復旦大學 FUDAN UNIVERSITY





Medical Image Analysis
Diffeomorphic Demons

Lecturer: Xinzhe Luo

School of Data Science, Fudan University

Contents

大数据学院
School of Data Science

Introduction

- Definition & Application

Background

- Registration revisited
- Demons

Theoretical Insight

- Alternate optimization & Demons forces

Diffeomorphic Demons

- Diffeomorphisms & Vector fields
- Velocity fields & Exponentials

Implementation & Demo

- Demons & Diffeomorphic Demons

Motivation

大数据学院
School of Data Science

- **What is medical image registration?**

- **Motivation.** An alignment problem between radiological images.

- **Modalities in medical image registration.**
  - Intra-modality
    - Repeated imaging over time, e.g. MR/MR, CT/CT,
    - Changes in intensity and shape, e.g. lesion development
  - Inter-modality
    - Combining modalities gives extra information, e.g. PET/CT, CT/MR
    - Combine anatomical and functional information, image fusion, e.g. DWI/fMRI

Example: MR-CT registration

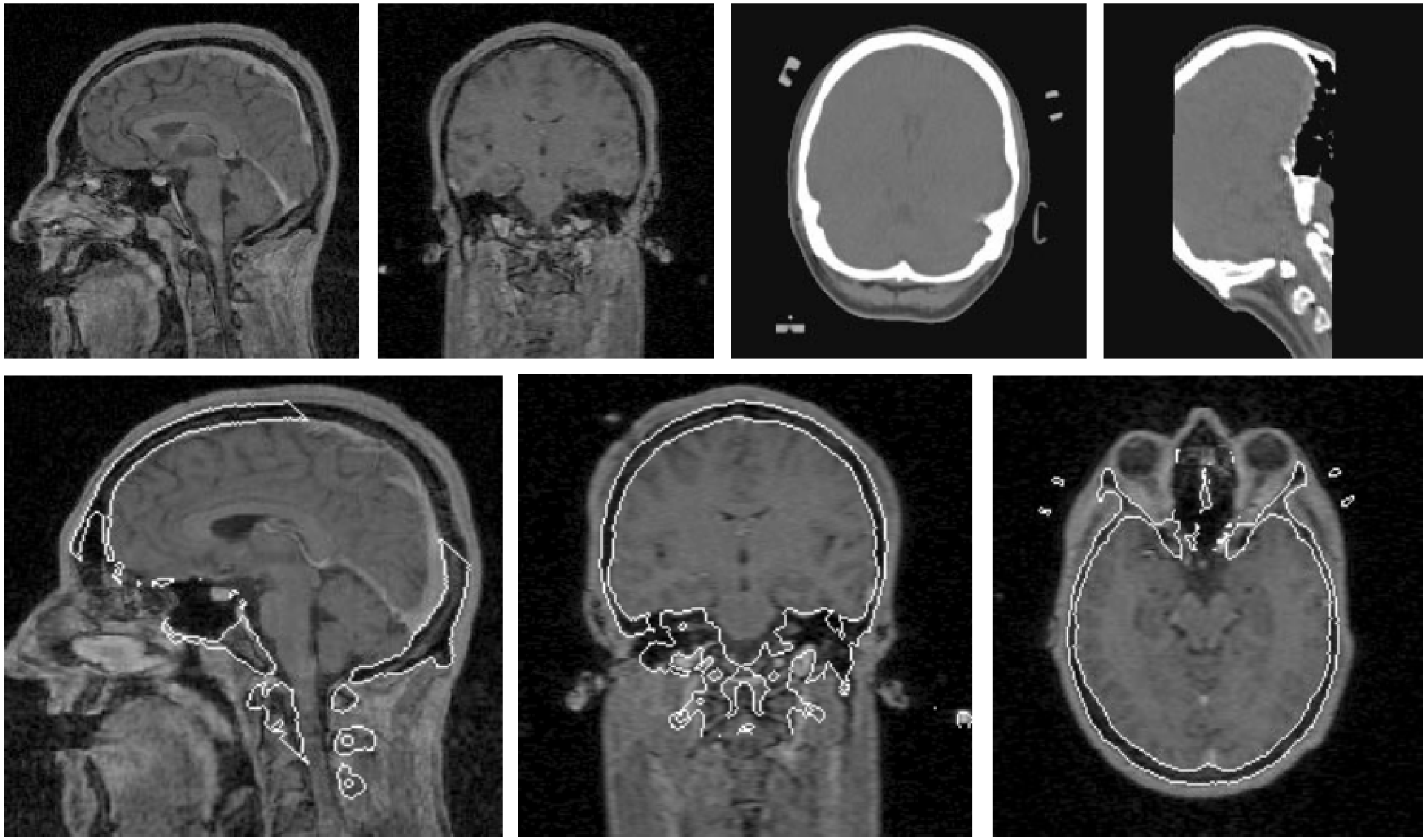


Figure 4. Top row: unregistered MR (left) and CT (right) images. The MR images are shown in the original sagittal plane, and reformatted coronal plane. The CT images in the original oblique plane, and reformatted sagittal plane. Note the different field of view of the images. Bottom panel, MR images in sagittal, coronal and axial planes with the outline of bone, thresholded from the registered CT scan, overlaid. The registration transformation has 10 degrees of freedom, to correct for errors in the machine supplied voxel dimensions and gantry tilt angle.

Hill, D. L. G., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, *46*(3), R1–R45. https://doi.org/10.1088/0031-9155/46/3/201

Mostly used applications

- **Diagnosis.**
  - Combining information from multiple imaging modalities
- **Studying disease progression.**
  - Monitoring changes in size, shape, position or image intensity over time
- **Image guided surgery or radiotherapy.**
  - Relating preoperative images and surgical plans to the physical reality of the patient
- **Patient comparison or atlas construction.**
  - Relating one individual's anatomy to a standardized atlas

Formal definition of registration

大数据学院 School of Data Science

- **Definition.** Spatial alignment between images of the same or different subjects.
  - Transformation $\boldsymbol{T}$ that can relate the position of features in one image or coordinate space with the position of the corresponding feature in another image or coordinate space
  - Registration $\mathcal{T}$ both relates the position of corresponding features and enables us to compare the intensity at those corresponding positions

- **Remarks.**
  - $\boldsymbol{T}: \Omega_M \to \Omega_F$ maps between coordinates, whereas $\mathcal{T}$ maps an image to an image. However, $\mathcal{T}$ is not an intensity mapping.

Hill, D. L. G., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, *46*(3), R1–R45. https://doi.org/10.1088/0031-9155/46/3/201

Registration objective

- **Problem.** To compare images F and M, it is not possible to compare $F(\boldsymbol{x}_F)$ and $M(\boldsymbol{x}_M)$ since F is not defined at location $\boldsymbol{x}_F$. We then introduce the notation $M^{\mathcal{T}}$ for the image M transformed with a given mapping $\mathcal{T}$.

- **Objective.** $F: \boldsymbol{x}_F \in \Omega_F \mapsto F(\boldsymbol{x}_F)$, $M: \boldsymbol{x}_M \in \Omega_M \mapsto M(\boldsymbol{x}_M)$. If $\mathcal{T}$ accurately registers the images, then $F(\boldsymbol{x}_F)$ and $M^{\mathcal{T}}(\boldsymbol{x}_M)$ will represent the same location in the object to within some error depending on $\mathcal{T}$.

Hill, D. L. G., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, *46*(3), R1–R45. https://doi.org/10.1088/0031-9155/46/3/201

Registration process

- The registration process involves recovering the spatial transformation $\boldsymbol{T}$ which maps coordinates over the entire domain of interest, i.e. which maps from Ω_M to Ω_F within the overlapping portion of the domain, denoted as

$$\Omega_{F,M}^{\boldsymbol{T}} = \{\boldsymbol{x}_F \in \Omega_F | \boldsymbol{T}^{-1}(\boldsymbol{x}_F) \in \Omega_M\}.$$

- **Feature-based registration.**
  - Identify features in F and M → calculate $\boldsymbol{T}$ for these features → iteratively determine $\boldsymbol{T}$ → infer $\mathcal{T}$ from $\boldsymbol{T}$ when the algorithm converges
- **Intensity-based registration.**
  - Calculate the voxel similarity measure → iteratively determine the transformation $\mathcal{T}$

Hill, D. L. G., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, *46*(3), R1–R45. https://doi.org/10.1088/0031-9155/46/3/201

The discrete nature of images

大数据学院 School of Data Science

- **Discretization.** $\Omega = \widetilde{\Omega} \cap \Gamma_\zeta$
  - $\widetilde{\Omega}$ is a bounded continuous set, called the field of view
  - Γ_ζ is the sampling grid, characterized by the anisotropic sampling spacing $\zeta = (\zeta^x, \zeta^y, \zeta^z)$.
- **Problems.** The intersection of the discrete domains $\Omega_F \cap \boldsymbol{T}(\Omega_M)$ is likely to be the empty set.
  - Fast interpolation algorithms are imperfect
  - When the image M being transformed has higher-resolution sampling than image F, introducing aliasing

- **Remark.** $\mathcal{T}$ thus deals with both the spatial mapping $\boldsymbol{T}$ and the interpolation or blurring used during resampling.

Types of transformation

大数据学院
School of Data Science

Dimensionality.

- 3D-3D, 2D-3D, 2D-2D

Nature of the transformation.

- Rigid body, e.g. structures of interest are bone or are enclosed in bone
- Affine, e.g. account for scanner introduced errors with scaling and skew
- Deformable, e,g. free-form deformation, non-parametric
- One-to-one & diffeomorphism

Modalities involved.

- Intra-modality, e.g. changes in anatomy or pathology; SSD, CC
- Inter-modality, e.g. information-based techniques

Optimization and capture range

- **Problem:** converge to local optima

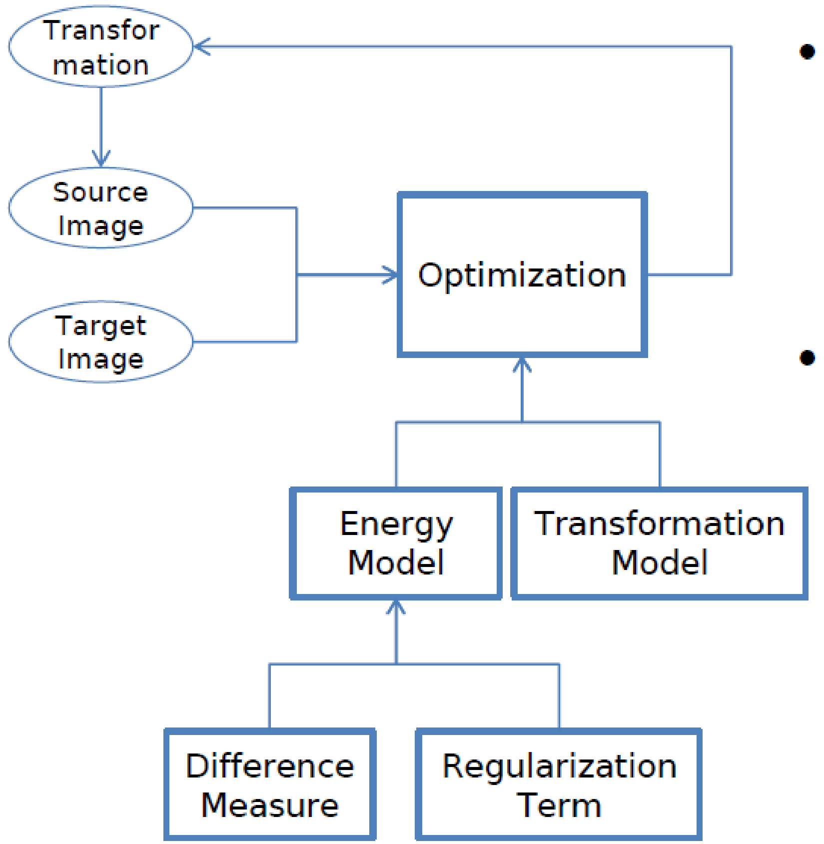
- **Solution:** multi-resolution approaches.

- **Capture range.** Start the algorithm within the capture range of the correct optimum, i.e. within the portion of the parameter space in which the algorithm is more likely to converge to the correct optimum than the incorrect global one.

Registration revisited

Intensity-based framework.

- Did we need another framework?
- Devil in optimization and numeric.



Demons rationale (Thirion, 1995)

大数据学院
School of Data Science

- **More like image matching than registration**
- **Trade theory for efficiency.**
- **Transformation:** each pixel has a displacement vector.
- **Intuitive image forces to compute independent pixel displacements.**
- **Regularize displacements by Gaussian smoothing.**

Non-rigid matching

- **Attraction.** Matching based on an analogy with gravitation
  - A point P of the deformable model M is attracted by all the points P' in F which are similar: K the similarity measure, D the distance

$$\vec{f}(P) = \sum_{P' \in F} \frac{K(P, P')}{D(P, P')} \overrightarrow{PP'}$$

  - Simplified case: consider only contour points with equal similarity between points in M and S.

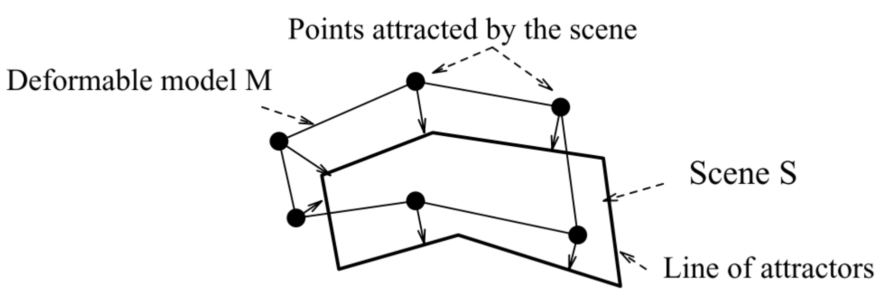


Figure 2. Deformable model with attraction.

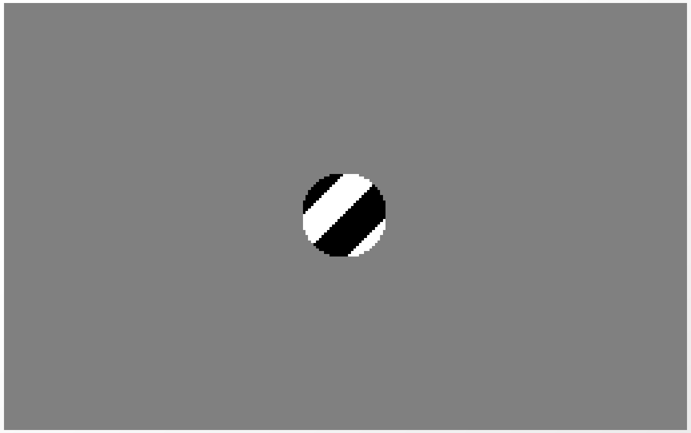
- **Distance is central.**

Non-rigid matching

- **Optical flow.** $m(p) = I(p,t)$, $f(p) = I(p, t+\Delta t)$
  - Brightness constancy constraint: $I(p,t) = I(p+\Delta p, t+\Delta t)$
  - Linear approximation: $I(p+\Delta p, t+\Delta t) \approx I(p,t) + \frac{\partial I}{\partial p}\Delta p + \frac{\partial I}{\partial t}\Delta t$, then we have $\frac{\partial I}{\partial p}\Delta p + \frac{\partial I}{\partial t}\Delta t = 0$, and $\frac{\partial I}{\partial p} v = -\frac{\partial I}{\partial t}$.
  - Optical flow equation, two unknowns (i.e. norm & orientation)

$$\vec{v} \cdot \vec{\nabla} f = m - f$$

- **The aperture problem.**



How is the grating moving?

https://en.wikipedia.org/wiki/File:Aperture_problem_animated.gif

Non-rigid matching

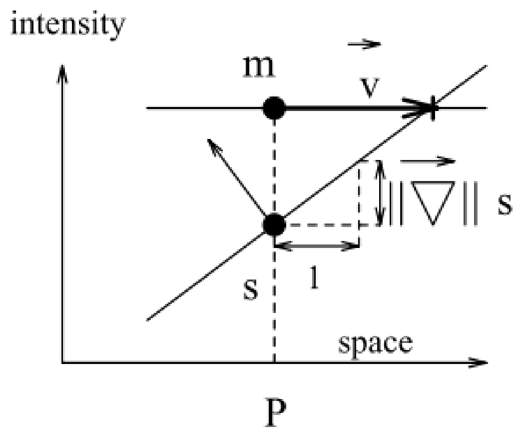
大数据学院 School of Data Science

- **Optical flow solution.** Steepest descent, i.e.

$$\vec{v} = \frac{(m-f)\vec{\nabla} f}{\left\|\vec{\nabla} f\right\|^2}$$

- **Regularization.** Avoid division by zero. Multiplying the equation by $\frac{\left\|\vec{\nabla} f\right\|^2}{\left\|\vec{\nabla} f\right\|^2+(m-f)^2}$ gives

$$\vec{v} = \frac{(m-f)\vec{\nabla} f}{\left\|\vec{\nabla} f\right\|^2 + (m-f)^2}$$



Thirion, J. P. (1998). Image matching as a diffusion process: An analogy with Maxwell's demons. Medical Image Analysis, 2(3), 243–260. https://doi.org/10.1016/S1361-8415(98)80022-4

Diffusion models

- **Maxwell's demons.** Polarity is central.

