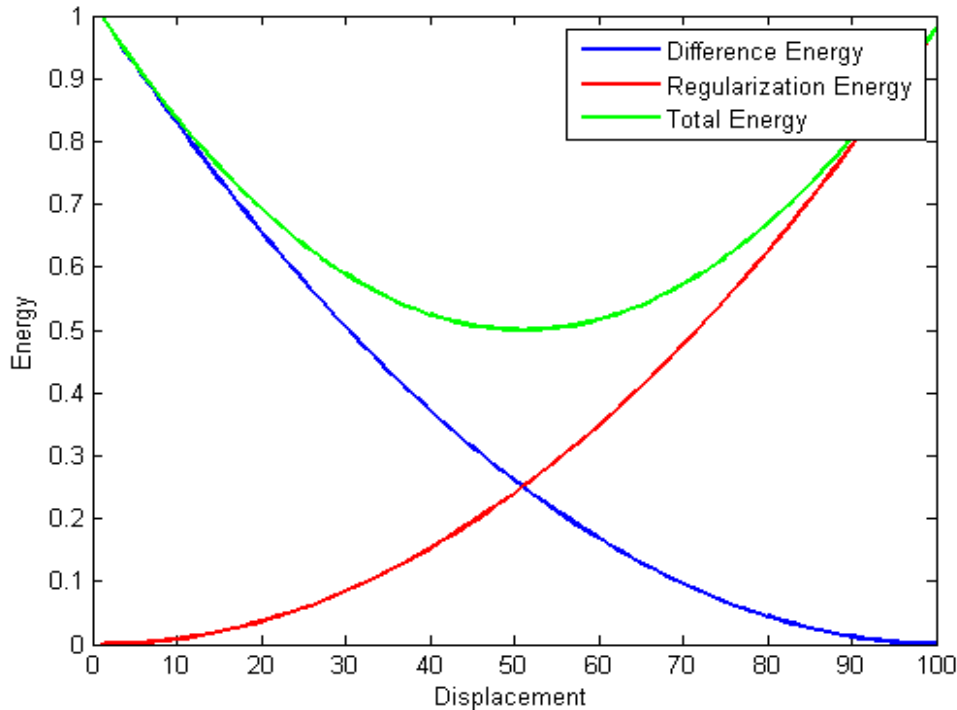
- The good old SSD

$$J[\varphi] = \int_{\Omega} \|\nabla \varphi(x)\|^2 dx$$

- Diffusion Regularization

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla \varphi(x)\|^2 dx$$

Intuition to the Problem Model



Solution as a compromise between

- Reduction of difference between images

$$\mathcal{D}[I_T, I_S, \varphi]$$

- Penalty for large deformations

$$\mathcal{S}[\varphi]$$

$$\mathcal{J}[I_T, I_S, \varphi] = \mathcal{D}[I_T, I_S, \varphi] + \alpha \mathcal{S}[\varphi]$$

Problem Model as Functional

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Modelling on *Variational Level*
("Energy Level")

Euler-Lagrange Differential Equation (non-linear)

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

Approximate the solution of the non-linear problem by an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-Seidel, SOR, Multigrid))

Euler-Lagrange Equations for Standard Problem

- Model Functional
for $y : \mathbb{R} \rightarrow \mathbb{R}$

$$I[y] = \int_{\Omega} F(x, y, y_x) dx$$

- Euler-Lagrange Eq.:

$$0 = \frac{\partial F}{\partial y} - \frac{d}{dx} \frac{\partial F}{\partial y_x}$$

From Deformable Model to corresponding PDE

▫ Model Functional:
$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Functional

corresponding Euler-Lagrange Equation

$$\min_{\varphi} \int_{\Omega} (I_1(x) - I_2(\varphi(x)))^2 dx \Rightarrow 0 = -2 \cdot (I_1(x) - I_2(\varphi(x))) \nabla I_2(\varphi(x))$$

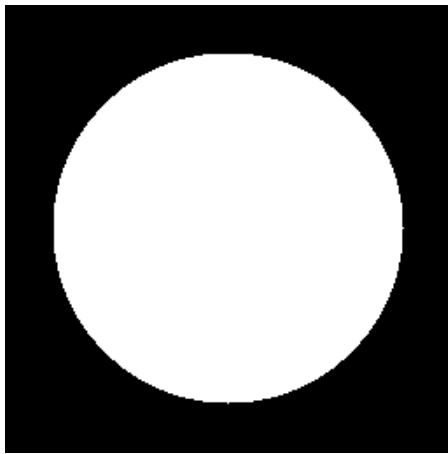
$$\min_{\varphi} \int_{\Omega} \|\nabla\varphi(x)\|^2 dx \Rightarrow 0 = \Delta\varphi(x)$$

