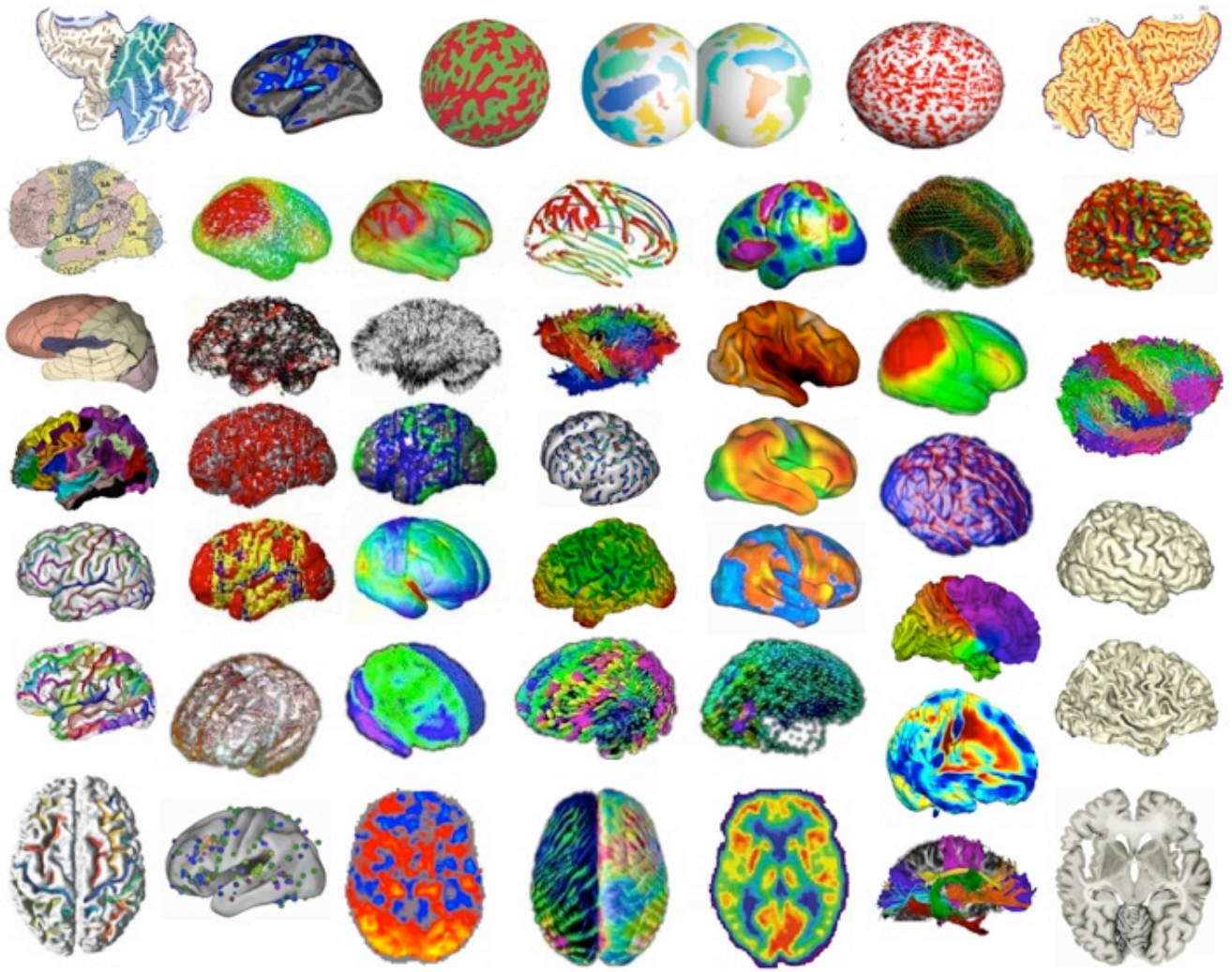
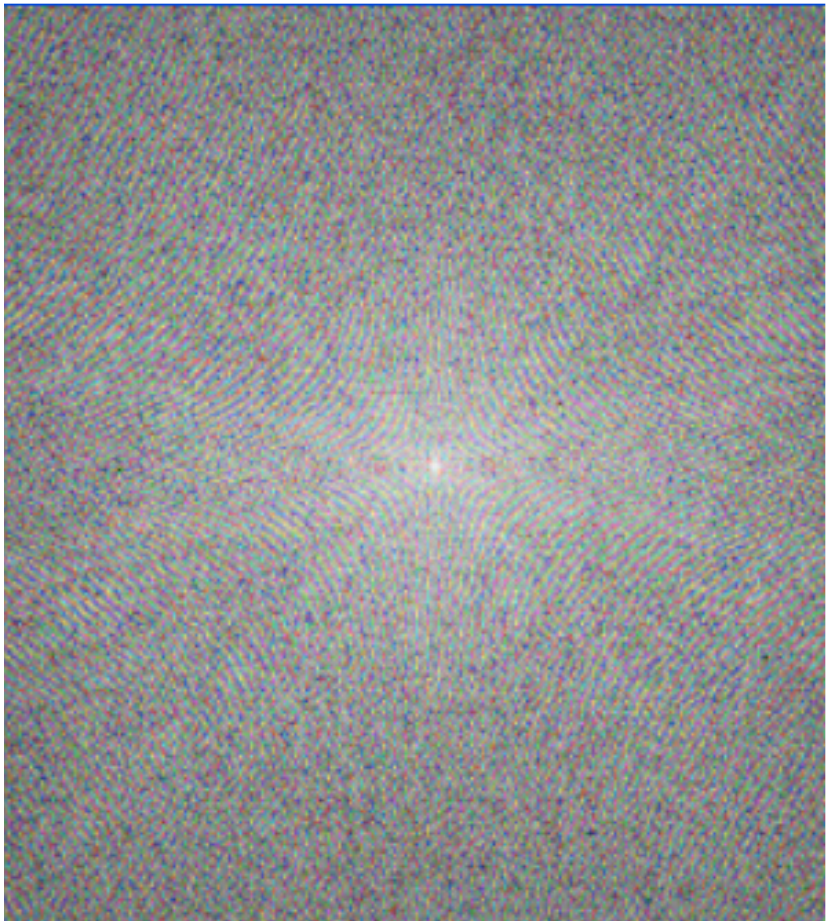
Image Processing and Registration