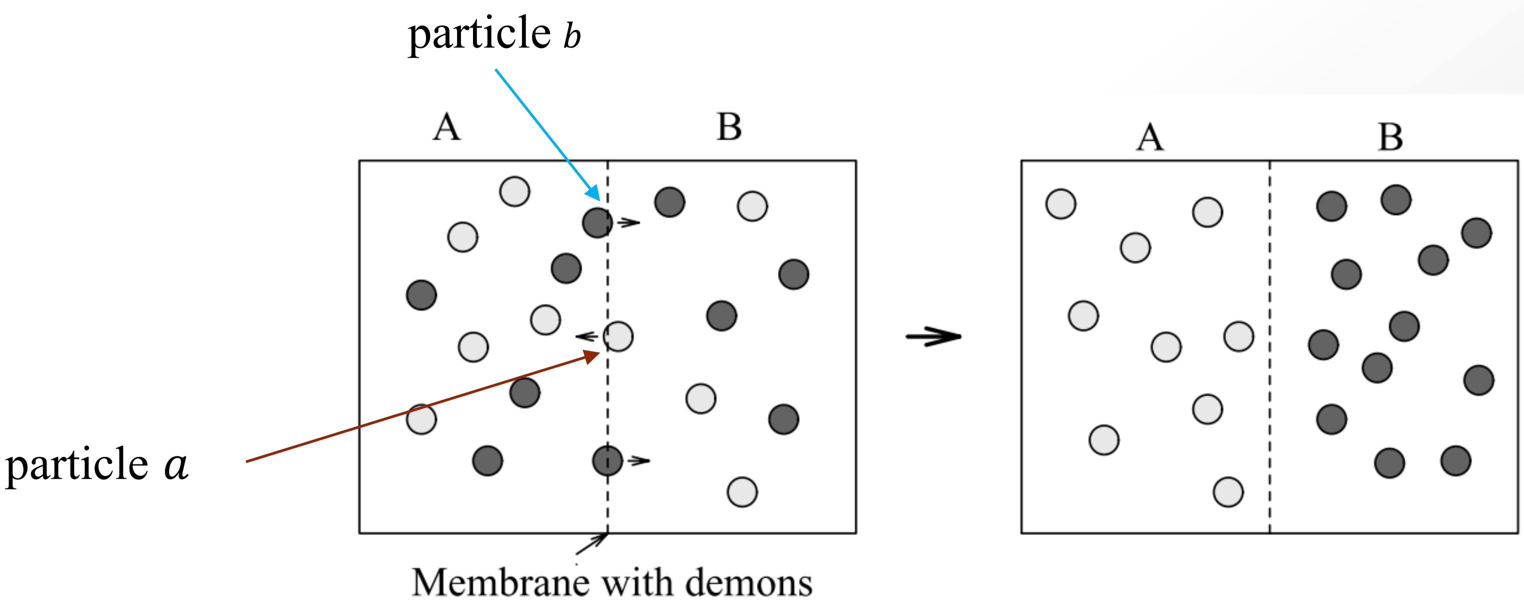
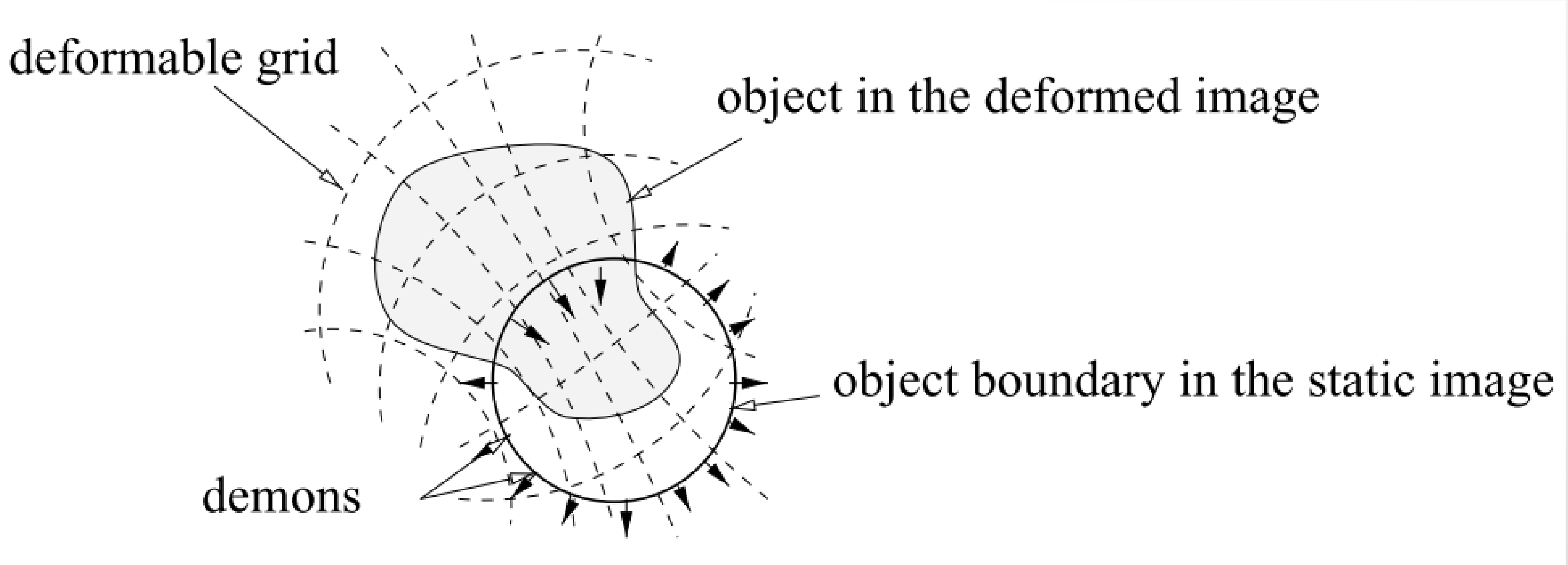


Figure 4. Maxwell's demons and a mixed gas.

Demons for image matching

大数据学院
School of Data Science

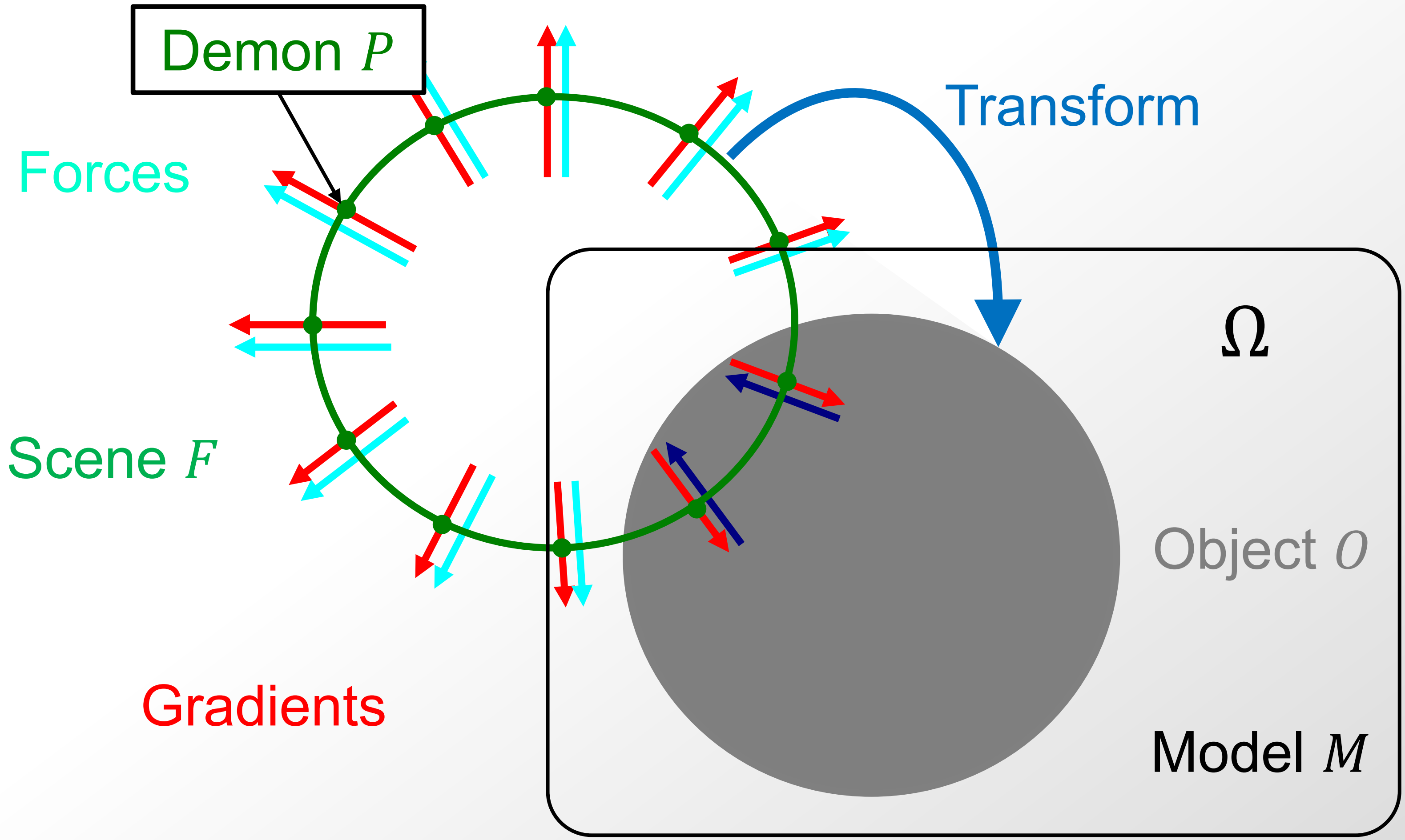
- **Main idea.** To consider the objects boundaries ∂F in the fixed image F as semi-permeable membranes and to let the other moving image M, considered as a deformable grid model, diffuse through these interfaces, by the action of effectors D_F (demons) situated within the membranes[1].



Thirion, J. P. (1998). Image matching as a diffusion process: An analogy with Maxwell's demons. Medical Image Analysis, 2(3), 243–260.

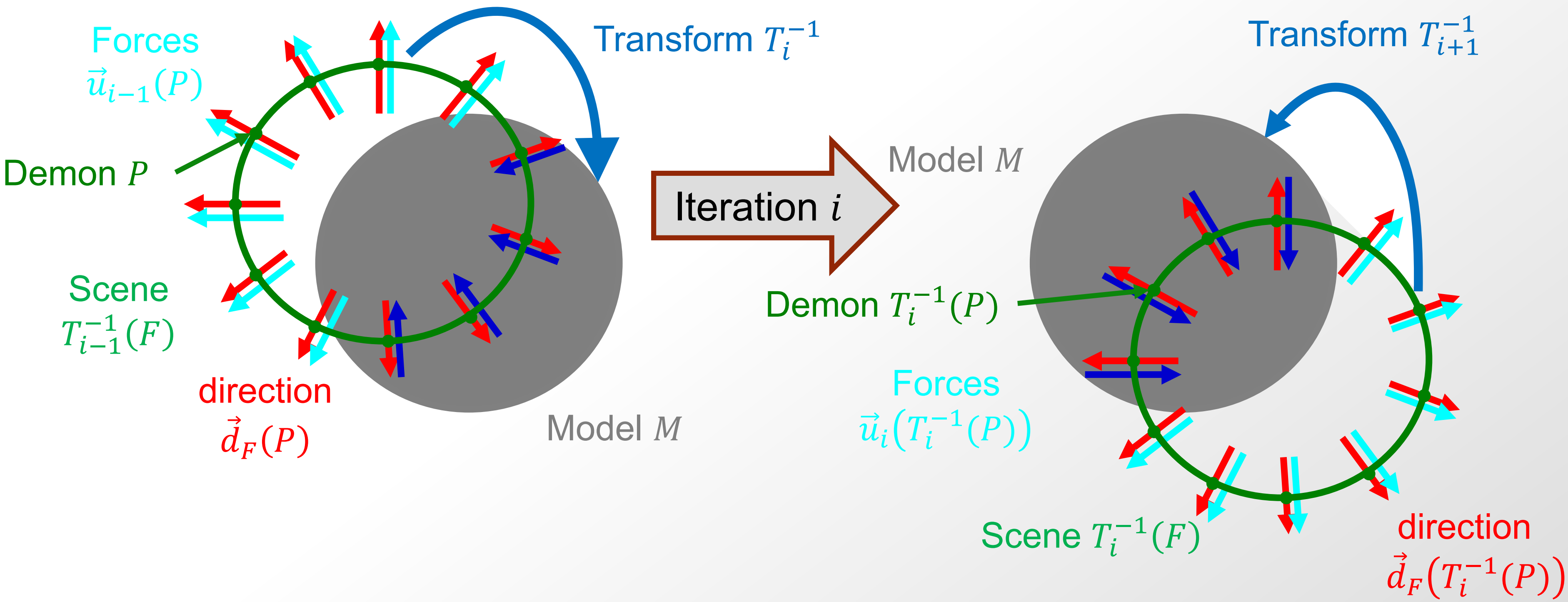
Demons: A toy example

- **Demons is a family of algorithms**
- **Demons 0:** two identical discs are matched rigidly



Demons iterations

- **The iterative part.** For each demon $P \in D_F$, compute the associated demon force $\vec{u}_i(P)$ which depends on the demon direction $\vec{d}_F(P)$ and the polarity of M at point $T_i^{-1}(P)$.



Demons implementation (step-by-step)

Matching as an iterative process

Step 1
- Pre-compute the demons D_F
- D_F can be the whole image grid or restricted to the contour

Step 2
- Compute the forces $\{u\}$ between the current transformed diffusing model $T_i(M)$ and the scene F

Step 3
- Compute the next transformation T_{i+1} from T_i and $\{u\}$

iteration

Demons variants

大数据学院 School of Data Science

Possible variants:

- The selection of demons D_F (whole image grid, contour points)
- Deformation space $\mathcal{T}$ (rigid, affine, free-form)
- Interpolation method for computing intensities m of M at $T_i^{-1}(P)$
- Formula that gives a demon its force $\vec{u}$ (constant, gradient-based)

Demons 0 recap:

- Disc contour of F for D_F
- Rigid transforms for $\mathcal{T}$
- No interpolation because $m\left(T_i^{-1}(P)\right)$ is analytically defined
- Constant magnitude forces

Demons variants

Demons 1: a complete grid of demons

- $D_F = \left\{ f \in F \middle| \vec{\nabla} f \neq 0 \right\}$
- $\mathcal{T}$: free-form deformation, the displacement field given by applying a Gaussian filter to the current elementary displacement $\vec{s}(P)$ to encourage linear elasticity
- $m\left(P + \vec{s}(P)\right)$ estimated using trilinear interpolation in M
- Demon forces implicitly given by optical flow, the displacement $\vec{u} = -\vec{v}$

$$\vec{v} = \frac{\left(m(P') - f(P)\right)\vec{\nabla} f(P)}{\left(\vec{\nabla} f(P)\right)^2 + \left(m(P') - f(P)\right)^2}$$

Demons algorithm recap

Using more convenient notations:

- F and M represent fixed and moving image, respectively

Demons iterations:

- Initial transformation s, given by displacement fields $\boldsymbol{s}$
- Compute image forces $\boldsymbol{u}$ to push $M \circ s$ towards F, i.e. make $M \circ (s + \boldsymbol{u})$ more similar to F
- $s \longleftarrow$ Gaussian smoothing on $s + \boldsymbol{u}$

Beyond Thirion's view

大数据学院
School of Data Science

- **Demons are efficient but have shortcomings:**
  - No strong theoretical analysis
    - Somewhat *ad hoc*, e.g. Not presented as energy minimization
  - $\Longrightarrow$ Difficult to generalize
    - Similarity measures, deformation constraints, etc.
  - $\Longrightarrow$ No insight on convergence

- **More of a framework than an algorithm:**
  - Empirical versus theoretical
  - Why does it work?

Take a break

https://i0.wp.com/dariusforoux.com/wp-content/uploads/2019/06/take-a-break.png?fit=665%2C499&ssl=1

Alternate optimization

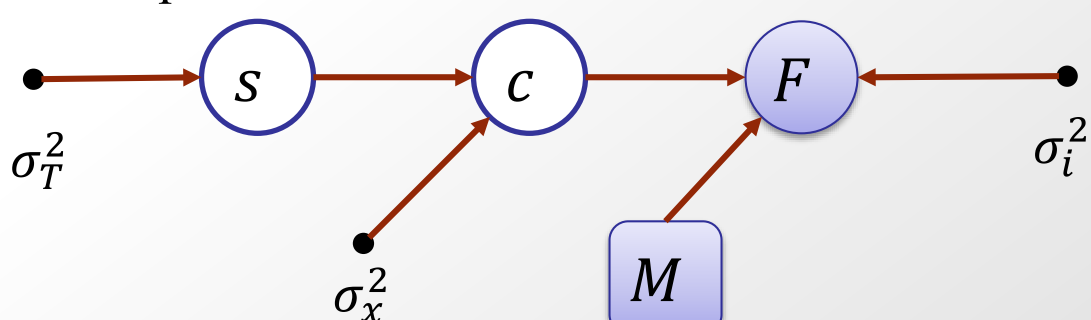
Optimization for classical registration algorithm

$$\hat{s} = \arg\min_{s} E(s) = \arg\min_{s} \frac{1}{\sigma_i^2} \mathrm{Sim}(F, M \circ s) + \frac{1}{\sigma_T^2} \mathrm{Reg}(s)$$

Introducing a prior on s, and hidden variable c, the point correspondences (Cachier et al., 2003)

$$E(c,s) = \left\| \frac{1}{\sigma_i}(F - M \circ c) \right\|^2 + \frac{1}{\sigma_x^2} \mathrm{dist}(s,c)^2 + \frac{1}{\sigma_T^2} \mathrm{Reg}(s)$$

- Classically, $\mathrm{dist}(s,c) = \|s - c\|$, $\mathrm{Reg}(s) = \|\nabla s\|^2$
- Graphical representation:



Alternate optimization

- $\Longrightarrow$ **Alternate optimization.**
- **Step 1.** Optimize $\left\|\frac{1}{\sigma_i}(F - M \circ c)\right\|^2 + \frac{1}{\sigma_x^2}\mathrm{dist}(s, c)$ w.r.t. c, given s
- **Step 2.** Optimize $\frac{1}{\sigma_x^2}\mathrm{dist}(s, c) + \frac{1}{\sigma_T^2}\mathrm{Reg}(s)$ w.r.t. s, given c; closed-form solution when the regularization is quadratic and uniform.
- Given the regularization criterion $\mathrm{Reg}(s) = \|\nabla s\|^2$, the optimal regularized deformation field is the convolution of the correspondence field by a Gaussian kernel.

Algorithm 1 (Mapping). Compositve demons iterations

- Current transformation s (given by a dense displacement field $\boldsymbol{s}$):

$$s = Id + \boldsymbol{s}$$

- **Step 1.** Compute the correspondence update field $\boldsymbol{u}$ by minimizing

$$E_s^{corr}(\boldsymbol{u}) = \|F - M \circ s \circ (Id + \boldsymbol{u})\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\|\boldsymbol{u}\|^2$$

- **Step 2. Fluid-like regularization:** $\boldsymbol{u} \leftarrow K_{\text{fluid}} * \boldsymbol{u}$,

- **Compositive update:**

Let $c \leftarrow s \circ (Id + \boldsymbol{u})$

- **(Step 2.) Diffusion-like regularization (Gaussian):**

$$\boldsymbol{s} \leftarrow K_{\text{diff}} * \boldsymbol{c} \text{ or } s \leftarrow Id + K_{\text{diff}} * (c - Id)$$

Algorithm 2 (Vector space). Additive demons iterations

- **Step 1.** Compute the correspondence update field $\boldsymbol{u}$ to minimize

$$E_s^{corr}(\boldsymbol{u}) = \|F - M \circ (s + \boldsymbol{u})\|^2 + \frac{\sigma_i^2}{\sigma_x^2} \|\boldsymbol{u}\|^2$$

- **Step 2. Fluid-like regularization:** $\boldsymbol{u} \leftarrow K_{\text{fluid}} * \boldsymbol{u}$

- **Additive update:**

Let $\boldsymbol{c} \leftarrow \boldsymbol{s} + \boldsymbol{u}$

- **(Step 2.) Diffusion-like regularization (Gaussian):**

$$\boldsymbol{s} \leftarrow K_{\text{diff}} * \boldsymbol{c} \text{ or } s \leftarrow Id + K_{\text{diff}} * (c - Id)$$

Compositive and additive demons

- **Remarks.** Addition of spatial transformations is computationally efficient but has no geometric meaning.
  - $\Longrightarrow$ slower convergence
  - $\Longrightarrow$ away from the gold standard

- **On the other hand,** composition is natural and geometrically meaningful.
  - The composition $s \circ (Id + \boldsymbol{u})$ requires to warp the dense displacement field $\boldsymbol{s}$ with $\boldsymbol{u}$ and to add the result with $\boldsymbol{u}$.

$$s \circ (Id + \boldsymbol{u}) = (Id + \boldsymbol{s}) \circ (Id + \boldsymbol{u})$$
$$= Id + \boldsymbol{u} + \boldsymbol{s} \circ (Id + \boldsymbol{u})$$

Demons forces

大数据学院 School of Data Science

- **Find an optimal update field $\boldsymbol{u}$.** Consider the intensity difference at a given point,

$$\varphi_p(s) = F(p) - M \circ s(p)$$

  - Denote $\varphi_p^s(\boldsymbol{u}) = F(p) - M \circ s \circ (Id + \boldsymbol{u})(p)$ in the compositive case and $\varphi_p^s(\boldsymbol{u}) = F(p) - M \circ (s + \boldsymbol{u})(p)$ in the additive case.

- **Assume that the follow linearization is available:** $\varphi_p^s(\boldsymbol{u}) \approx \varphi_p^s(0) + J^p \cdot \boldsymbol{u}(p) = F(p) - M \circ s(p) + J^p \cdot \boldsymbol{u}(p).$

- **Independence of pixels.**

Demons forces

大数据学院
School of Data Science

Independence of pixels.