▫ PDE

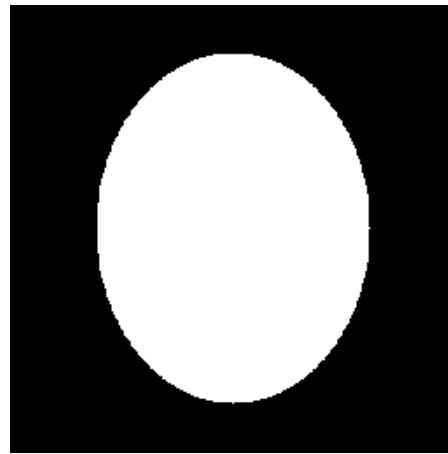
$$\Delta\varphi(x) = -2 \cdot (I_1(x) - I_2(\varphi(x))) \nabla I_2(\varphi(x))$$

$$A(\varphi) = f(\varphi)$$

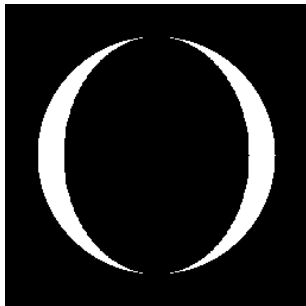
Difference-Term Euler-Lagrange Equations as *Forces*



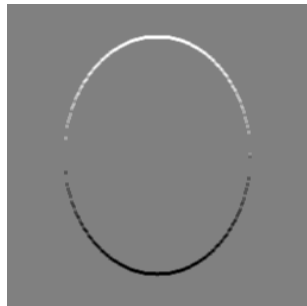
Target I_T



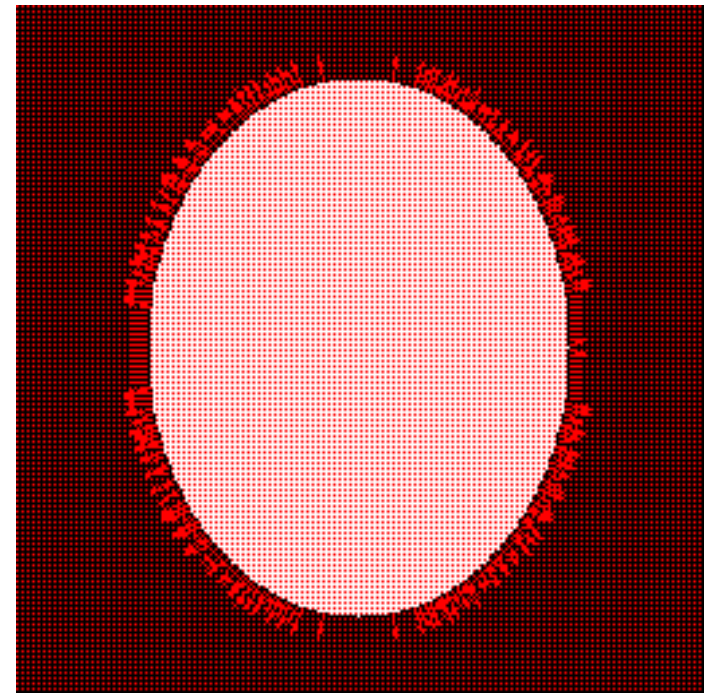
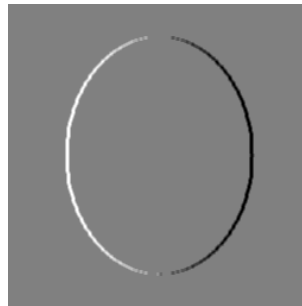
Source I_S



$(I_T - I_S(\varphi))$



$\nabla I_S(\varphi)$



$f_{SSD}(I_T, I_S, \varphi)$

$$-(I_T - I_S(\varphi)) \nabla I_S(\varphi) = f_{SSD}(I_T, I_S, \varphi)$$

Problem Model as Functional

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Modelling on *Variational Level*
("Energy Level")

Non-linear Euler-Lagrange PDE

$$\begin{aligned}\Delta\varphi(x) &= -2 \cdot (I_1(x) - I_2(\varphi(x)))\nabla I_2(\varphi(x)) \\ A(\varphi) &= f(\varphi)\end{aligned}$$

