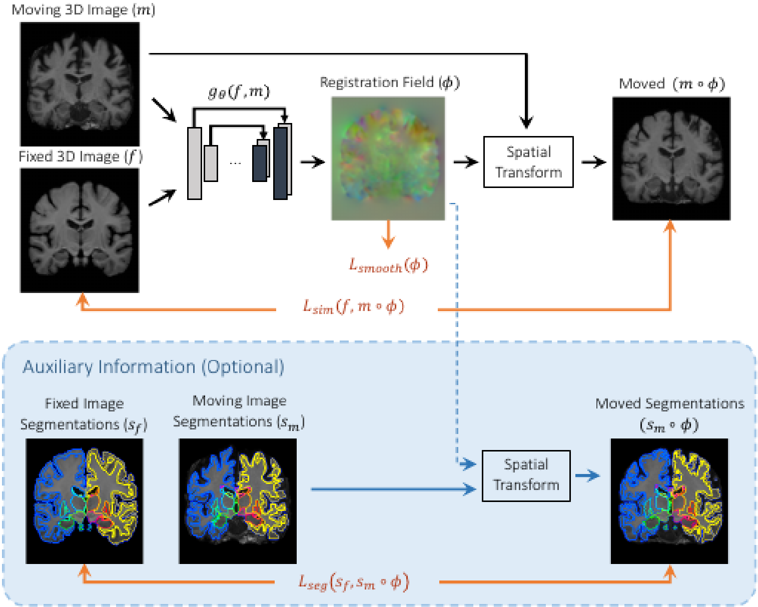
VoxelMorph

Balakrishnan et al. IEEE TMI 2019

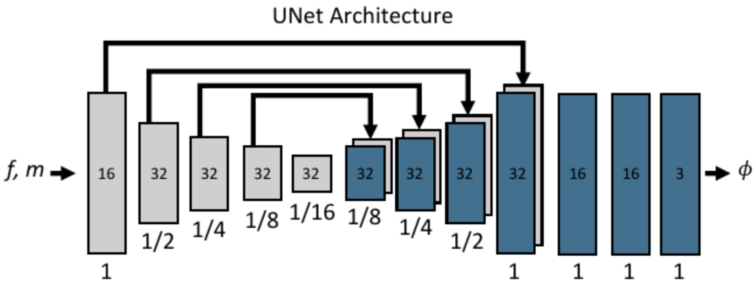
- Input images f, m
- Looking for $\hat{\phi} = \arg\min_{\phi} L_{\text{sim}}(f, m \circ \phi) + \lambda L_{\text{smooth}}(\phi)$
- Learn a network to estimate $\hat{\phi} = g(f, m)$
- L_{sim} is SSD or CC. $L_{\text{smooth}}(\phi) = \|\nabla \mathbf{u}\|^2, \quad \phi(\mathrm{x}) = \mathrm{x} + \mathrm{u}$
- Linear interpolation to calculate $m \circ \phi$
- Very fast and accurate
- Handles small deformations, assumes affine preregistration

`http://voxelmorph.csail.mit.edu`

Voxelmorph overview



Voxelmorph network



leaky ReLU, 3D convolutions

Spatial transformation

For each voxel $\mathbf{p}$, we compute a (subpixel) voxel location $\mathbf{p}' = \mathbf{p} + \mathbf{u}(\mathbf{p})$ in m. Because image values are only defined at integer locations, we linearly interpolate the values at the eight neighboring voxels:

$$m \circ \boldsymbol{\phi}(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{Z}(\mathbf{p}')} m(\mathbf{q}) \prod_{d \in \{x,y,z\}} (1 - |\mathbf{p}'_d - \mathbf{q}_d|), \qquad (3)$$

where $\mathcal{Z}(\mathbf{p}')$ are the voxel neighbors of $\mathbf{p}'$, and d iterates over dimensions of Ω. Because we can compute gradients or sub-

Loss function

$$\mathcal{L}_{us}(f, m, \phi) = \mathcal{L}_{sim}(f, m \circ \phi) + \lambda \mathcal{L}_{smooth}(\phi),$$

$$MSE(f, m \circ \phi) = \frac{1}{|\Omega|} \sum_{p \in \Omega} [f(\mathbf{p}) - [m \circ \phi](\mathbf{p})]^2 .$$

$$CC(f, m \circ \phi) = \sum_{\mathbf{p} \in \Omega} \frac{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \hat{f}(\mathbf{p}))([m \circ \phi](\mathbf{p}_i) - [\hat{m} \circ \phi](\mathbf{p})) \right)^2}{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \hat{f}(\mathbf{p}))^2 \right) \left(\sum_{\mathbf{p}_i} ([m \circ \phi](\mathbf{p}_i) - [\hat{m} \circ \phi](\mathbf{p}))^2 \right)} .$$

$$\mathcal{L}_{smooth}(\phi) = \sum_{\mathbf{p} \in \Omega} \|\nabla \mathbf{u}(\mathbf{p})\|^2,$$