- Mean squared error

Rewrite the energy function into:

$$E_s^{corr}(\boldsymbol{u}) \approx \|F(p) - M \circ s(p) + J^p \cdot \boldsymbol{u}(p)\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\|\boldsymbol{u}\|^2$$

$$\approx \frac{1}{2|\Omega_P|}\sum_{p\in\Omega_P}\left\|\begin{bmatrix} F(p) - M \circ s(p) \\ 0 \end{bmatrix} + \begin{bmatrix} J^p \\ \frac{\sigma_i(p)}{\sigma_x} I \end{bmatrix} \cdot \boldsymbol{u}(p)\right\|_2^2$$

where Ω_P is the overlap between F and $M \circ s$.

Demons forces

大数据学院 School of Data Science

- **Solve for individual pixel** p, the following normal equation:

$$\begin{bmatrix} J^{p^T} & \frac{\sigma_i(p)}{\sigma_x} I \end{bmatrix} \cdot \begin{bmatrix} J^p \\ \frac{\sigma_i(p)}{\sigma_x} I \end{bmatrix} \cdot \boldsymbol{u}(p)$$

$$= -\begin{bmatrix} J^{p^T} & \frac{\sigma_i(p)}{\sigma_x} I \end{bmatrix} \cdot \begin{bmatrix} F(p) - M \circ s(p) \\ 0 \end{bmatrix}$$

which simplifies into

$$\left(J^{P^T} \cdot J^P + \frac{\sigma_i^2(p)}{\sigma_x^2} I \right) \cdot \boldsymbol{u}(p) = -\big(F(p) - M \circ s(p)\big) \cdot J^{P^T}.$$

Demons forces

- **Sherman-Morrison formula:**

$$\left(A + uv^T\right)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

- **Solution:**

$$\boldsymbol{u}(p) = -\frac{F(p) - M \circ s(p)}{\|J^p\|^2 + \frac{\sigma_i^2(p)}{\sigma_x^2}} \cdot J^{p^T}.$$

If the image noise $\sigma_i(p) = |F(p) - M \circ s(p)|$, we get the demons forces proposed by Thirion.

Demons forces

- **Control the maximum step length by choosing σ_x**

$$\|\boldsymbol{u}(p)\| \leq \frac{\sigma_x}{2}.$$

- **First-order Taylor expansion:**

$$\varphi_p^s(\boldsymbol{u}) = \varphi_p^s(0) + J_s^{\varphi_p} \cdot \boldsymbol{u} + O\left(\|\boldsymbol{u}\|^2\right)$$

where for the compositive update rule

$$J_s^{\varphi_p} = \frac{\partial \varphi_p^s(\boldsymbol{u})}{\partial \boldsymbol{u}(q)^T}\bigg|_{\boldsymbol{u}=0} = -\frac{\partial M \circ s\big((Id+\boldsymbol{u})(p)\big)}{\partial \boldsymbol{u}(q)^T}\bigg|_{\boldsymbol{u}=0}$$

$$= -\frac{\partial M \circ s\big(p+\boldsymbol{u}(p)\big)}{\partial \boldsymbol{u}(q)^T}\bigg|_{\boldsymbol{u}=0} = -\delta_{p,q}\frac{\partial M \circ s(\rho)}{\partial \rho^T}\bigg|_{\rho=p}$$

$$= -\delta_{p,q}\nabla_p^T(M \circ s)$$

- By plugging $J^p = -\nabla_p(M \circ s)$ into the solution, we get a Gauss-Newton step for the compositive update rule.

Variants of the demons forces

- **Moving image force**

$$J^p = -\nabla_p^T (M \circ s)$$

- **Fixed image force**

$$J^p = -\nabla_p^T F$$

- **Symmetric forces** (linked to ESM, efficient second-order minimization)

$$J^p = -\frac{1}{2}\left(\nabla_p^T (M \circ s) + \nabla_p^T F\right)$$

Spatial transformations

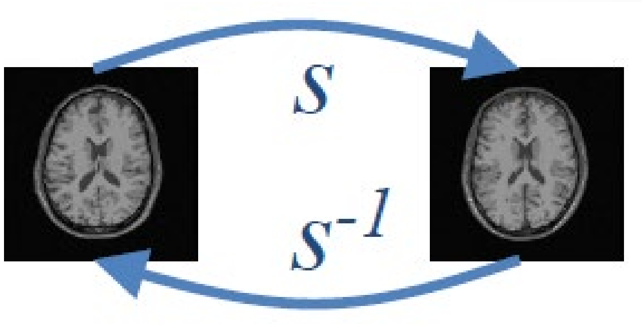
- Spatial transformations do not necessarily form vector spaces
  - G is a vector space
- Addition: no geometric meaning
  - $s_1, s_2 \in G \nRightarrow s = s_1 + s_2 \in G$

- Natural operation: composition
  - $s_1, s_2 \in G \Rightarrow s = s_1 \circ s_2 \in G,\ s\colon p \in G, s(p) = s_1\big(s_2(p)\big)$

- Generally, additive and compositive demons algorithms do not ensure invertibility.

Introducing diffeomorphism

- Deformable registration needs to address the transformations
- Diffeomorphic registration
  - One-to-one, invertible, mapping
  - No foldings
  - Preserves topology
  - Sound assumption if no privileged direction



- Not all problems benefit from diffeomorphisms!
  - Different application ⟹ different constraints
  - E.g. topology changes from tumour resections
- Should leverage a relevant representation of the transformation space

Diffeomorphism

- **Definition.** A map between two manifolds $f: M \to M'$ is called a diffeomorphism if
  - f is one-to-one and onto (bijective)
  - f and f^{-1} are C^∞, a.k.a. smooth

- **Topological space.** $(X, \mathcal{T})$ is called a topological space if
  - $X, \emptyset \in \mathcal{T}$
  - If $O_i \in \mathcal{T}, i = 1, \ldots, n$, then $\bigcap_{i=1}^n O_i \in \mathcal{T}$
  - If $O_\alpha \in \mathcal{T}, \forall \alpha$, then $\bigcup_\alpha O_\alpha \in \mathcal{T}$

Continuity and homeomorphism

- **Continuous mapping.** $(X, \mathcal{T})$ and $(Y, \mathcal{S})$ are two topological space, a mapping $f: X \to Y$ is called continuous or C^0 if

$$O \in \mathcal{S} \Rightarrow f^{-1}(O) \in \mathcal{T}$$

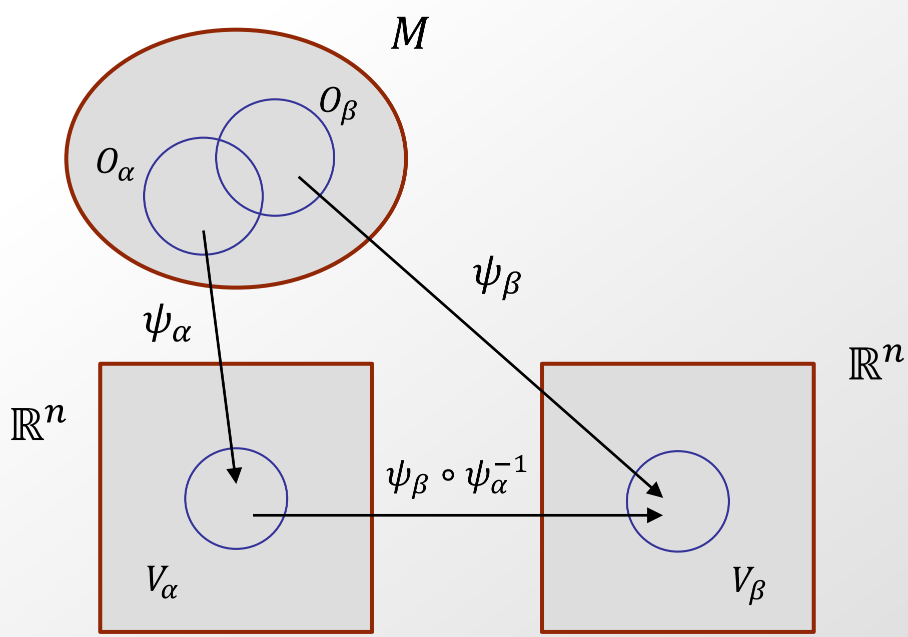
- **Homeomorphism.** A mapping $f: X \to Y$ is called a homeomorphism if
  - f is one-to-one and onto
  - f is continuous from $(X, \mathcal{T})$ to $(Y, \mathcal{S})$
  - $f^{-1}: Y \to X$ is also continuous

Manifold

大数据学院 School of Data Science

Definition. n-dimensional manifold M: looks like $\mathbb{R}^n$ locally

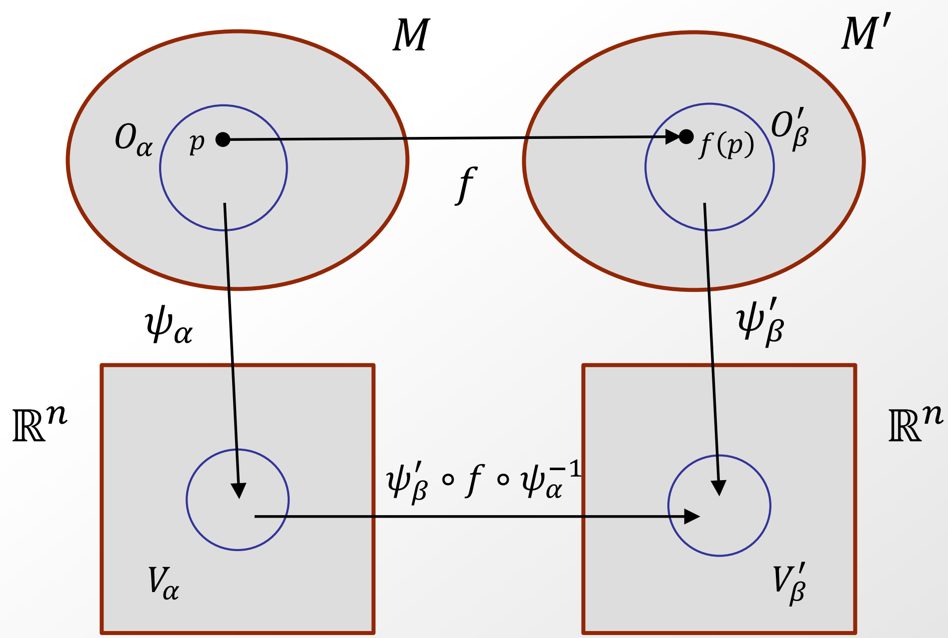
- Topological space: $(M, \mathcal{T})$
- Open cover: $M = \bigcup_\alpha O_\alpha$, $O_\alpha \in \mathcal{T}$
- Homeomorphism: $\forall O_\alpha$, $\exists$ homemorphism $\psi_\alpha : O_\alpha \to V_\alpha \subset \mathbb{R}^n$
- Smooth composition: if $O_\alpha \cap O_\beta \neq \emptyset$, then $\psi_\beta \circ \psi_\alpha^{-1}$ is C^∞ on $\psi_\alpha[O_\alpha \cap O_\beta] \subset \mathbb{R}^n$.



Smooth mapping

- **C^r-continuity**. $f: M \to M'$ is C^r between two manifolds if
  - $\forall p \in M$ and $O_\alpha \ni p$ open on M, as well as $O'_\beta \ni f(p)$ open on M',
  - the corresponding homeomorphisms $\psi_\alpha: O_\alpha \to V_\alpha \subset \mathbb{R}^n$ and $\psi'_\beta: O'_\beta \to V'_\beta \subset \mathbb{R}^{n'}$, such that

$$\psi'_\beta \circ f \circ \psi_\alpha^{-1}: V_\alpha \to V'_\beta \text{ is } C^r$$



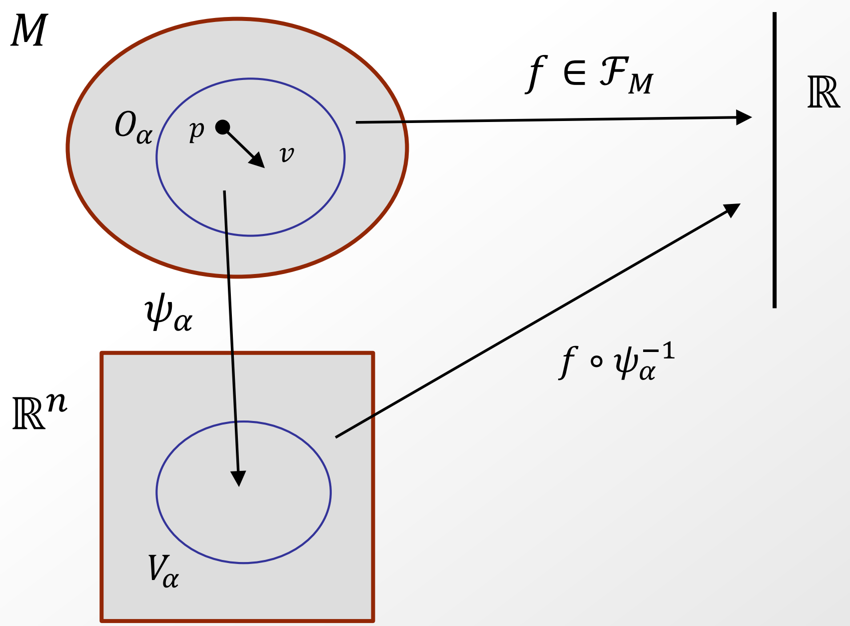
Spatial transformation

- For spatial transformation
  - $M = M' = \mathbb{R}^2$ or $\mathbb{R}^3$
  - $\psi_\alpha = \psi'_\beta = Id$

- **Diffeomorphism Recap.** A map between two manifolds $f: M \to M'$ is called a diffeomorphism if
  - f is one-to-one and onto (bijective)
  - f and f^{-1} are C^∞, i.e. smooth

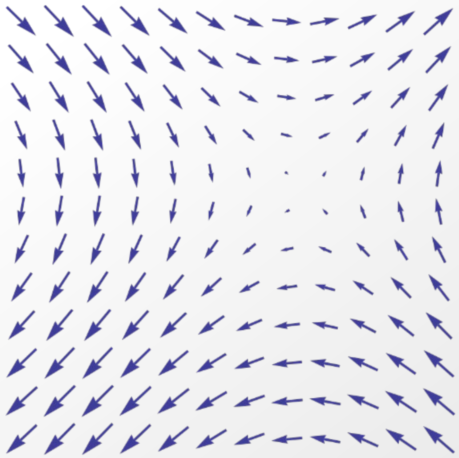
Vector field

- **Vector on manifold.** A vector v at point p on the manifold M is defined as a map $v: \mathcal{F}_M \to \mathbb{R}$, where $\mathcal{F}_M$ is the set of C^∞ scalar fields on M, satisfying the two following conditions:
  - $\forall f, g \in \mathcal{F}_M, \forall \alpha, \beta \in \mathbb{R}, v(\alpha f + \beta g) = \alpha v(f) + \beta v(g)$
  - $\forall f, g \in \mathcal{F}_M, v(fg) = f|_p v(g) + g|_p v(f)$



Vector field

- **Theorem (Coordinate basis).** The set of vectors at point p in the manifold M forms a vector space V_p. Besides, $\dim V_p = \dim M = n$.
  - Coordinate bases: $\{X_\mu\}$, $X_\mu(f) = \frac{\partial F(x)}{\partial x^\mu}$, where $F = f \circ \psi_\alpha^{-1}: \mathbb{R}^n \to \mathbb{R}$
- **Vector field.** Given a C^∞ scalar field $f: M \to \mathbb{R}$ on manifold M, define its vector field as $v(f): M \to \mathbb{R}, p \mapsto v_p(f)$. Besides, if $v(f) \in C^r$, $v(f)$ is called C^r-continuous.

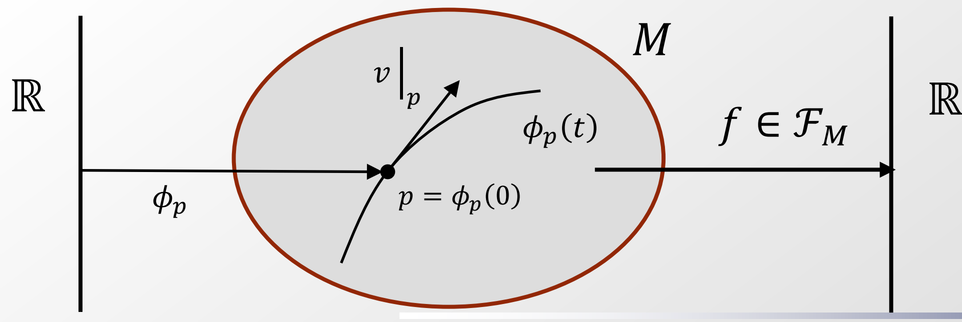


https://en.wikipedia.org/wiki/Vector_field#/media/File:VectorField.svg

One-parameter group of diffeomorphisms

- **Definition.** The one-parameter group of diffeomorphisms G of a manifold M is a C^∞ map $\phi: \mathbb{R} \times M \to M$, i.e. $G = \{\phi_t | t \in \mathbb{R}\}$ satisfying the two following conditions
  - $\forall t \in \mathbb{R}$, $\phi_t: M \to M$ is a diffeomorphism
  - $\forall t, s \in \mathbb{R}$, $\phi_t \circ \phi_s = \phi_{t+s}$

- **Theorem.** A one-parameter group of diffeomorphisms G on the manifold M defines a C^∞ vector field v on M.
  - $\forall p \in M$, $\phi_p: \mathbb{R} \to M$ is a curve on M, and $\phi_p(0) \equiv \phi(0, p) \equiv \phi_0(p) = p$.
  - $v|_p := \frac{\partial}{\partial t}|_p$



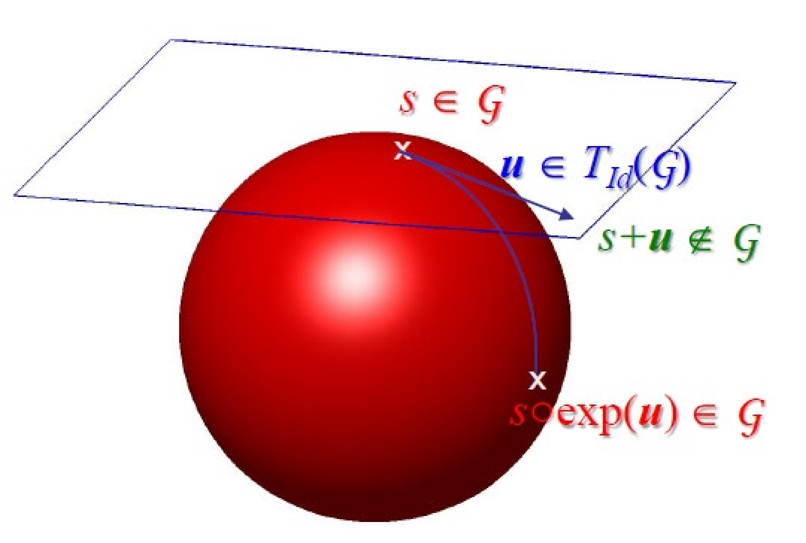
Walking on Lie groups

大数据学院
School of Data Science

- Inconsistency of additive optimization steps, leading to non-invertibility
  - Ad hoc or slow

$$s \leftarrow s + \boldsymbol{u}$$

- Provide a Lie group structure on the space of diffeomorphism
  - Optimize over a space of diffeomorphisms instead of the complete space of non-parametric spatial transformations



Walking on smooth vector field

- In terms of diffeomorphisms, what do one-parameter groups look like?