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

Approximate the solution of the non-linear problem by an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-Seidel, SOR, Multigrid))

Quasilinearization

- Approximate solution to $u = A^{-1}(f(u))$ by:

```
u(0) = u0 ; // initialize
i = 0 ;
until (convergence) // iterate
    u(i+1) = A-1(f(u(i))) ;
    i++ ;
end
```

Standard Fix-Point Iteration

$$u^{(i+1)} = A^{-1}(f(u^{(i)}))$$

$$A(u^{(i+1)}) = f(u^{(i)})$$

Alternative Fix-Point Iteration: “Time Marching”

$$\begin{aligned} & A(u^{(i+1)}) = f(u^{(i)}) \quad // \cdot \tau \\ + & \quad u^{(i+1)} = u^{(i)} \end{aligned}$$

$$u^{(i+1)} + \tau A(u^{(i+1)}) = u^{(i)} + \tau f(u^{(i)})$$

$$(\text{Id} + \tau A)(u^{(i+1)}) = u^{(i)} + \tau f(u^{(i)})$$

$$u^{(i+1)} = (\text{Id} + \tau A)^{-1}(u^{(i)} + \tau f(u^{(i)}))$$

Problem Model as Functional

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Modelling on *Variational Level*
("Energy Level")

Non-linear Euler-Lagrange PDE

$$\begin{aligned}\Delta\varphi(x) &= -2 \cdot (I_1(x) - I_2(\varphi(x)))\nabla I_2(\varphi(x)) \\ A(\varphi) &= f(\varphi)\end{aligned}$$

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

$$u^{(i+1)} = A^{-1}(f(u^{(i)}))$$

Approximate the solution of the non-linear problem by an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-Seidel, SOR, Multigrid))

Problem Model as Functional

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Modelling on *Variational Level*
("Energy Level")

Non-linear Euler-Lagrange PDE

$$\begin{aligned}\Delta\varphi(x) &= -2 \cdot (I_1(x) - I_2(\varphi(x)))\nabla I_2(\varphi(x)) \\ A(\varphi) &= f(\varphi)\end{aligned}$$

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

$$u^{(i+1)} = A^{-1}(f(u^{(i)}))$$

Approximate the solution of the non-linear problem by an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-Seidel, SOR, Multigrid))

Last Step: Solving System of Linear Equations

```
function u = VARREG(IT, IS)  
  
u=0; IS,u=IS;    // initialize  
  
until (convergence)  
  
    f    = (IT-IS,u)∇IS,u ;    // compute force  
  
    u    = solveLSE(A, f) ; // approximate u = A-1(f)  
  
    IS,u = warp(T, u) ;    // apply disp. u to image  
  
end  
  
return;
```

summary

Problem Model as Functional

$$J[\varphi] = \int_{\Omega} (I_T(x) - I_S(\varphi(x)))^2 dx + \alpha \int_{\Omega} \|\nabla\varphi(x)\|^2 dx$$

Modelling on *Variational Level*
("Energy Level")

Non-linear Euler-Lagrange PDE

$$\begin{aligned}\Delta\varphi(x) &= -2 \cdot (I_1(x) - I_2(\varphi(x)))\nabla I_2(\varphi(x)) \\ A(\varphi) &= f(\varphi)\end{aligned}$$

Necessary Condition for Minimum of Functional
("Don't minimize a functional – solve an equation!")

Quasilinearization

$$u^{(i+1)} = A^{-1}(f(u^{(i)}))$$

Approximate the solution of the non-linear problem by an iteration using a linearized version of the problem
(Fixed-Point Iteration, Time Marching, ...)

Discretization

Discretize the linear problem in order to actually compute the solution on the computer
(Finite Difference Method, Finite Element Method...)

Solve Linear System

Methods are needed for solving the linear systems arising in the Quasilinearization
(Direct Solvers, Iterative Methods (Conjugate Gradients, Jacobi, Gauß-Seidel, SOR, Multigrid))

Method Summary

```
function u = VARREG(IT, IS)  
  
u=0; IS,u=IS; // initialize  
  
until (convergence)  
  
    f = (IT-IS,u)∇IS,u ; // compute force  
    u = solveLSE(A, f) ; // approximate  $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{f})$   
    IS,u = warp(T, u) ; // apply disp.  $\mathbf{u}$  to image  
  
end  
  
return;
```

fin