arno klein
arno@binarybottle.com

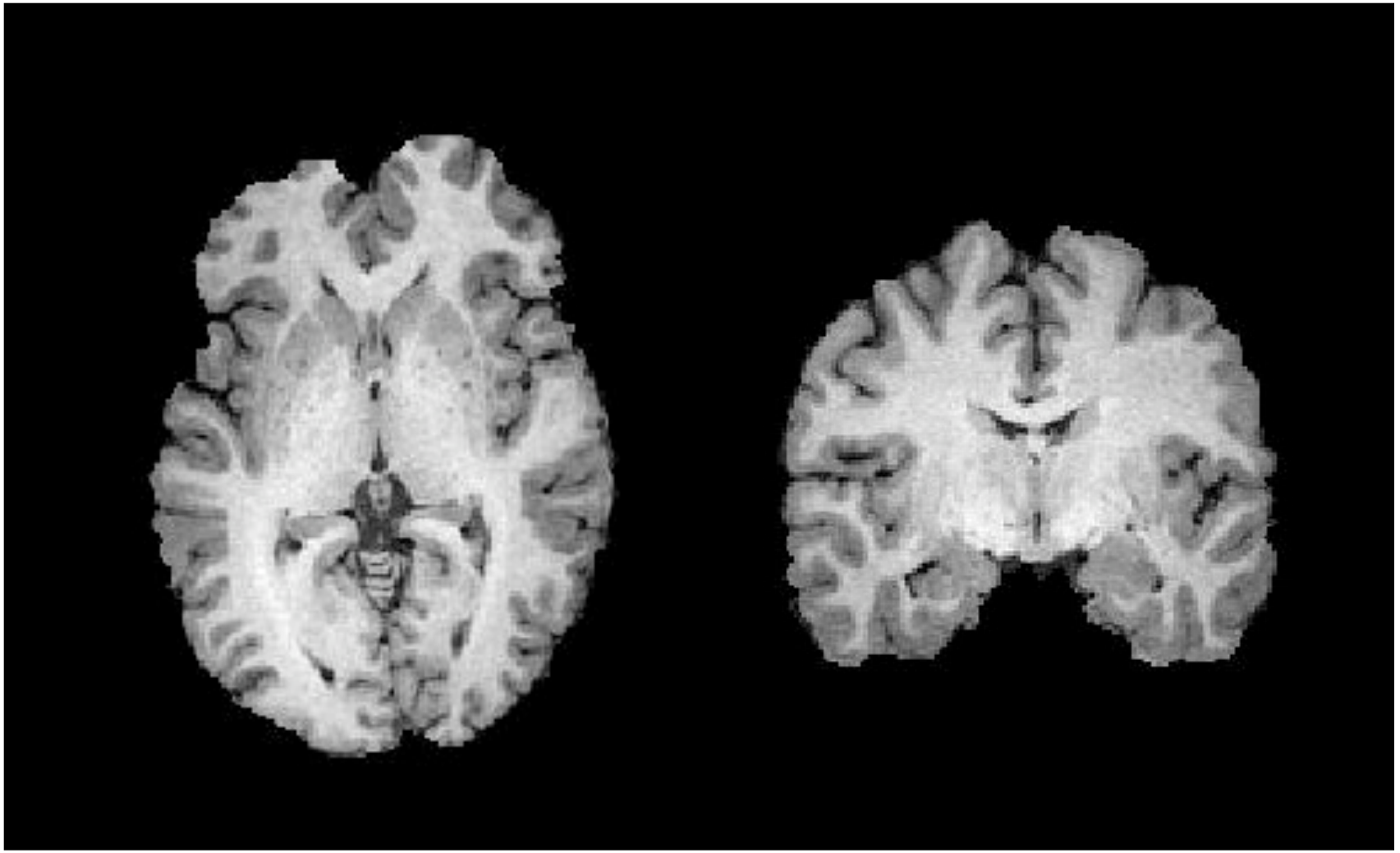
http://www.mindboggle.info/lectures