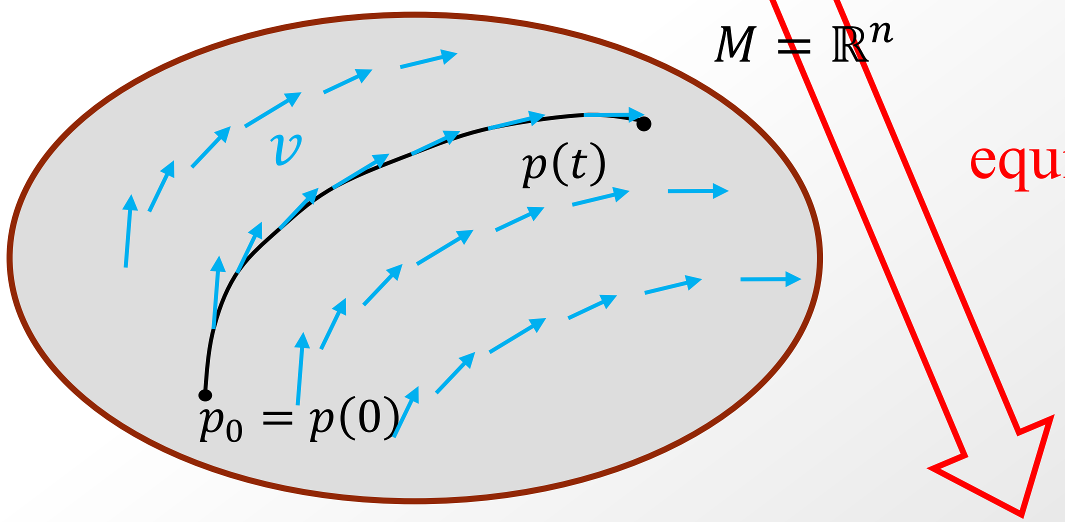
- Quite intuitively, they are all obtained via the integration of stationary ODEs, i.e. ODEs whose speed vector dose not depend on time:

$$\frac{d}{dt}p(t) = v\big(p(t)\big)$$

Arsigny, V., Commowick, O., Pennec, X., & Ayache, N. (2006). A Log-Euclidean Framework for Statistics on Diffeomorphisms. In *Medical image computing and computer-assisted intervention : MICCAI 2006* (Vol. 9, pp. 924–931).

Walking on smooth vector field

- Let $\frac{\partial p(t)}{\partial t} = v\big(p(t)\big)$ be a stationary ODE, where v is a smooth vector field on $\mathbb{R}^n$. Define its flow as the family of mapping $\phi_v(\cdot, t): \mathbb{R}^n \to \mathbb{R}^n$, such that $p(t) = \phi_v(p_0, t)$ is the unique solution of the ODE with $p(0) = p_0$.



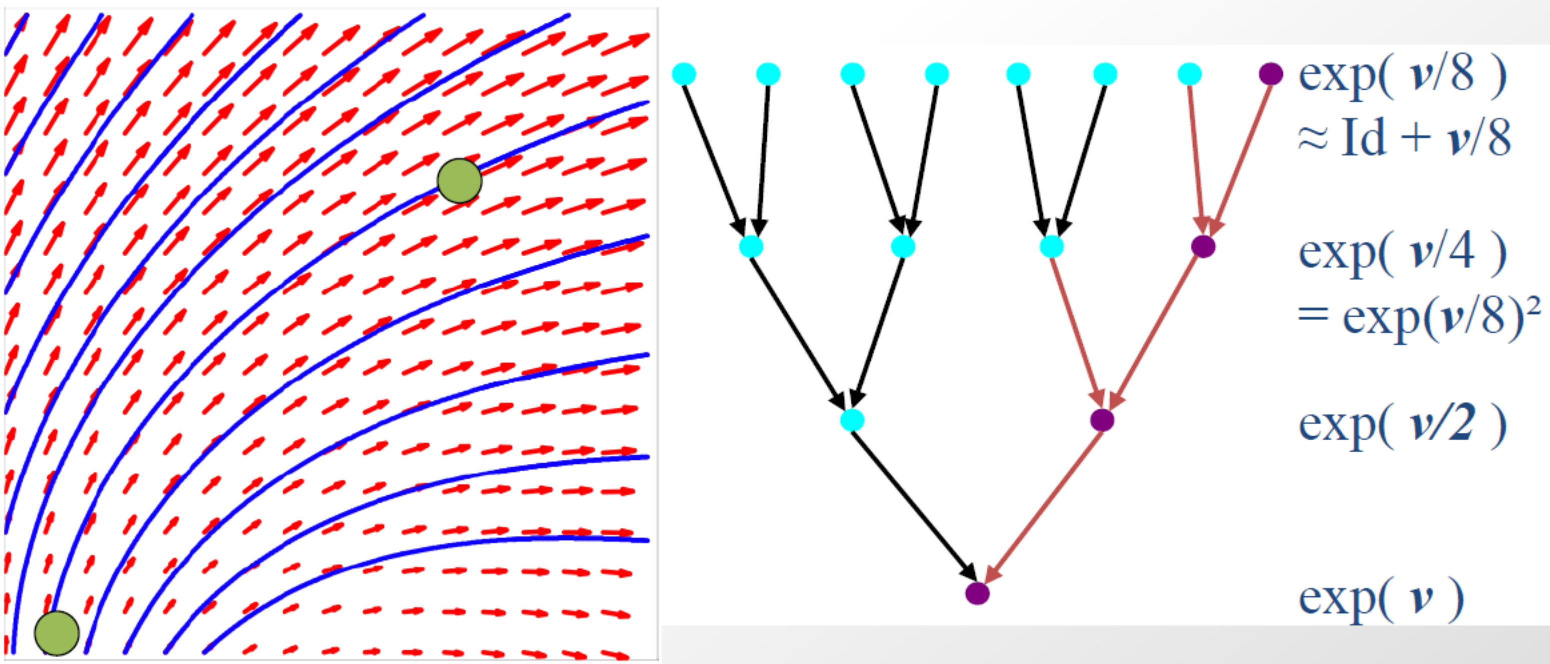
- **Theorem.** A C^∞ vector field on M defines a one-parameter group of diffeomorphisms, e.g. $G = \{\phi_{vt}: p \mapsto p(t) | t \in \mathbb{R}\}$.

Arsigny, V., Commowick, O., Pennec, X., & Ayache, N. (2006). A Log-Euclidean Framework for Statistics on Diffeomorphisms. In *Medical image computing and computer-assisted intervention : MICCAI 2006* (Vol. 9, pp. 924–931).

Exponential of vector fields

- **Definition.** Define the exponential $\exp(v)$ of a smooth vector field v as the flow at time 1 of the stationary ODE $\dot{p} = v(p)$.
- From the properties of one-parameters subgroups, we have for any integer K

$$\exp(v) = \left(\exp\left(K^{-1}v\right)\right)^K$$



Diffeomorphic demons

大数据学院 School of Data Science

- **Speed vector field.** In classical demons, $\boldsymbol{u}$ is a dense displacement field whereas in the diffeomorphic demons, $\boldsymbol{u}$ is considered as a speed vector field.

- **Update rule:**

$$s \leftarrow s \circ \exp(\boldsymbol{u}).$$

Fast vector field exponentials

Algorithm 3. Fast vector field exponentials

- **Scaling step:** Choose N such that $2^{-N}\boldsymbol{u}$ is close enough to 0, e.g. $\max_p \|2^{-N}\boldsymbol{u}(p)\| \le 0.5$

- **Exponentiation step:** Perform an explicit first order integration: $\boldsymbol{v}(p) \leftarrow 2^{-N}\boldsymbol{u}(p)$ for all pixels. As previously we use $v = Id + \boldsymbol{v}$

- **Squaring:** Do N recursive squarings of v: $v \leftarrow v \circ v$

Diffeomorphic demons algorithm

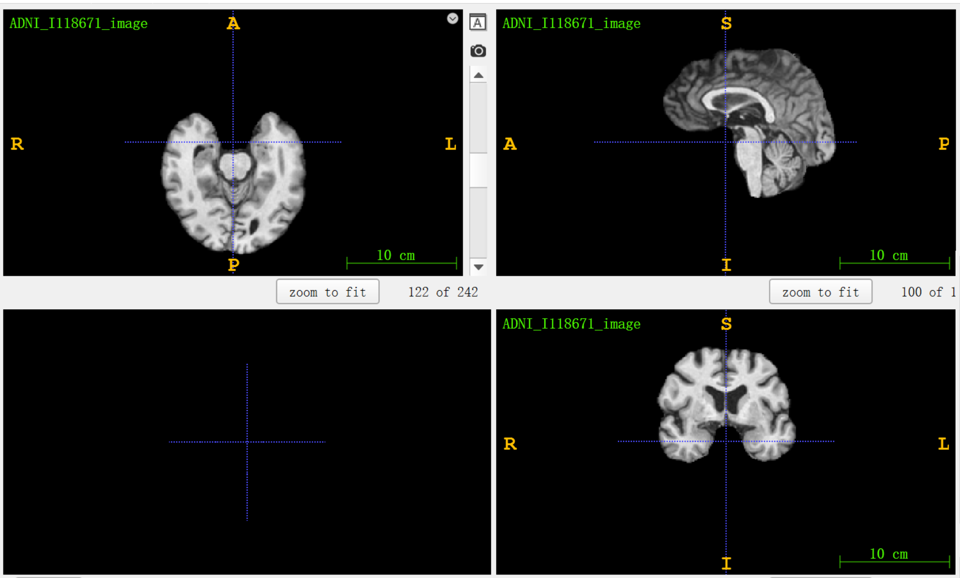
Algorithm 4. diffeomorphic demons iterations

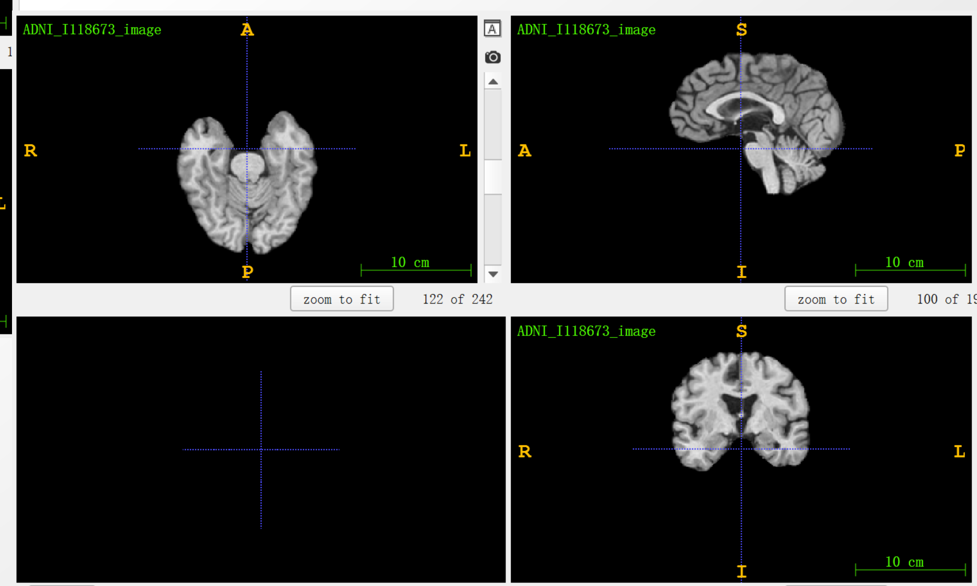
- Compute the correspondence update field $\boldsymbol{u}$ to minimize $E_s^{corr}(\boldsymbol{u})$
- **Fluid-like regularization:** $\boldsymbol{u} \leftarrow K_{\text{fluid}} * \boldsymbol{u}$
- **Exponential and composition:** Let $\boldsymbol{c} \leftarrow s \circ \exp(\boldsymbol{u})$
- **Diffusion-like regularization (Gaussian):**

$$\boldsymbol{s} \leftarrow K_{\text{diff}} * \boldsymbol{c}$$

Demo: Diffeomorphic demons

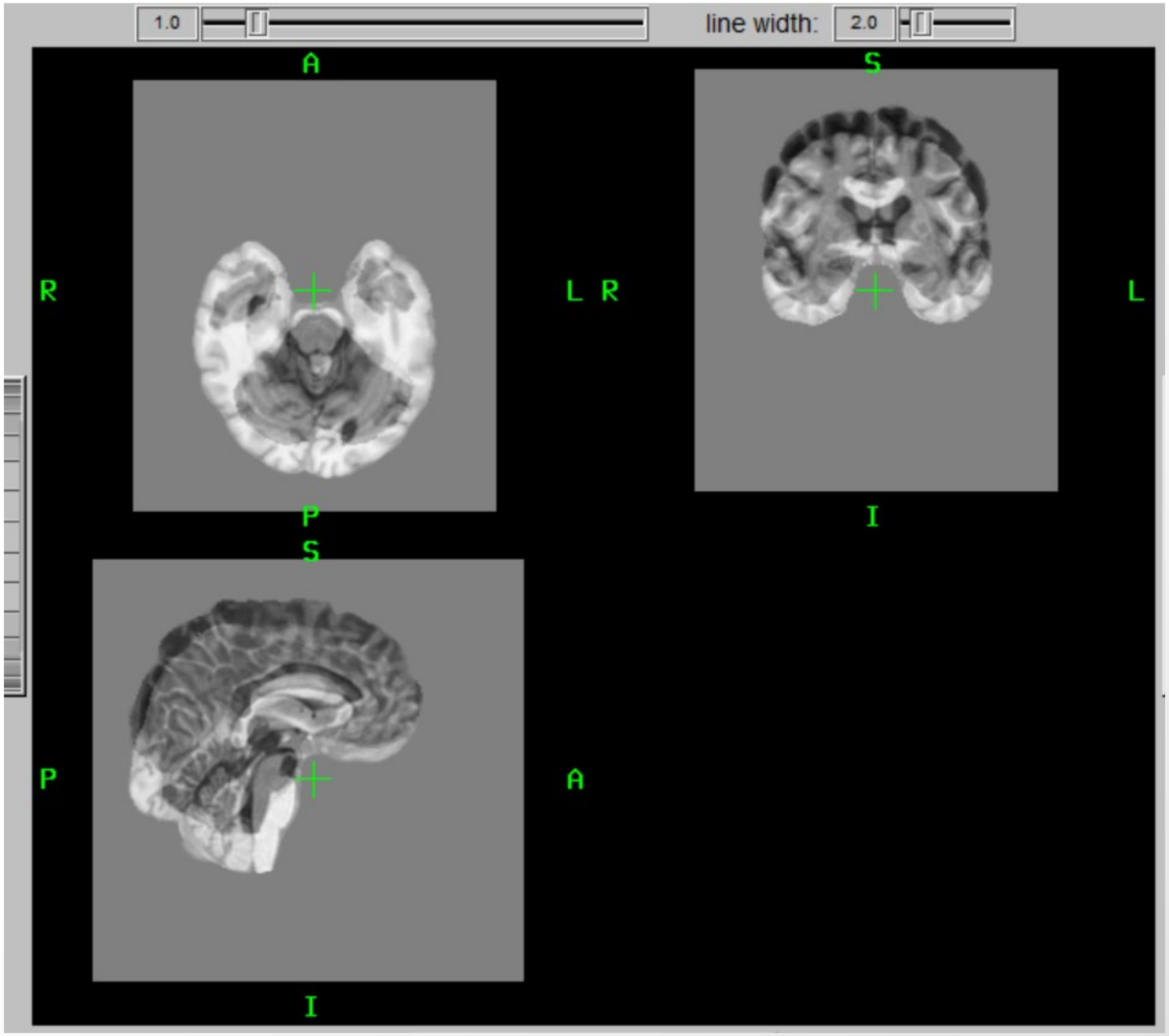
- **Dataset.** ADNI, Alzheimer's Disease Neuroimaging Initiative
- **Modality:** MRI, inter-subject
- **Image information:** $199 \times 240 \times 242$, $1mm \times 1mm \times 1mm$





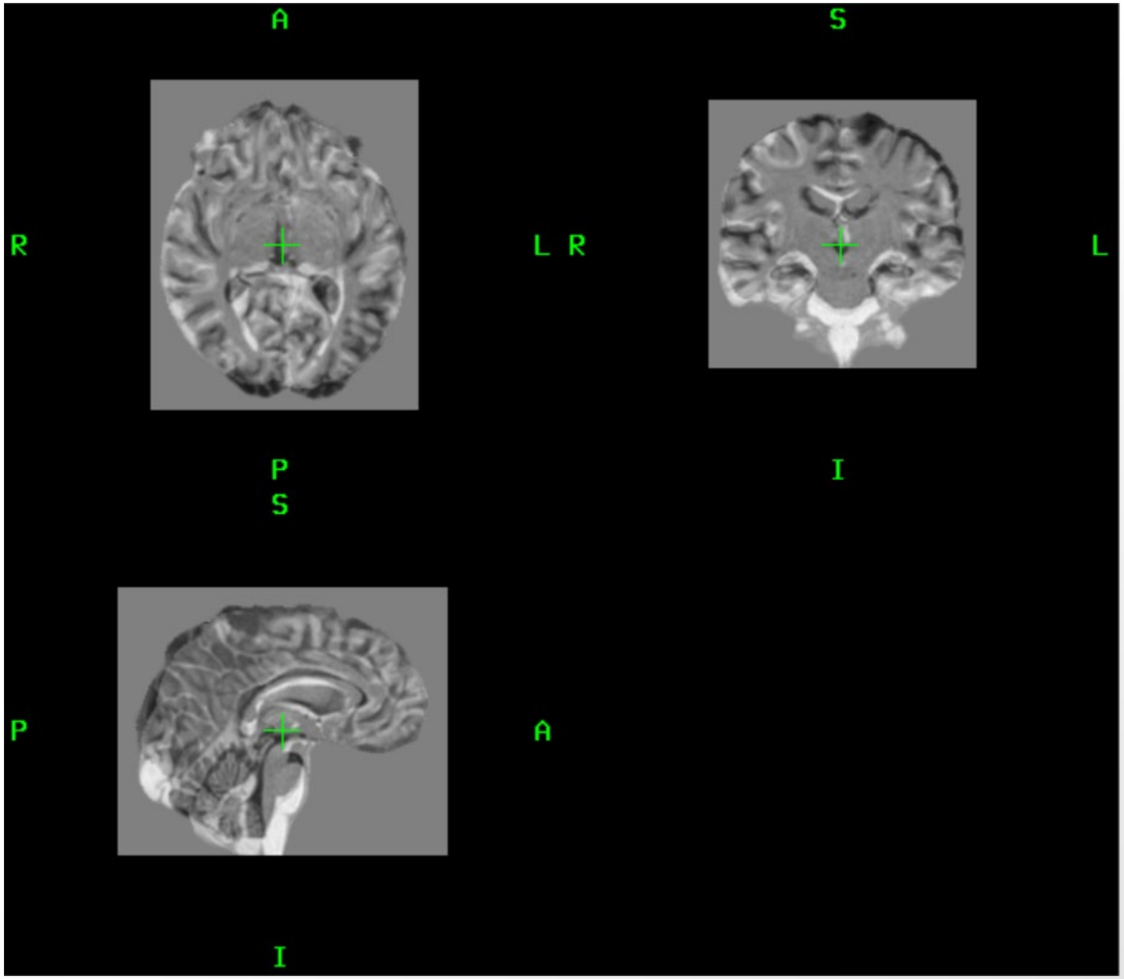
Demo: Diffeomorphic demons

- **Difference image.**



Demo: Diffeomorphic demons

- **Pre-processing:** crop-ROI, rigid & affine registration by zxhtools



Demo: Diffeomorphic demons

大数据学院
School of Data Science

- **Implementation using TensorFlow.**
- Normalize image intensities:

```
def normalize_image(image, normalization=None):
    if normalization == 'min-max':
        image -= tf.reduce_min(image)
        image /= tf.reduce_max(image)

    elif normalization == 'z-score':
        image = (image - tf.reduce_mean(image)) / tf.math.reduce_std(image)

    return image
```