Auxiliary loss function

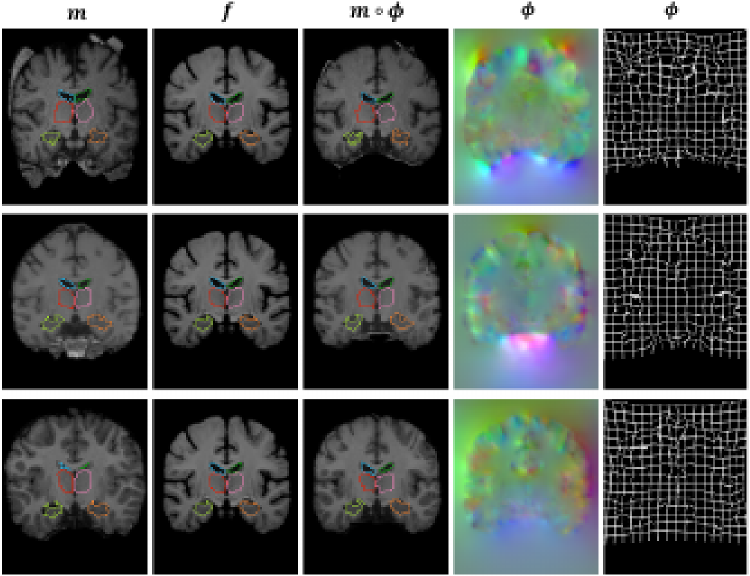
$$\text{Dice}(s_f^k, s_m^k \circ \phi) = 2 \cdot \frac{|s_f^k \cap (s_m^k \circ \phi)|}{|s_f^k| + |s_m^k \circ \phi|}. \qquad (8)$$

A Dice score of 1 indicates that the anatomy matches perfectly, and a score of 0 indicates that there is no overlap. We define the segmentation loss $\mathcal{L}_{seg}$ over all structures $k \in [1, K]$ as:

$$\mathcal{L}_{seg}(s_f, s_m \circ \phi) = -\frac{1}{K} \sum_{k=1}^{K} \text{Dice}(s_f^k, s_m^k \circ \phi). \qquad (9)$$

Only at training time

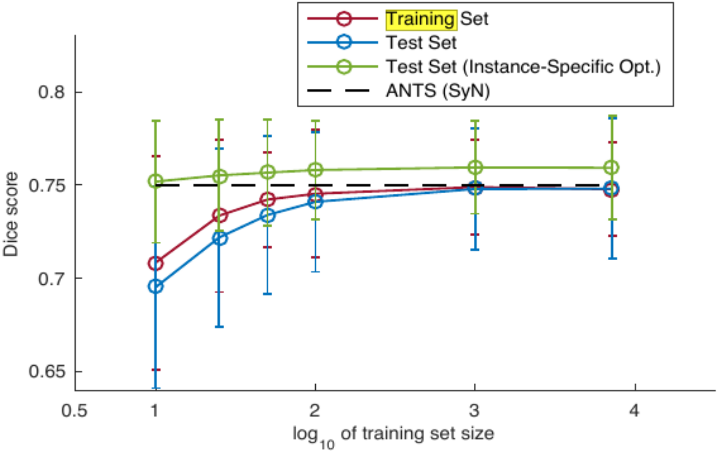
Example images



Quantitative results

Method	Dice	GPU sec	CPU sec	$\lvert J_\phi \rvert \leq 0$	% of $\lvert J_\phi \rvert \leq 0$
Affine only	0.584 (0.157)	0	0	0	0
ANTs SyN (CC)	0.749 (0.136)	-	9059 (2023)	9662 (6258)	0.140 (0.091)
NiftyReg (CC)	0.755 (0.143)	-	2347 (202)	41251 (14336)	0.600 (0.208)
VoxelMorph (CC)	0.753 (0.145)	0.45 (0.01)	57 (1)	19077 (5928)	0.366 (0.114)
VoxelMorph (MSE)	0.752 (0.140)	0.45 (0.01)	57 (1)	9606 (4516)	0.184 (0.087)

Training size



instance-specific optimization = gradient descent