Re: k-space to image



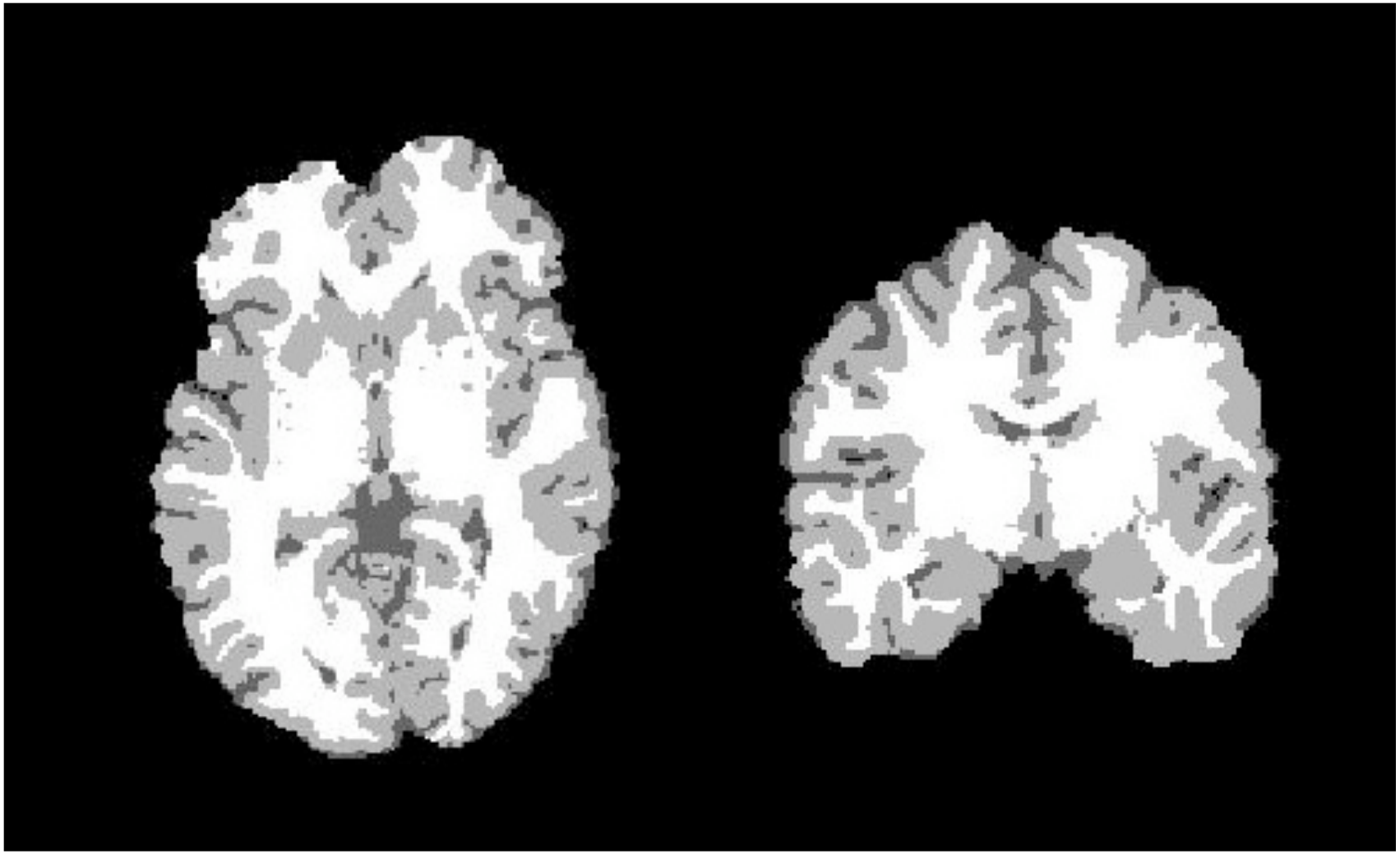
(www.mindboggle.info/lectures)