Demo: Diffeomorphic demons

大数据学院
School of Data Science

Compute demons forces.

```python
# compute demons forces
if self.demons_force == 'fixed':
    jacobian = - utils.compute_image_gradient(target_norm_image)
elif self.demons_force == 'moving':
    jacobian = - utils.compute_image_gradient(warped_source_norm_image)
elif self.demons_force == 'symmetric':
    jacobian = - (utils.compute_image_gradient(target_norm_image) + utils.compute_image_gradient(
        warped_source_norm_image)) / 2
```

```python
    def gradient_dx(fv):
        return (fv[:, 2:, 1:-1, 1:-1] - fv[:, :-2, 1:-1, 1:-1]) / 2

    def gradient_dy(fv):
        return (fv[:, 1:-1, 2:, 1:-1] - fv[:, 1:-1, :-2, 1:-1]) / 2

    def gradient_dz(fv):
        return (fv[:, 1:-1, 1:-1, 2:] - fv[:, 1:-1, 1:-1, :-2]) / 2

    def gradient_ixyz(image, fn):
        return tf.stack([fn(image[..., i]) for i in range(channels)], axis=-1)
```

Demo: Diffeomorphic demons

大数据学院
School of Data Science

- Compute correspondence update fields and control it by the maximum step length

```
# get correspondence update fields
update_fields = tf.negative((diff * jacobian) / (tf.norm(jacobian, axis=-1,
                                                        keepdims=True) ** 2 + diff ** 2),
                            name='update_fields')

# control update fields by the maximum step length
update_fields /= tf.reduce_max(tf.norm(update_fields, axis=-1))
update_fields *= self.max_length
```

- Fluid-like regularization

```
# fluid-like regularization
with tf.name_scope('fluid_regularization'):
    update_fields = utils.separable_gaussian_filter3d(update_fields, utils.gauss_kernel1d(sigma=1.))
```

Demo: Diffeomorphic demons

大数据学院 School of Data Science

- Update the correspondence fields, followed by diffusion-like regularization

```python
# update the correspondence fields
with tf.name_scope('update_correspondence_fields'):
    if self.demons_type == 'compositive':
        corres_fields = update_fields + transformer.SpatialTransformer(
            interp_method='linear', name='warp_vector_fields')([self.vector_fields,
                                                                 update_fields])

    elif self.demons_type == 'additive':
        corres_fields = self.vector_fields + update_fields

    elif self.demons_type == 'diffeomorphic':
        exp_update_fields = tf.expand_dims(transformer.integrate_vec(tf.squeeze(update_fields, 0),
                                                                     method='ss', nb_steps=self.exp_steps),
                                           axis=0, name='exp_update_fields')
        corres_fields = exp_update_fields + transformer.SpatialTransformer(
            interp_method='linear', name='warp_fields')([self.vector_fields, exp_update_fields])

# diffusion-like regularization
with tf.name_scope('diffusion_regularization'):
    deform_fields = utils.separable_gaussian_filter3d(corres_fields, utils.gauss_kernel1d(1.))
```

Demo: Diffeomorphic demons

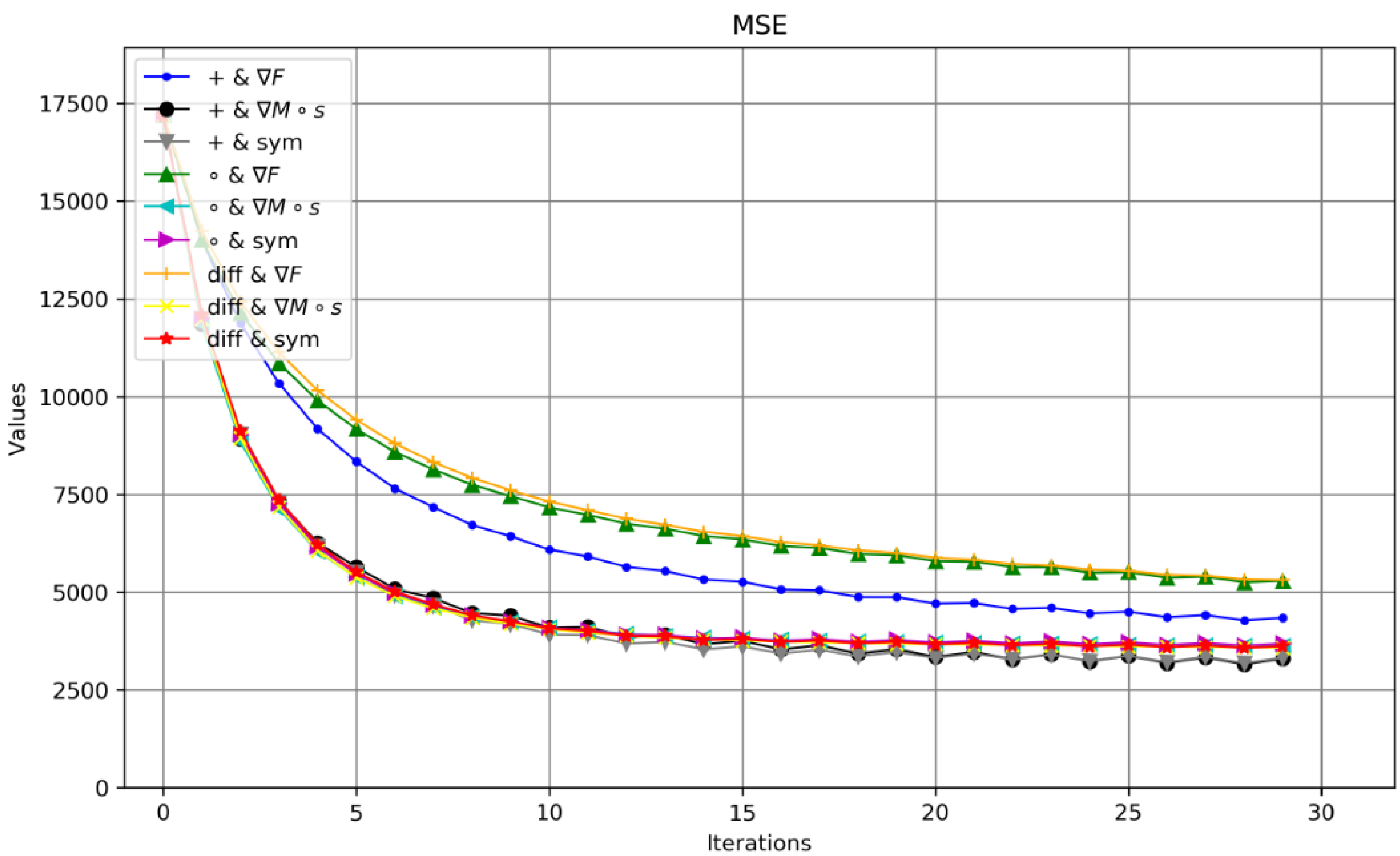
大数据学院 School of Data Science

- Compute local Jacobian determinant as an indicator for diffeomorphism

```python
with tf.name_scope('vector_jacobian_determinant'):
    n_dims = vector_fields.get_shape().as_list()[-1]
    def gradient_dx(fv):
        return (fv[:, 2:, 1:-1, 1:-1] - fv[:, :-2, 1:-1, 1:-1]) / 2

    def gradient_dy(fv):
        return (fv[:, 1:-1, 2:, 1:-1] - fv[:, 1:-1, :-2, 1:-1]) / 2

    def gradient_dz(fv):
        return (fv[:, 1:-1, 1:-1, 2:] - fv[:, 1:-1, 1:-1, :-2]) / 2

    def gradient_txyz(Txyz, fn):
        return tf.stack([fn(Txyz[..., i]) for i in range(n_dims)])

    dTdx = gradient_txyz(vector_fields, gradient_dx)  # [3, batch, nx, ny, nz]
    dTdy = gradient_txyz(vector_fields, gradient_dy)
    dTdz = gradient_txyz(vector_fields, gradient_dz)

    jacobian_det = tf.subtract((dTdx[0]+1)*(dTdy[1]+1)*(dTdz[2]+1) + dTdx[2]*dTdy[0]*dTdz[1] + dTdx[1]*dTdy[2]*dTdz[0],
                               dTdx[2]*(dTdy[1]+1)*dTdz[0] + (dTdx[0]+1)*dTdy[2]*dTdz[1] + dTdx[1]*dTdy[0]*(dTdz[2]+1),
                               name='jacobian_det')

    return jacobian_det
```

Demo: Diffeomorphic demons

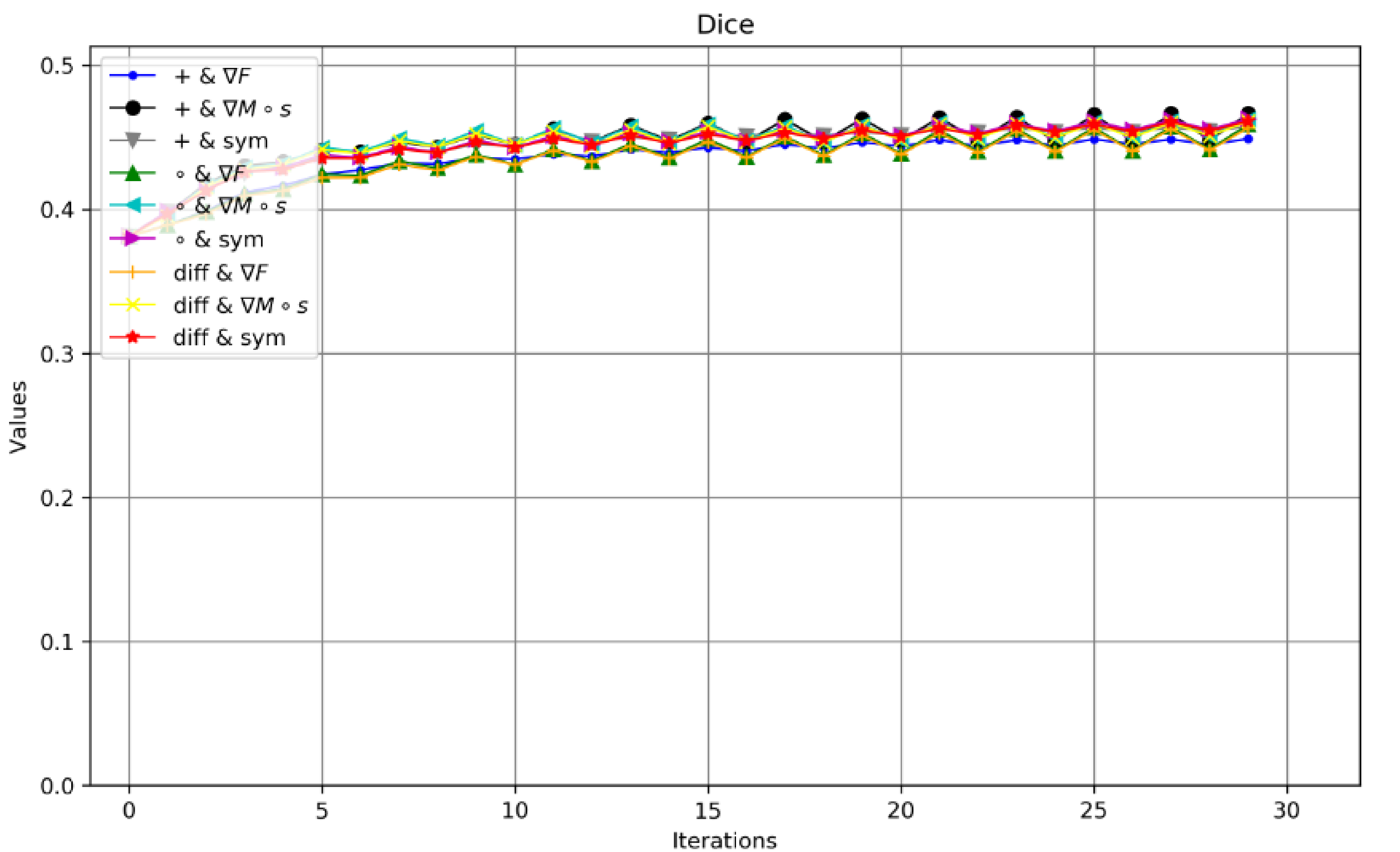
- **Metrics visualization:** Mean squared error



Demo: Diffeomorphic demons

大数据学院 School of Data Science

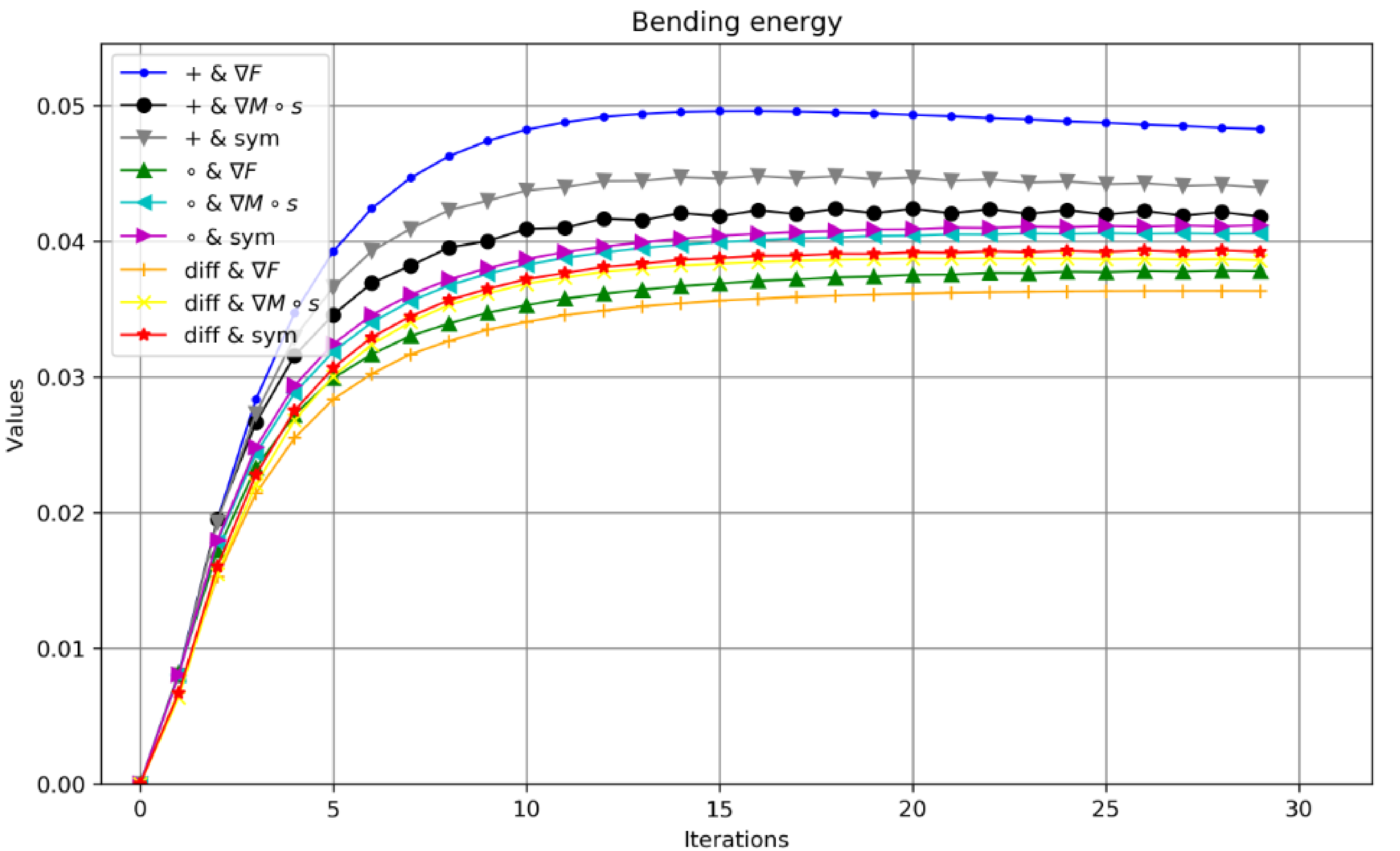
- **Metrics visualization:** Dice similarity metric



Demo: Diffeomorphic demons

大数据学院
School of Data Science

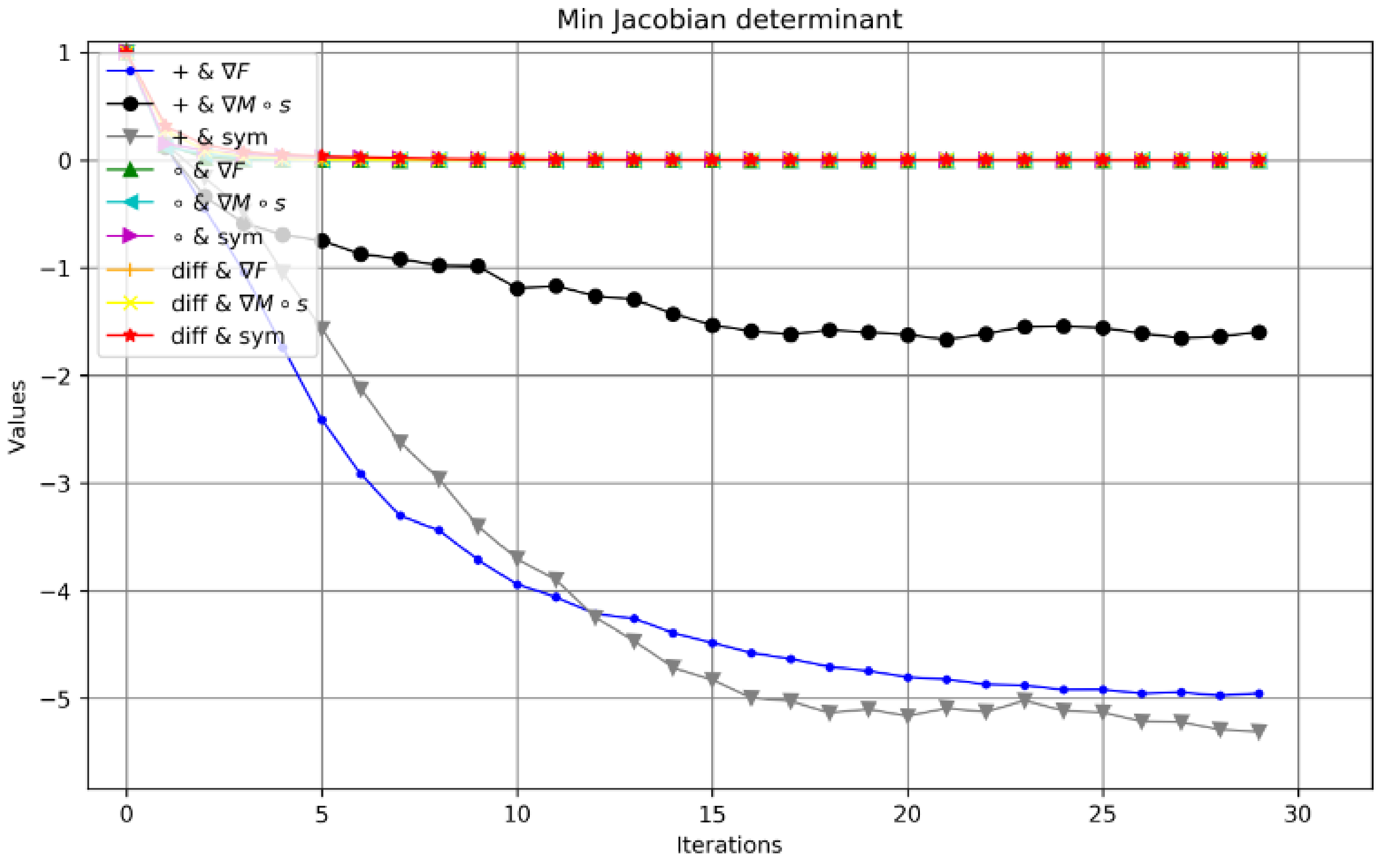
- **Metrics visualization:** Bending energy