skull-stripping

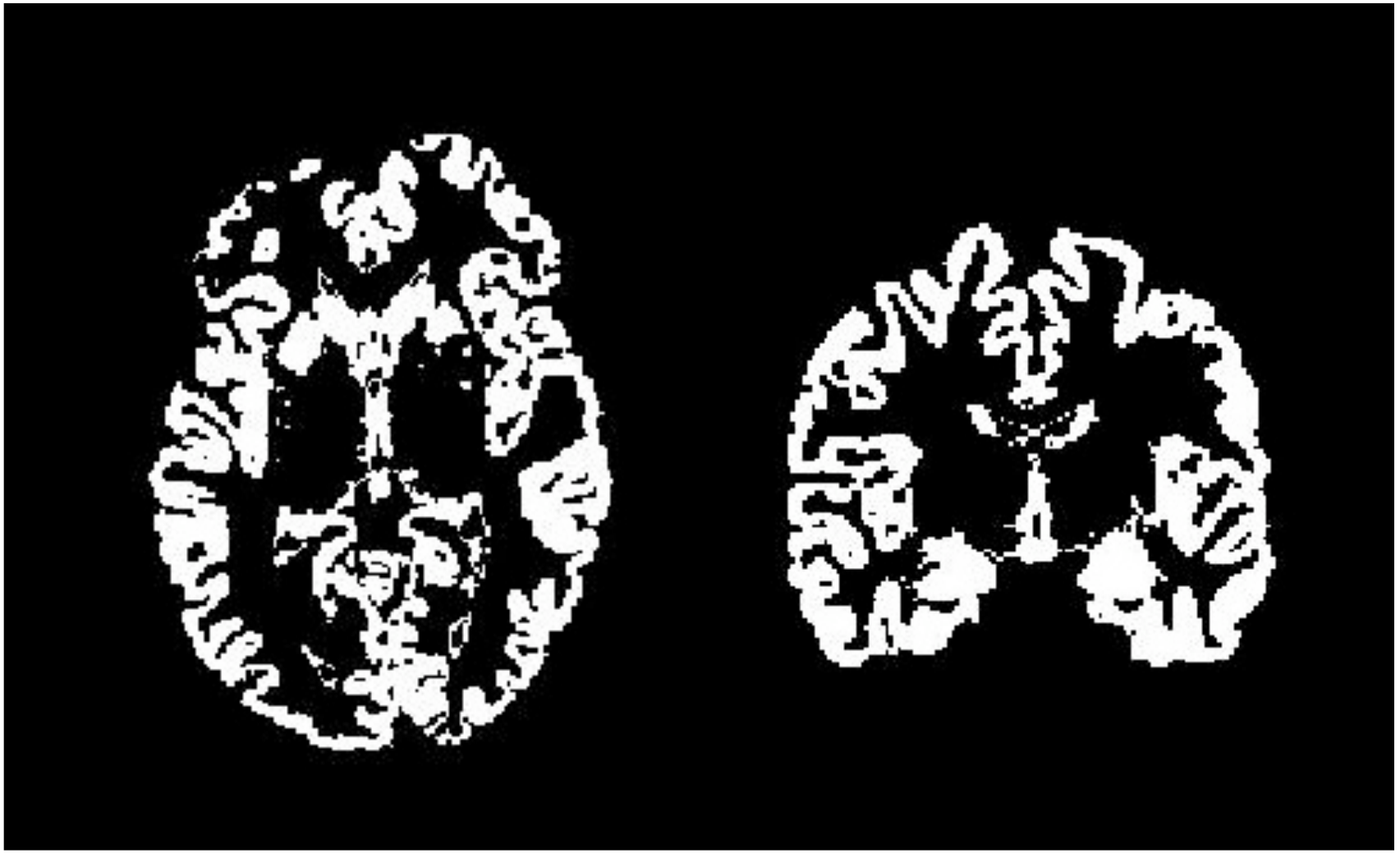


tissue-class segmentation

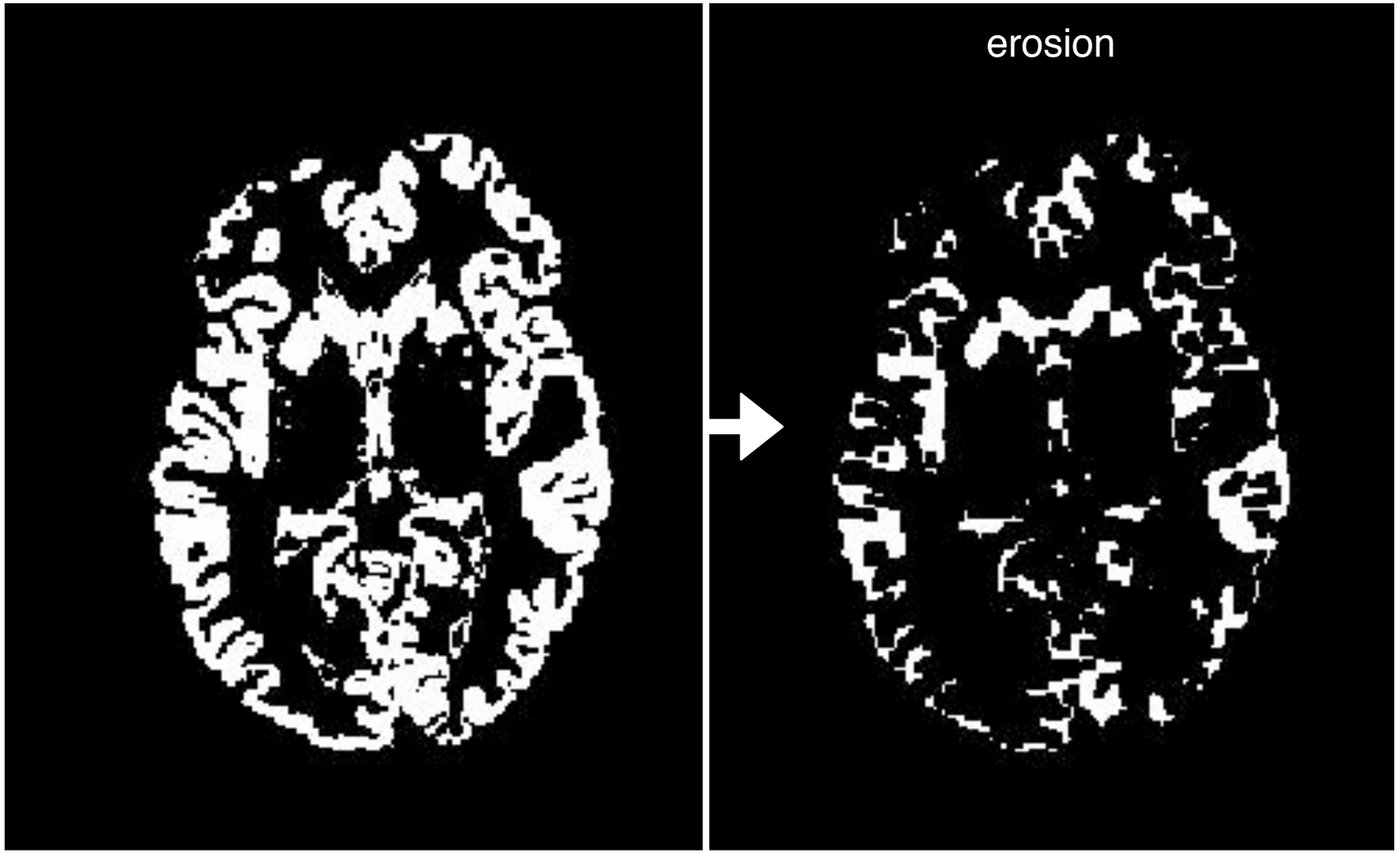
(+ inhomogeneity correction)