Demo: Diffeomorphic demons

大数据学院
School of Data Science

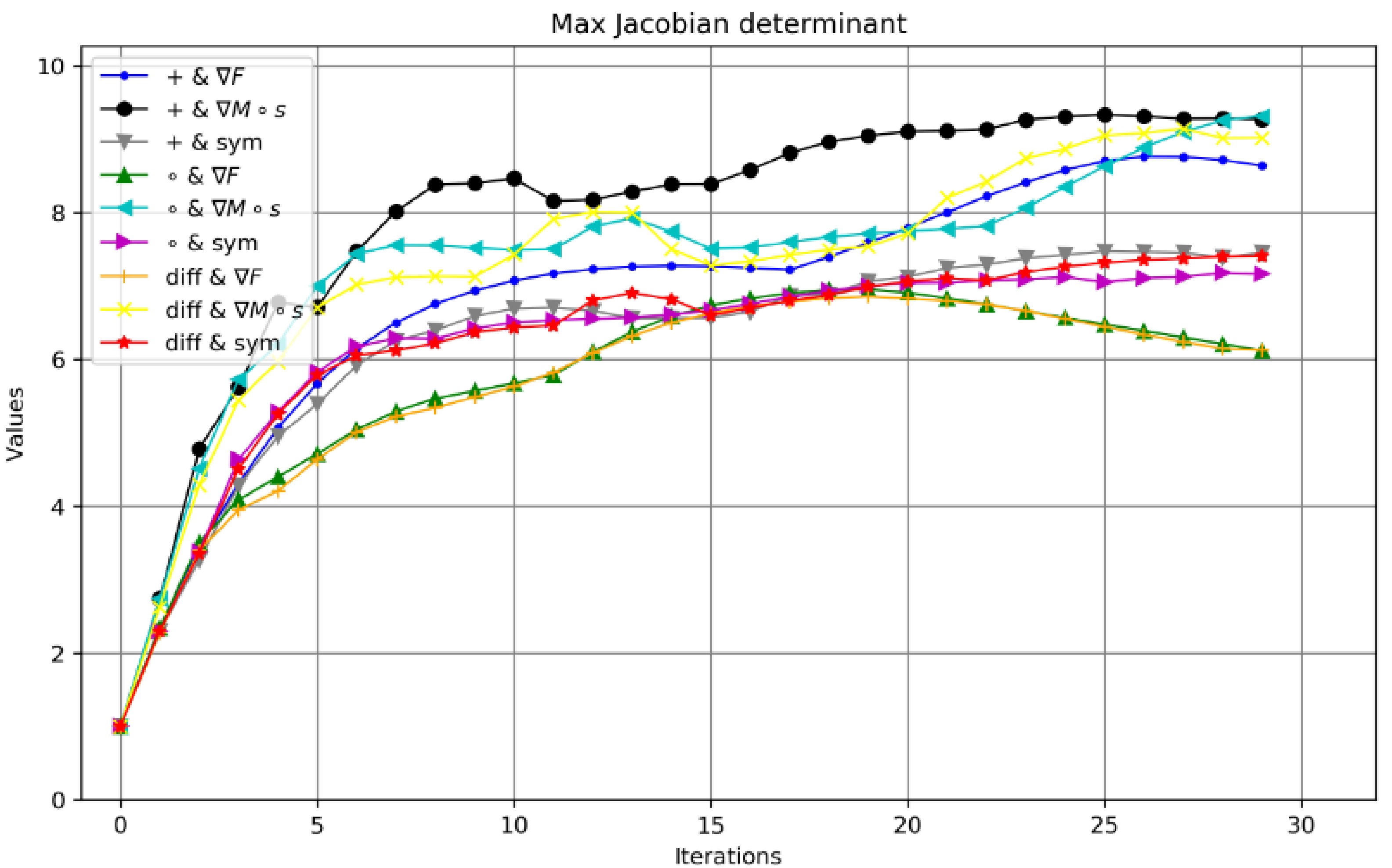
- **Metrics visualization:** Minimum Jacobian determinant



Demo: Diffeomorphic demons

大数据学院
School of Data Science

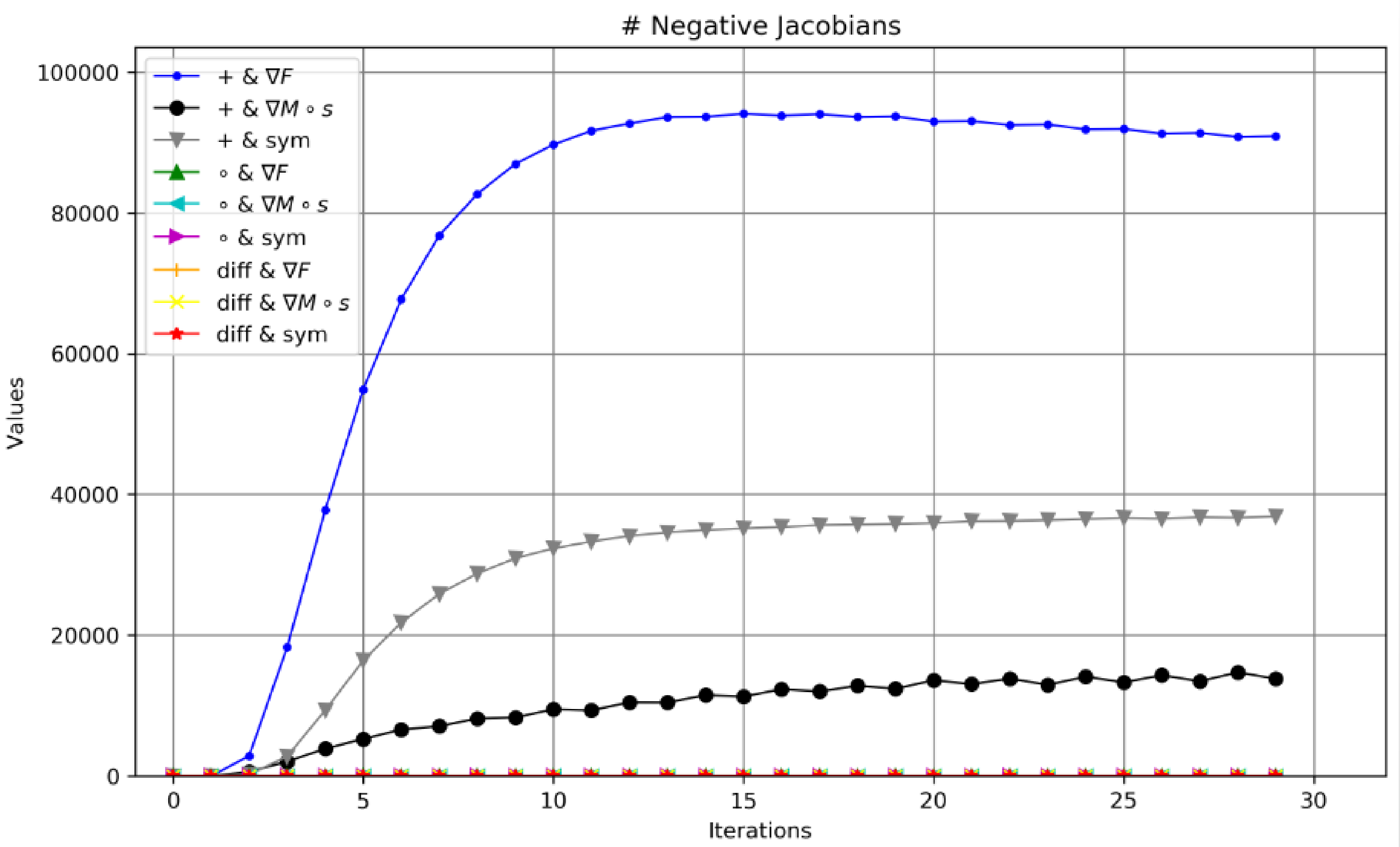
- **Metrics visualization:** Maximum Jacobian determinant



Demo: Diffeomorphic demons

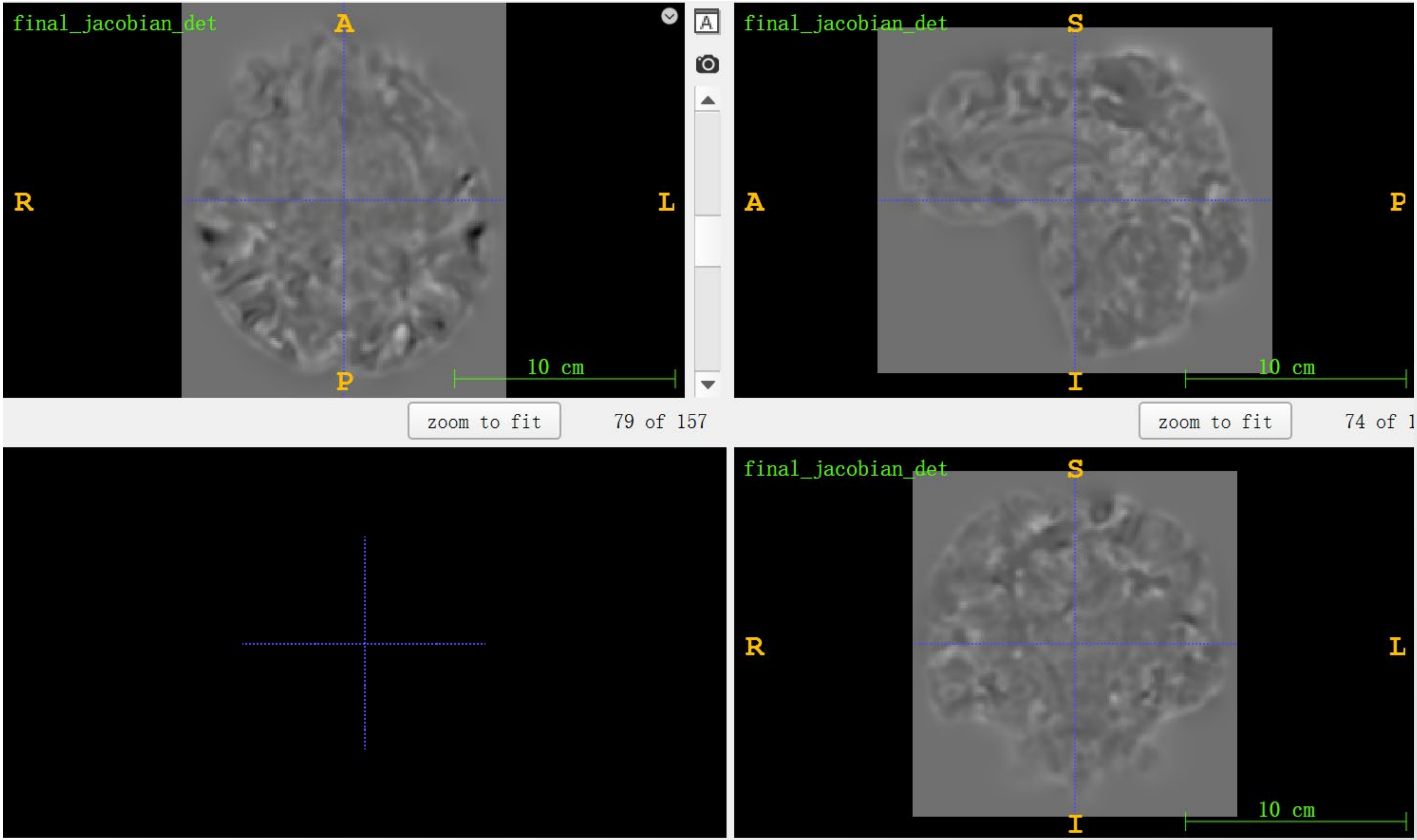
大数据学院 School of Data Science

- **Metrics visualization:** number of locations with $\left|J_s^p\right| \leq 0$



Demo: Diffeomorphic demons

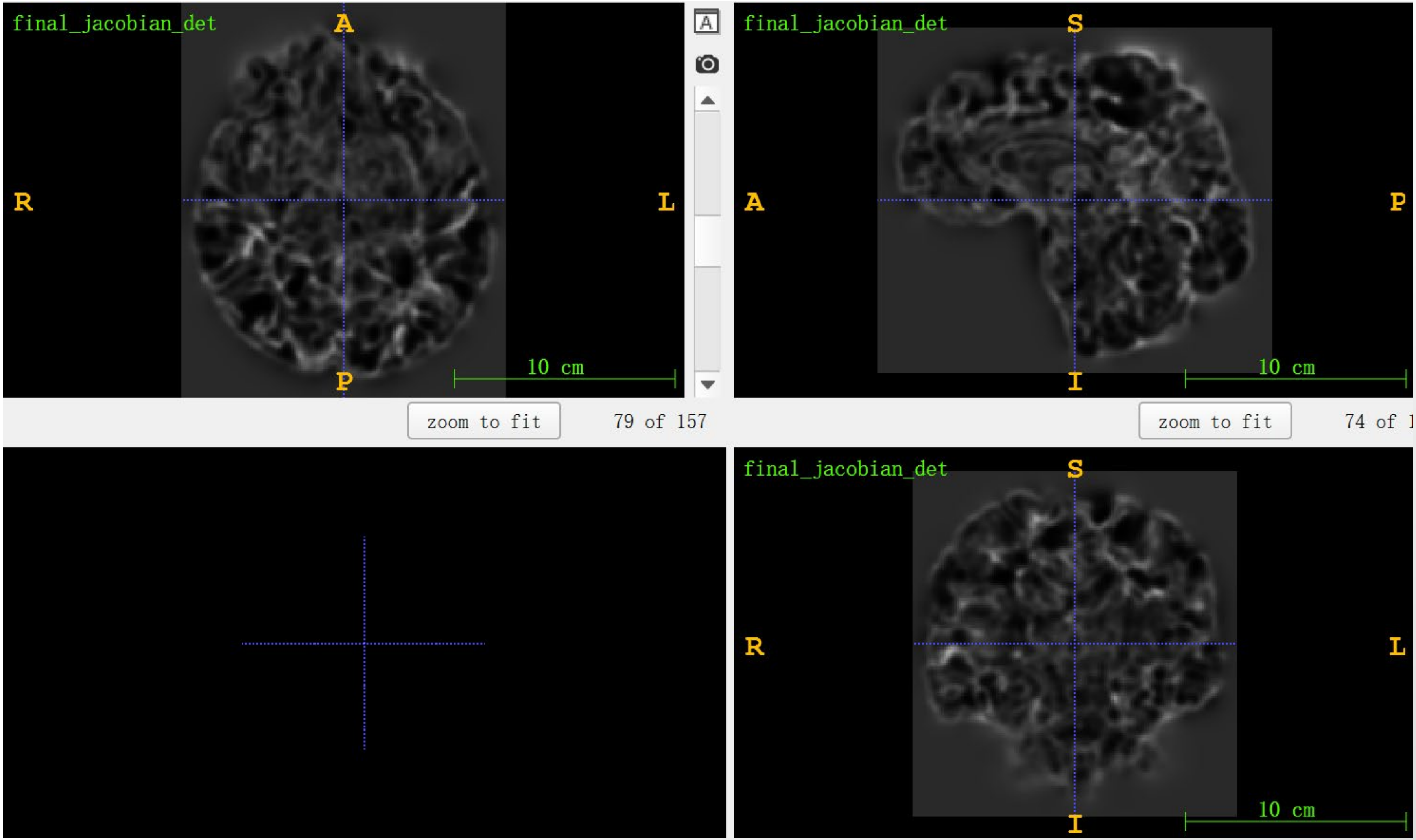
- **Results visualization.** Jacobian determinant (+ & ∇F)



Demo: Diffeomorphic demons

大数据学院 School of Data Science

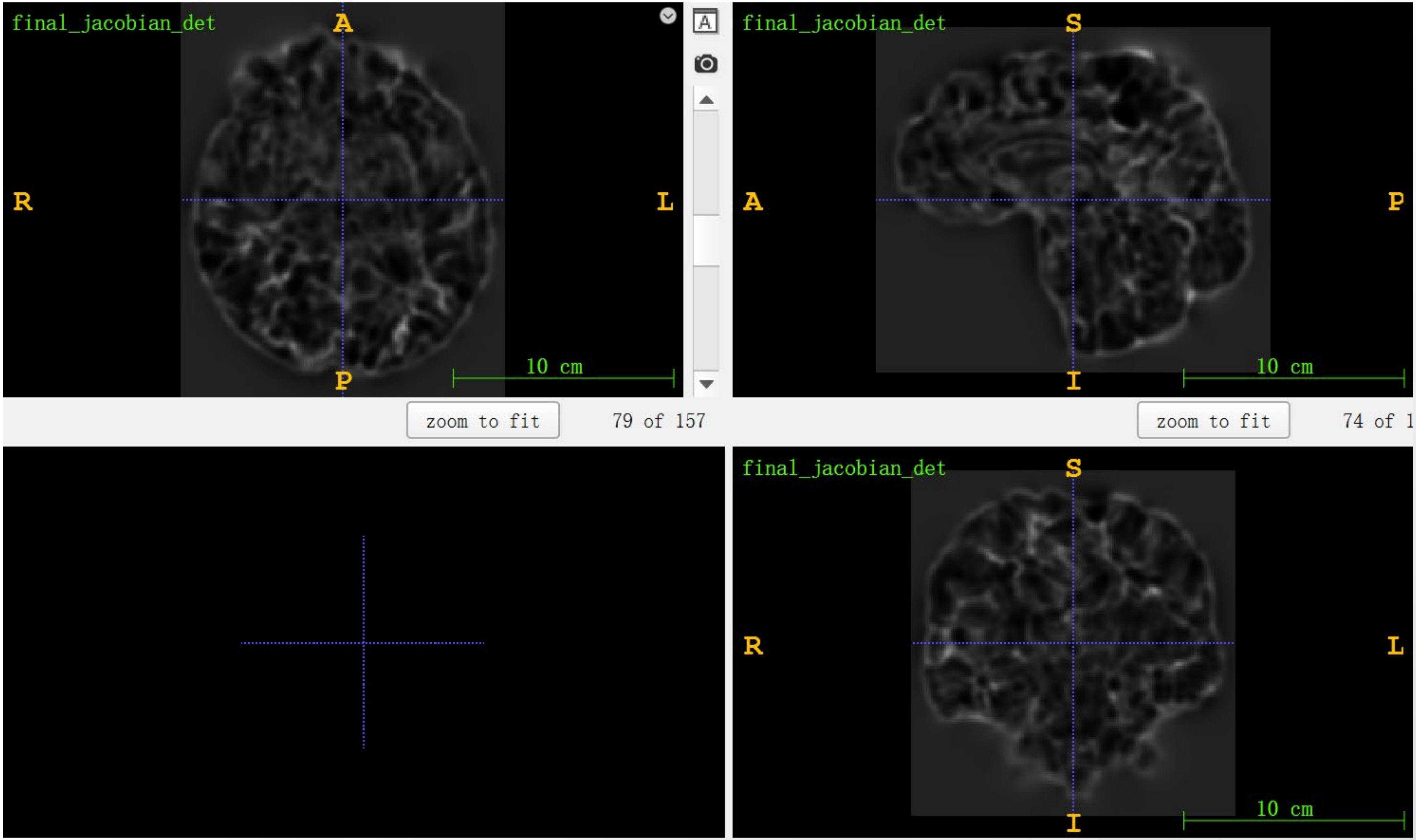
- **Results visualization.** Jacobian determinant (diff & ∇F)



Demo: Diffeomorphic demons

大数据学院 School of Data Science

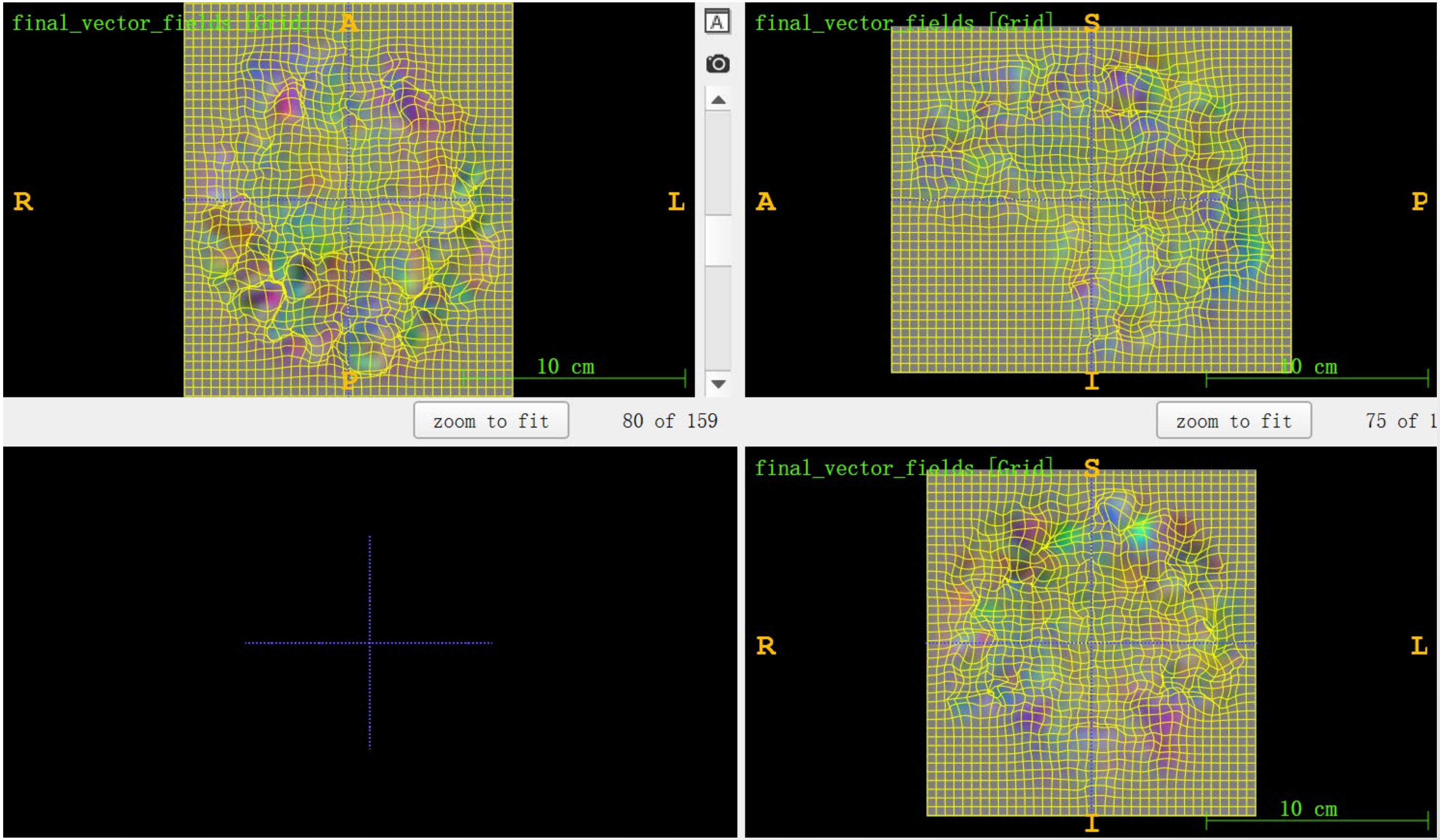
- **Results visualization.** Jacobian determinant (diff & sym)



Demo: Diffeomorphic demons

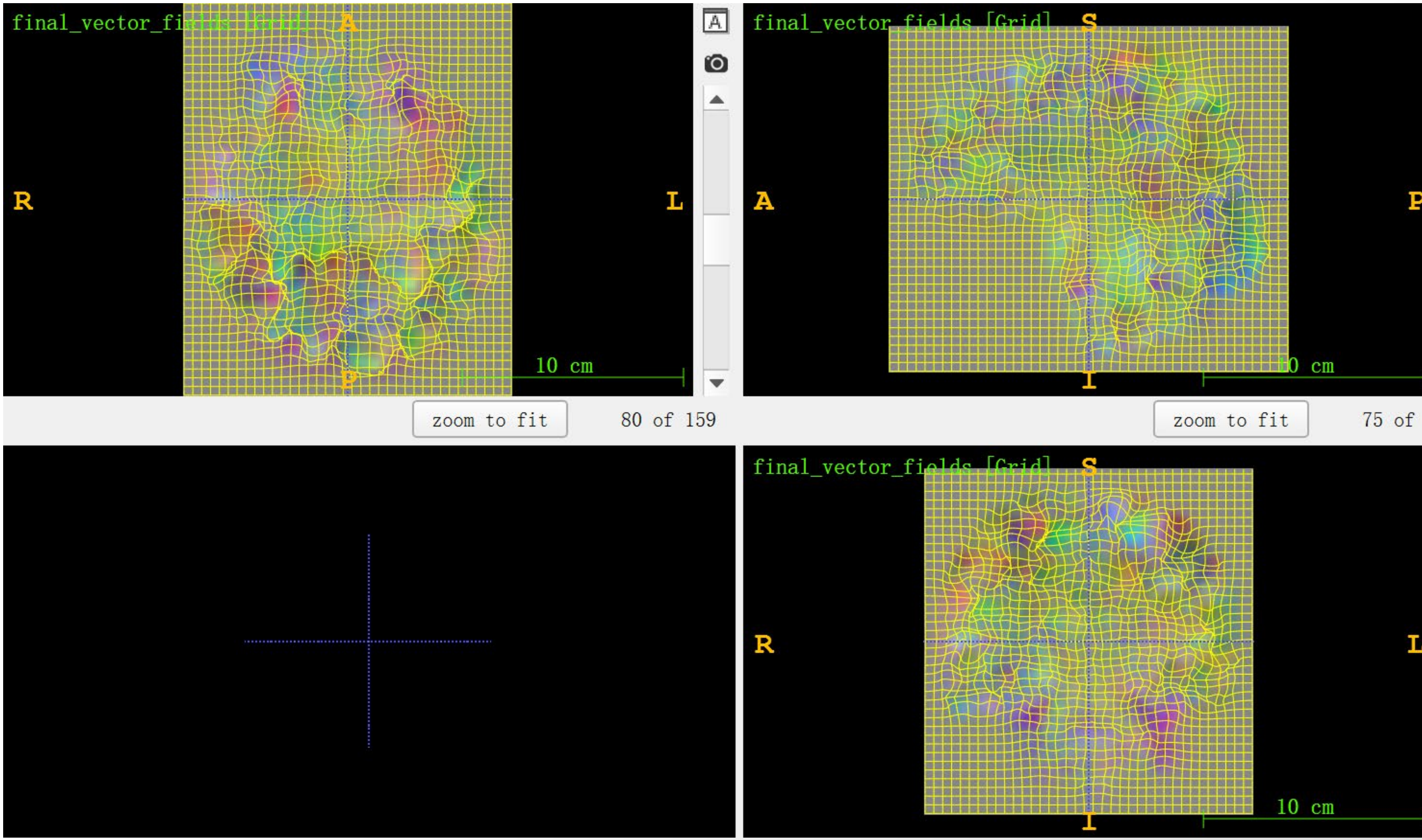
大数据学院 School of Data Science

- **Results visualization.** Vector fields (+ & ∇F)



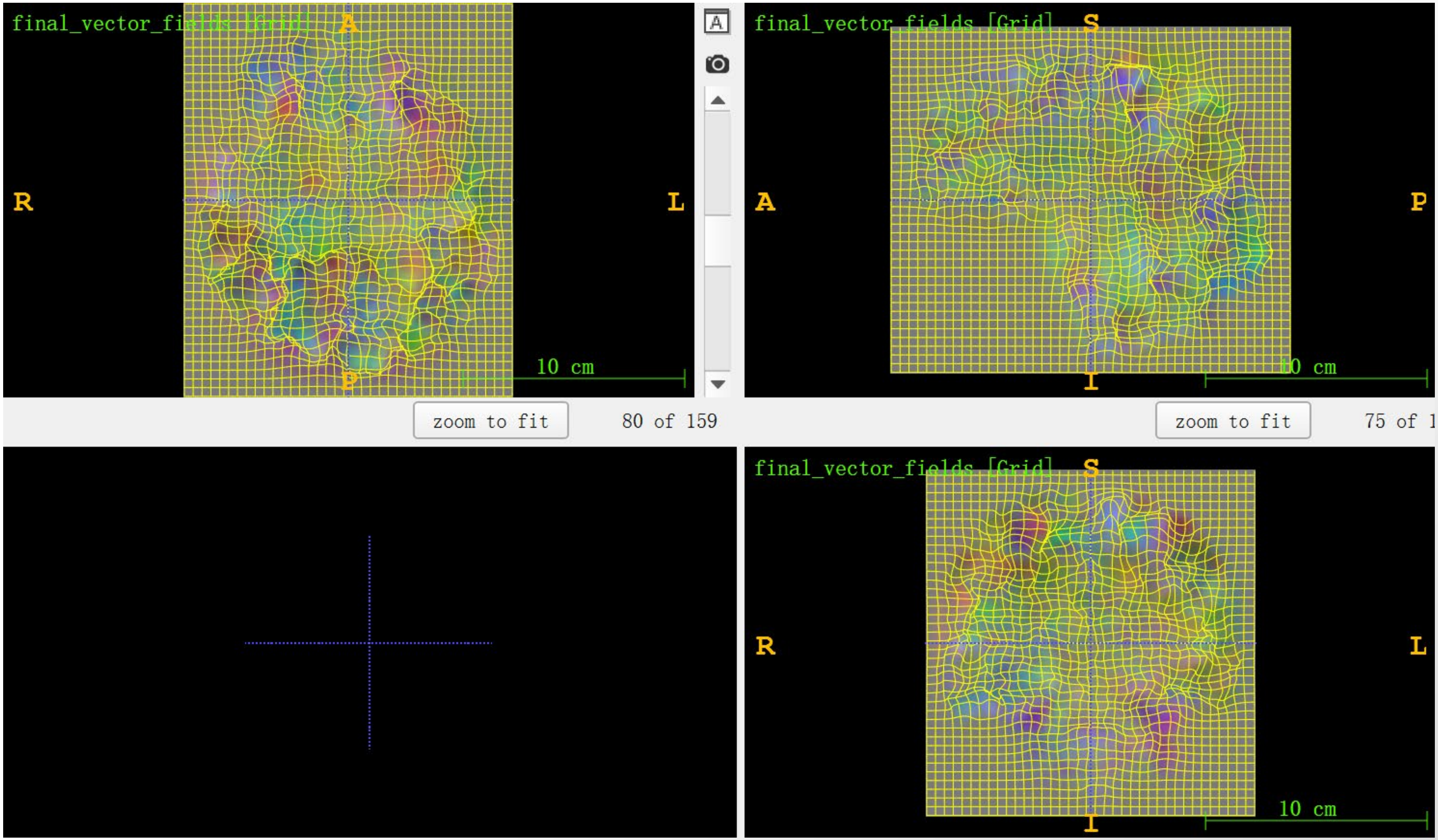
Demo: Diffeomorphic demons

- **Results visualization.** Vector fields (diff & ∇F)



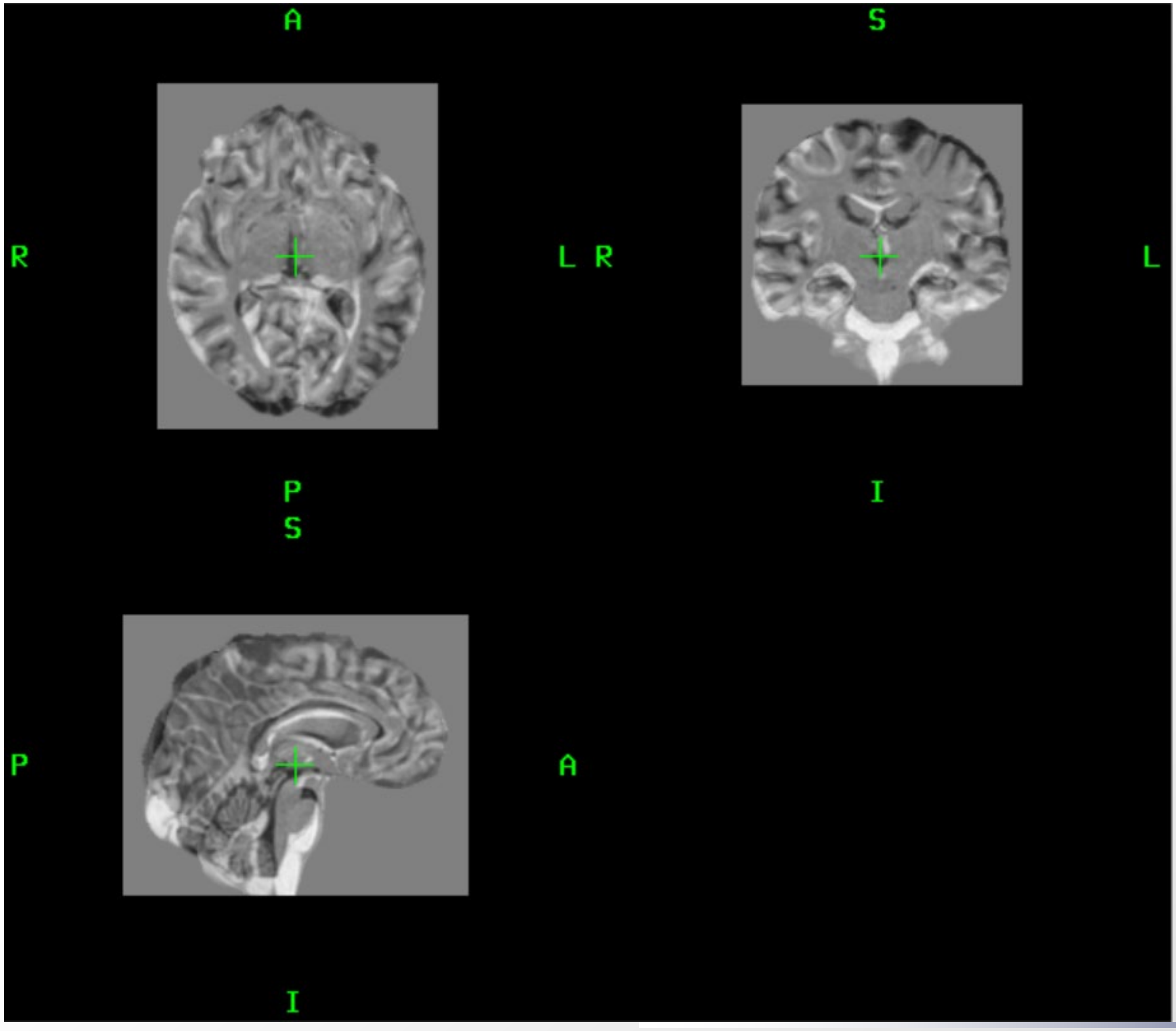
Demo: Diffeomorphic demons

- **Results visualization.** Vector fields (diff & sym)