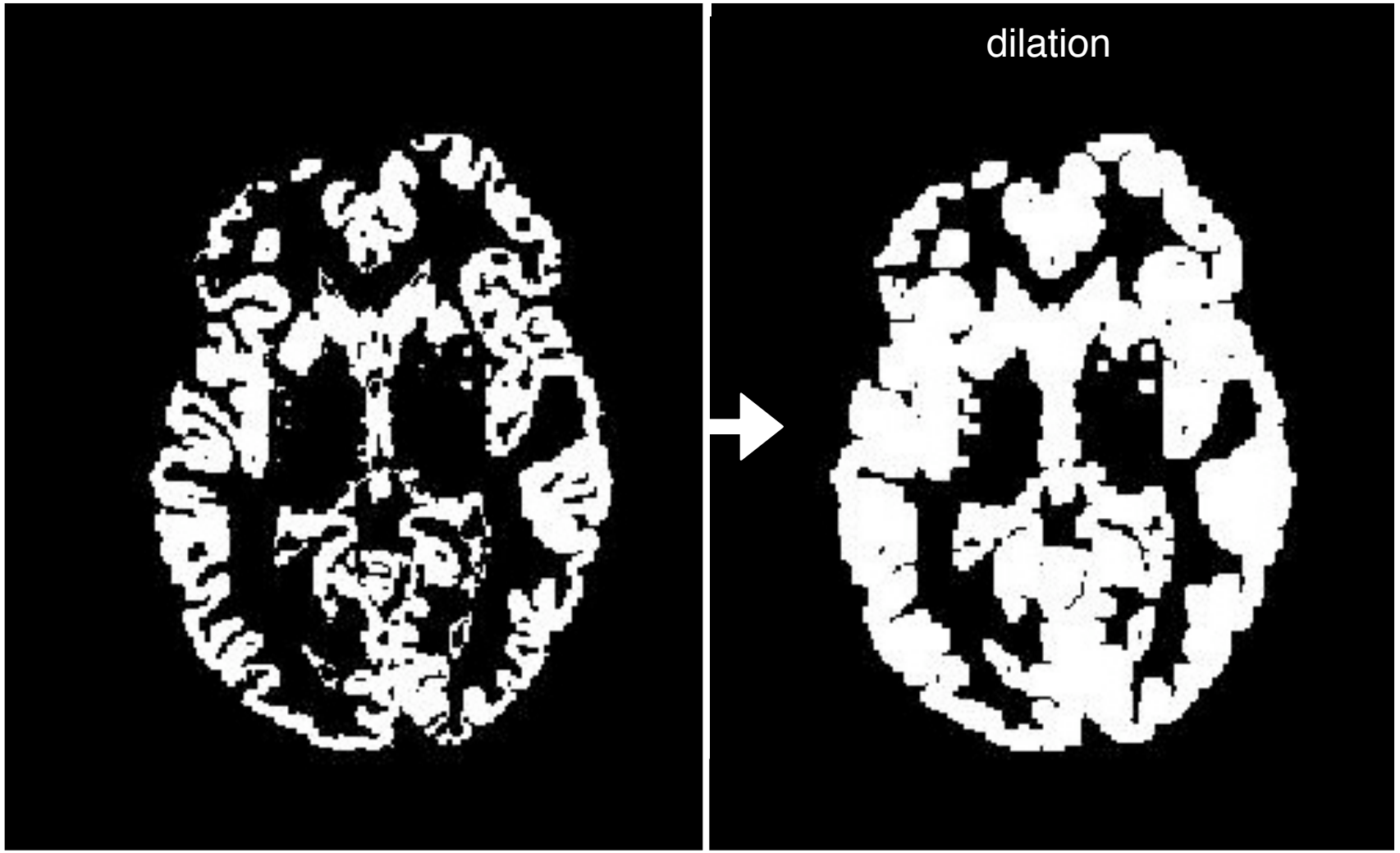
tissue-class segmentation



morphological image processing



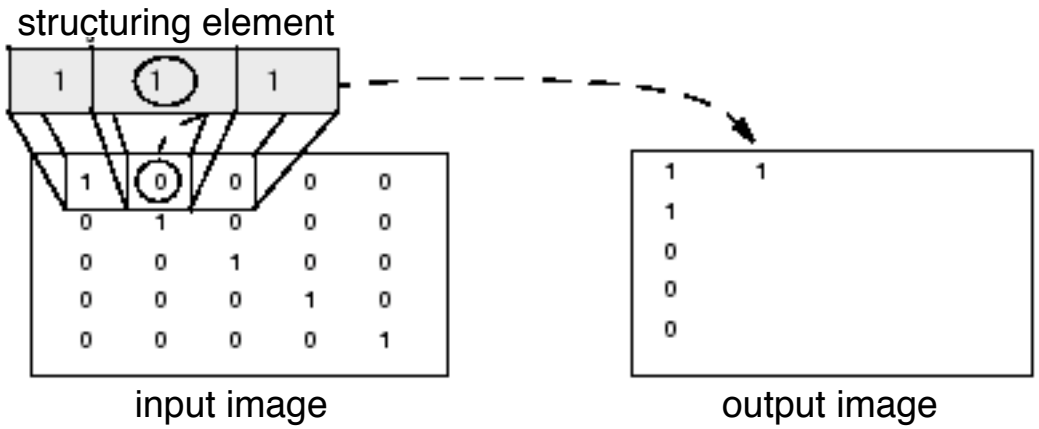
morphological image processing

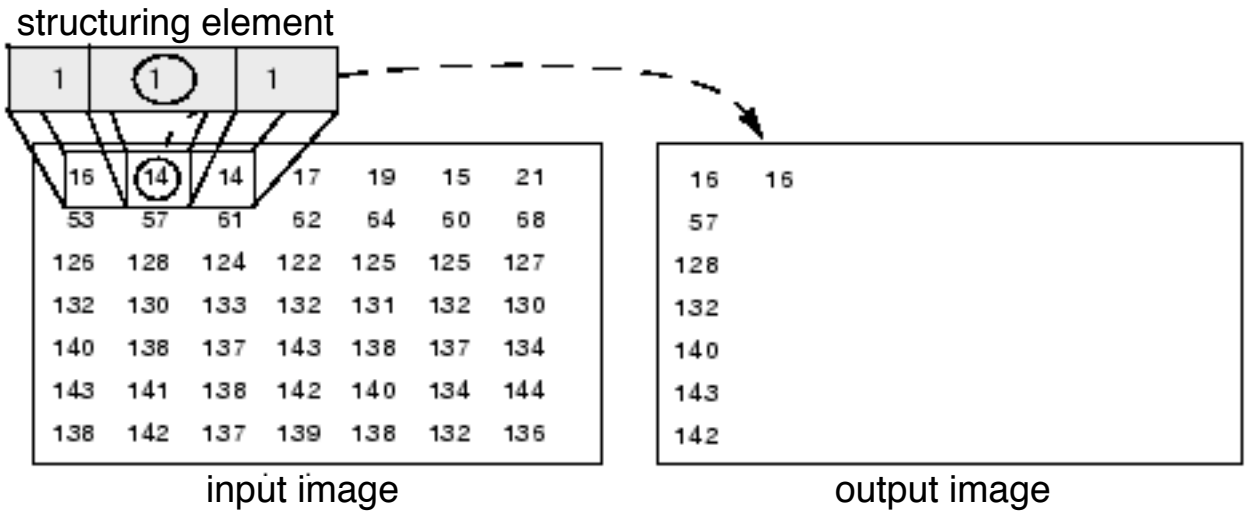


morphological image processing

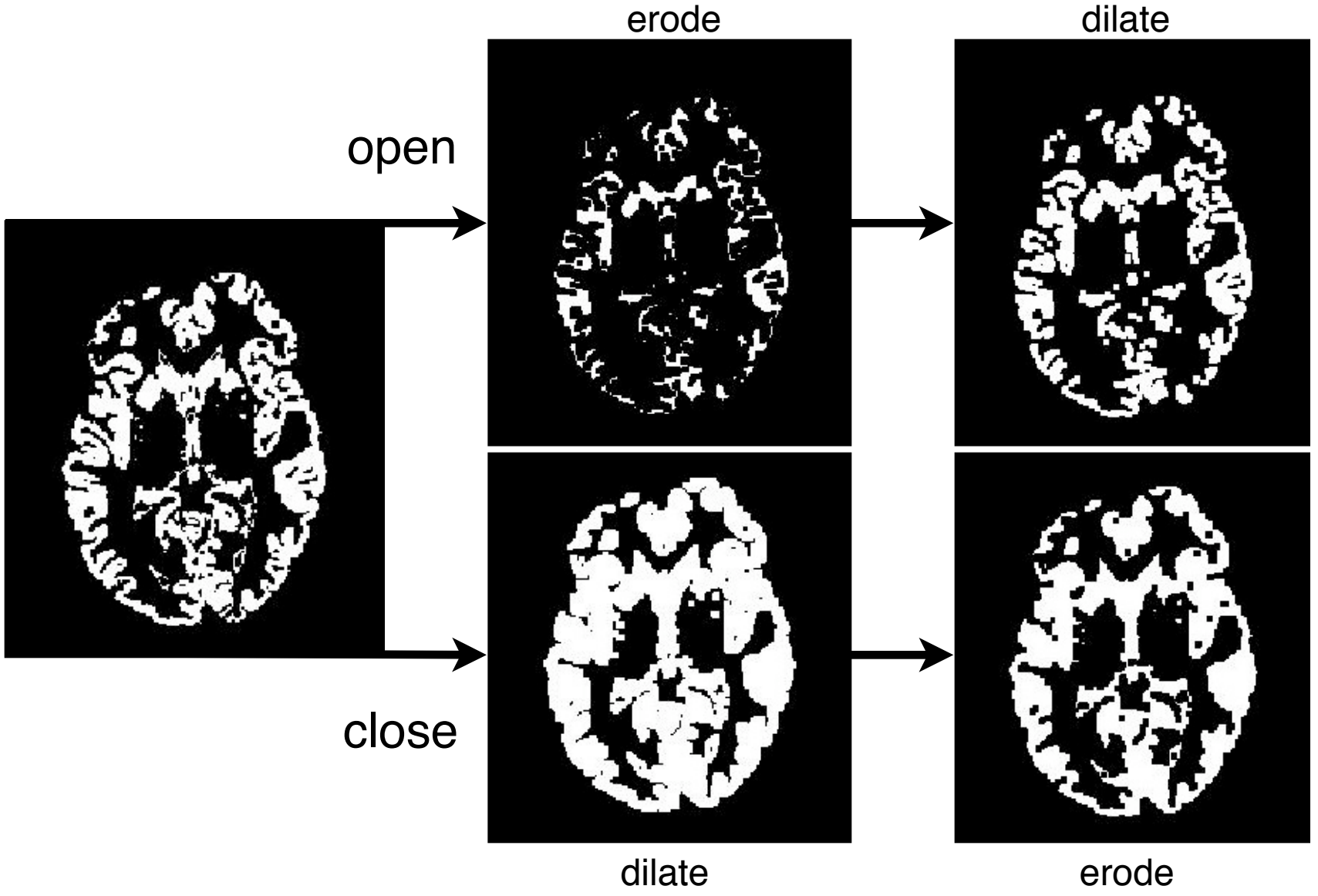
Dilation example