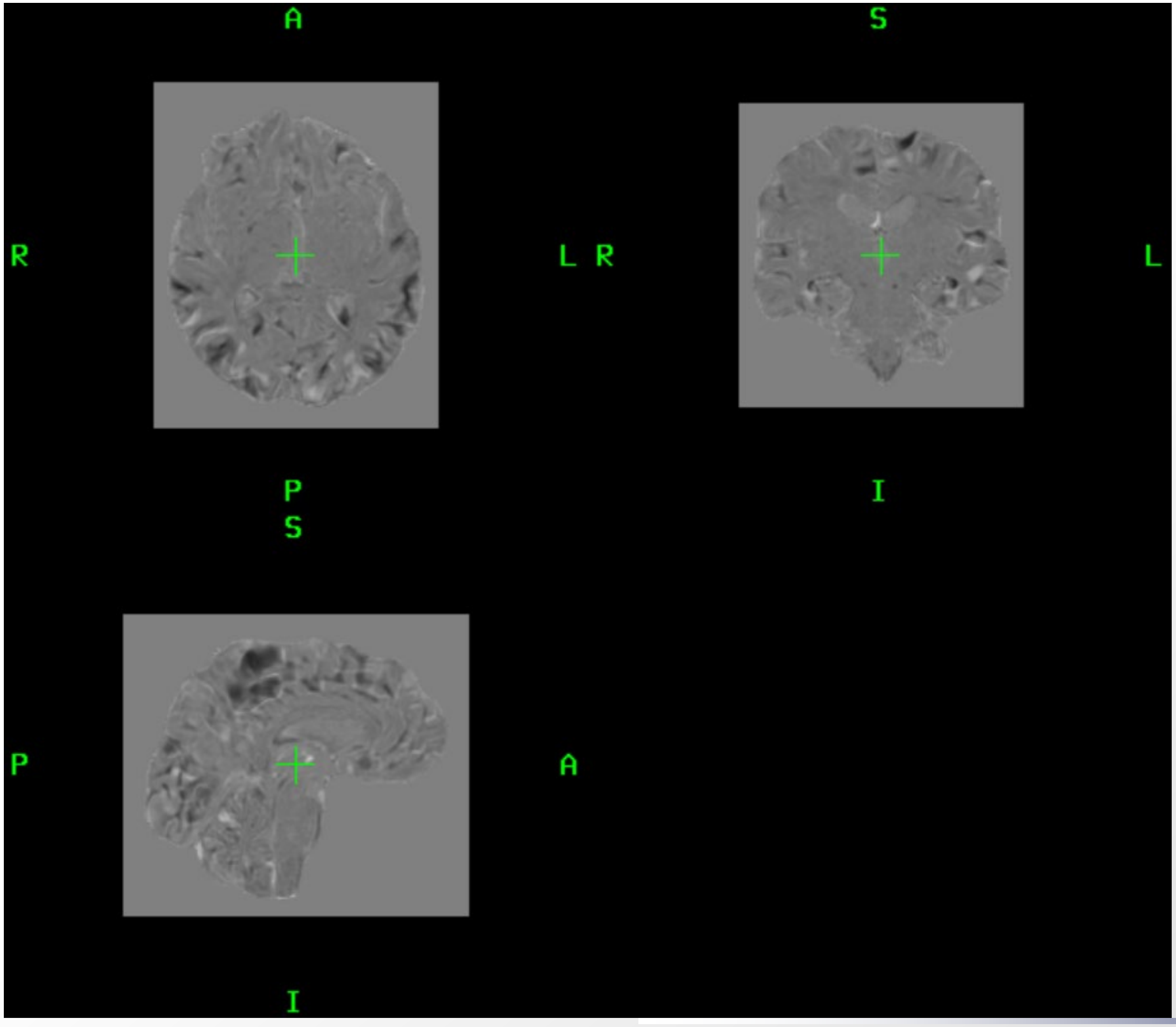
Demo: Diffeomorphic demons

- **Results visualization.** Before registration



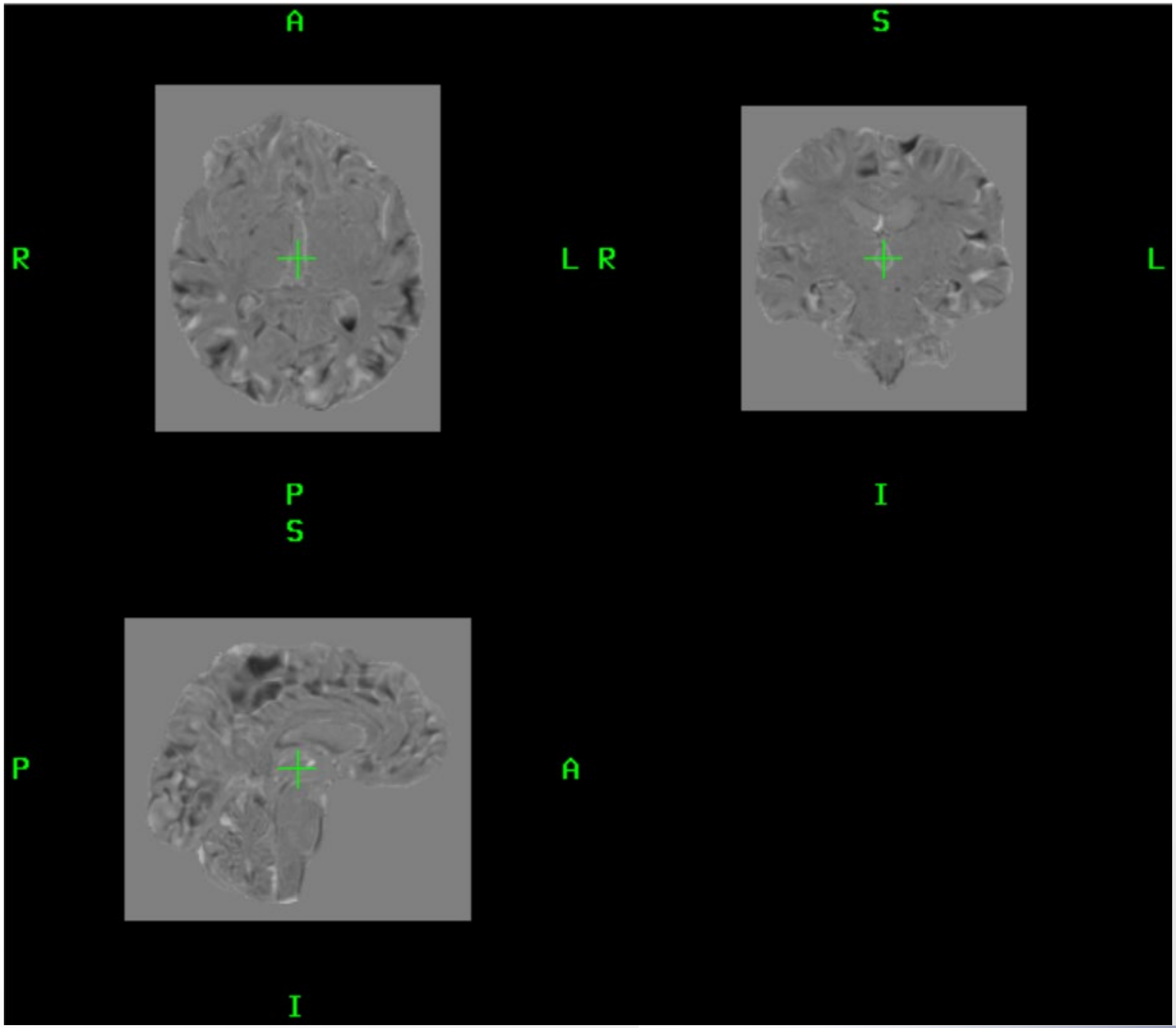
Demo: Diffeomorphic demons

- **Results visualization.** Difference image (+ & ∇F)



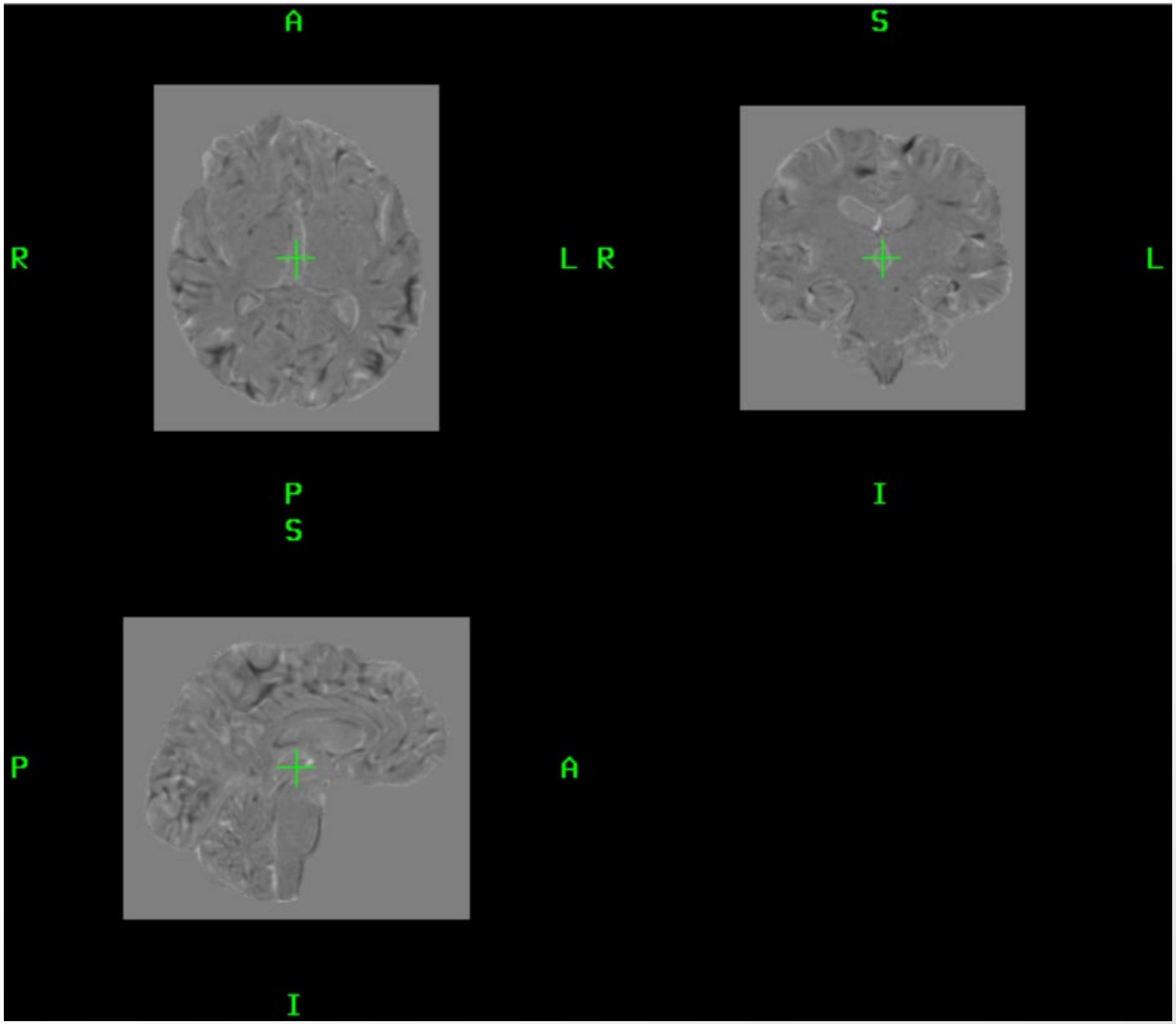
Demo: Diffeomorphic demons

- **Results visualization.** Difference image (diff & ∇F)



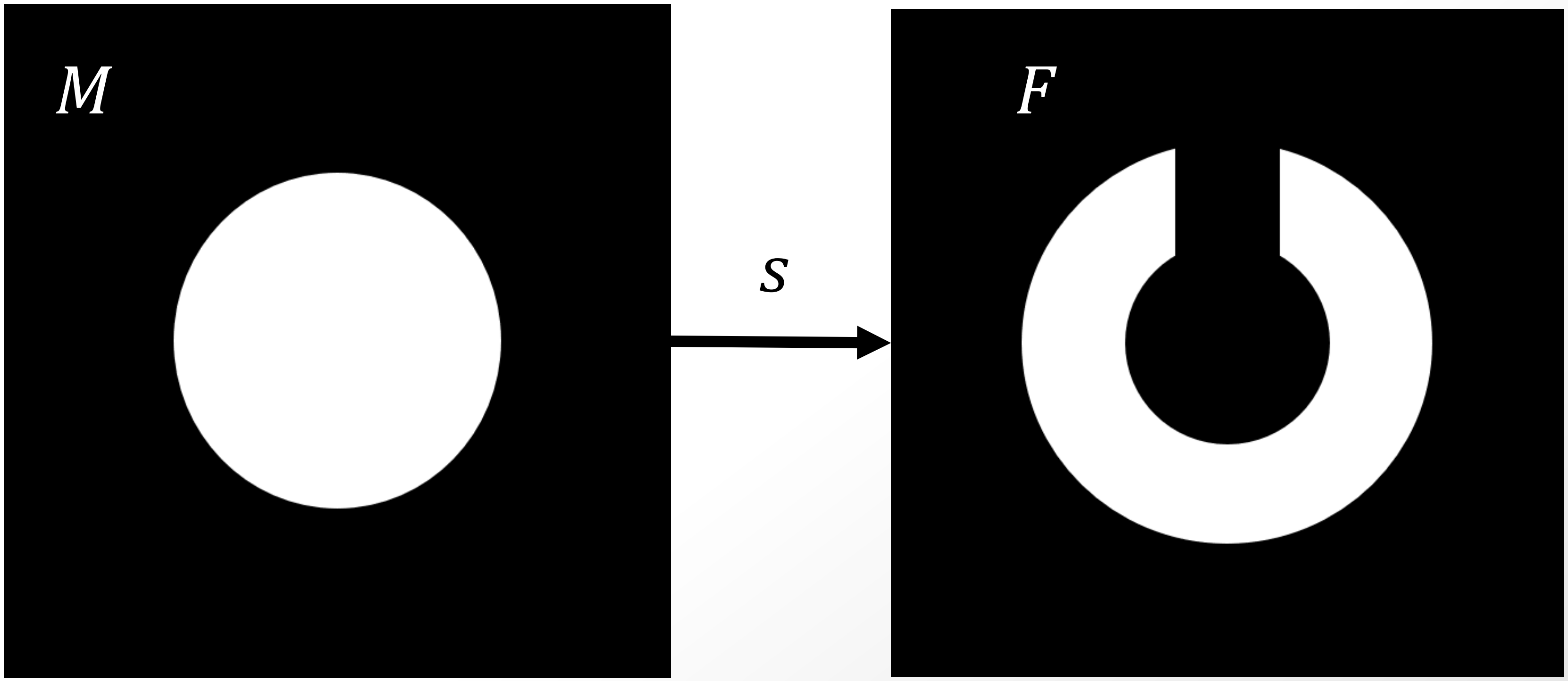
Demo: Diffeomorphic demons

- **Results visualization.** Difference image (diff & sym)



Demo: Diffeomorphic demons

- **Circle to C registration problem.**

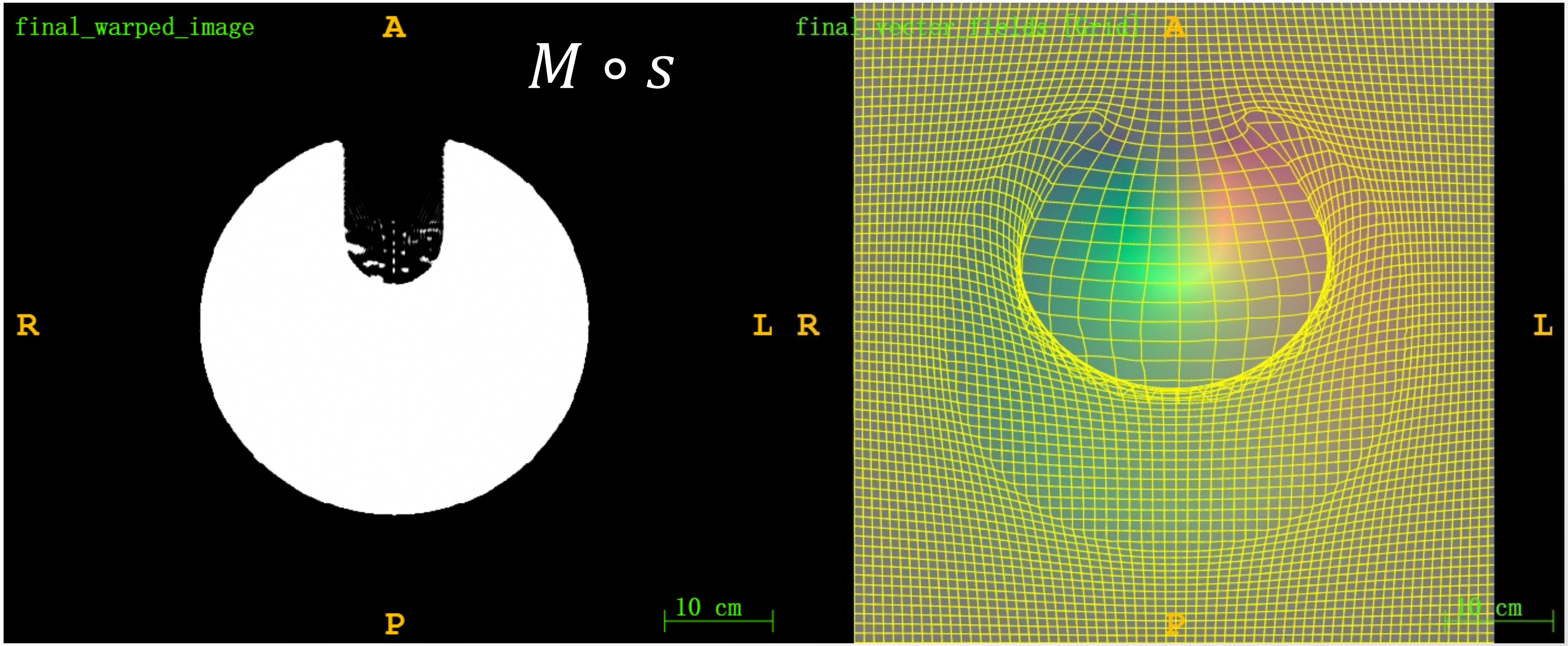


- Registration using additive demons and diffeomorphic demons as an evaluation for introducing diffeomorphisms.

Credits to YUE Qian for drawing the above pictures

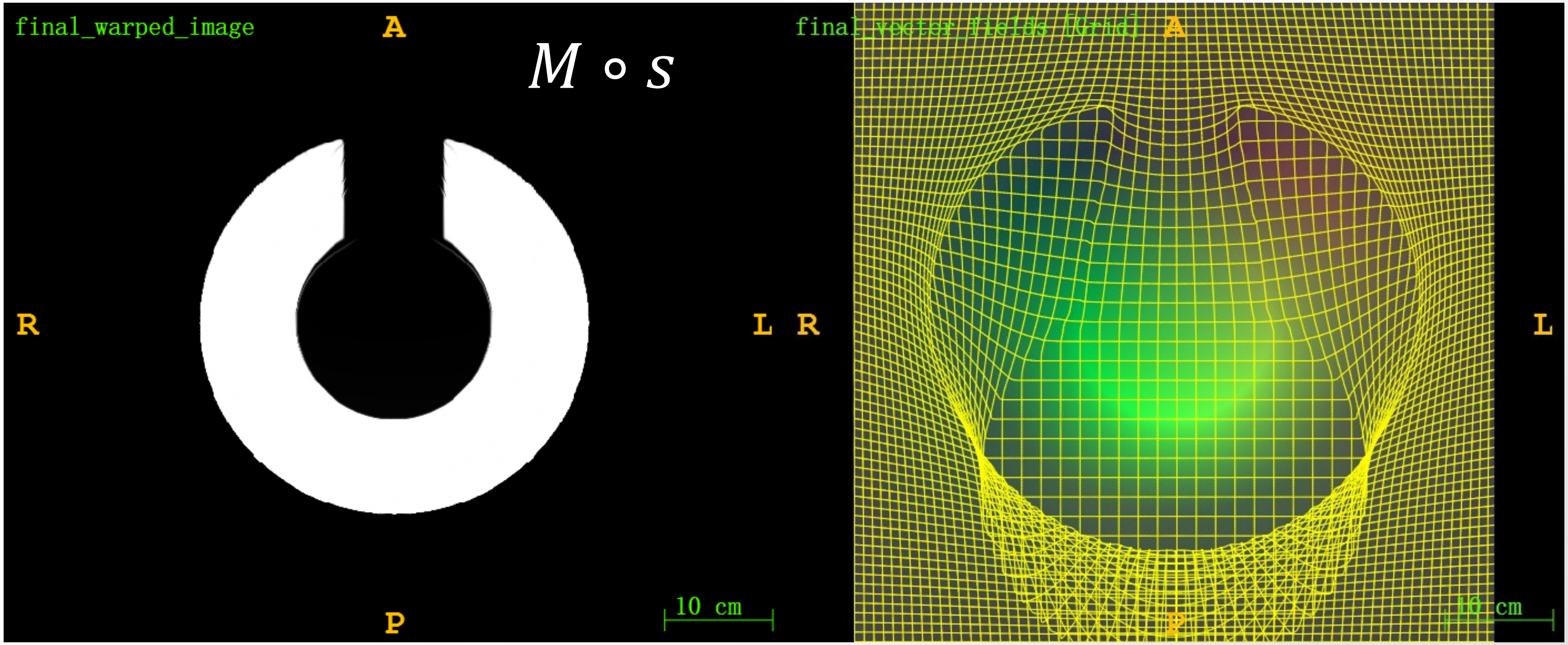
Demo: Diffeomorphic demons

- **Additive demons with $J^p = \nabla_p^T M \circ s$.**
- After 20000 iterations.



Demo: Diffeomorphic demons

- **Diffeomorphic demons with $J^p = \nabla_p^T M \circ s$.**
- After 20000 iterations.



References

大数据学院 School of Data Science

- Hill, D. L. G., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, *46*(3), R1–R45. https://doi.org/10.1088/0031-9155/46/3/201
- Vercauteren, T., Pennec, X., Perchant, A., & Ayache, N. (2009). Diffeomorphic demons: efficient non-parametric image registration. *NeuroImage*, *45*(1 Suppl), S61–S72. https://doi.org/10.1016/j.neuroimage.2008.10.040
- Thirion, J. P. (1998). Image matching as a diffusion process: An analogy with Maxwell's demons. *Medical Image Analysis*, *2*(3), 243–260. https://doi.org/10.1016/S1361-8415(98)80022-4

References

- Arsigny, V., Commowick, O., Pennec, X., & Ayache, N. (2006). A Log-Euclidean Framework for Statistics on Diffeomorphisms. In *Medical image computing and computer-assisted intervention : MICCAI 2006* (Vol. 9, pp. 924–931). https://doi.org/10.1007/11866565_113
- Dalca, A. V., Balakrishnan, G., Guttag, J., & Sabuncu, M. R. (2019). Unsupervised Learning of Probabilistic Diffeomorphic Registration for Images and Surfaces. https://doi.org/10.1016/j.media.2019.07.006
- Dr. Ulas Bagci, Lecture 15: Medical Image Registration I, Medical Image Computing (CAP 5937), CRCV, University of Central Florida, Spring 2017

References

大数据学院
School of Data Science

- Tom Vercauteren, Demons: A Deformable Registration Toolbox, MICCAI 2010 Tutorial: Intensity-based Deformable Registration
- Prof. Chuck Stewart, RPI, Dr. Luis Ibanez, Kitware , Lecture 20: Demons Registration, CSci 6971: Image Registration, Rensselaer Polytechnic Institute (RPI), April 16, 2004

- Credits to YUE Qian for drawing the pictures in page 84

復旦大學
FUDAN UNIVERSITY





Thank You！