maximum value in neighborhood

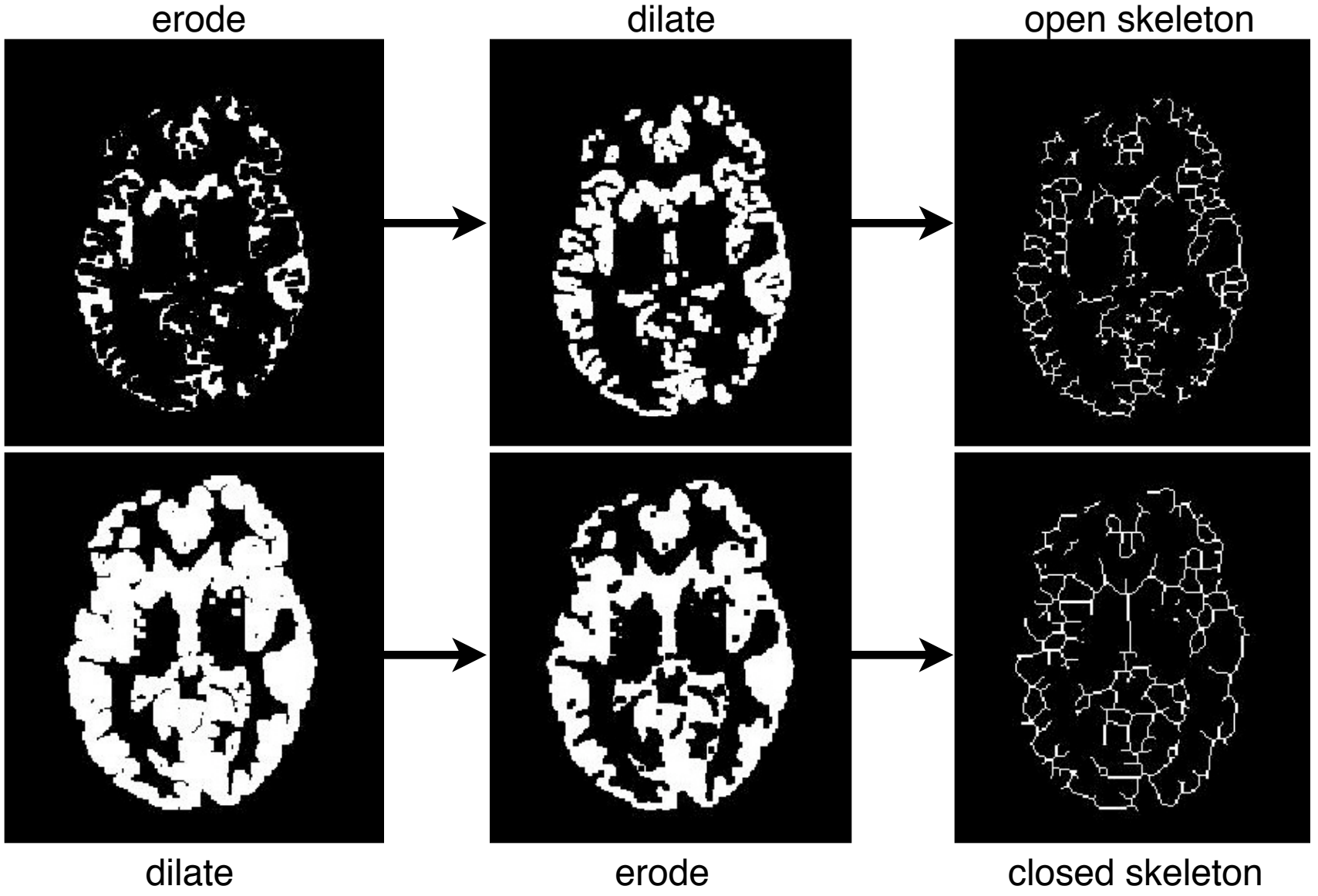




morphological image processing



morphological image processing



morphological reconstruction

Morphological reconstruction:

- Processing is based on two images, a marker and a mask, rather than one image and a structuring element.

- Processing repeats until stability (the image no longer changes).

- Processing is based on the concept of connectivity, rather than a structuring element

Examples:

1. pixel connectivity
2. flood-fill operations
3. peaks and valleys

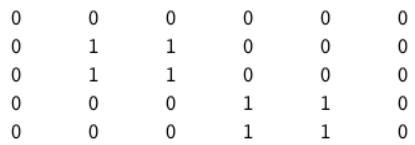
pixel connectivity

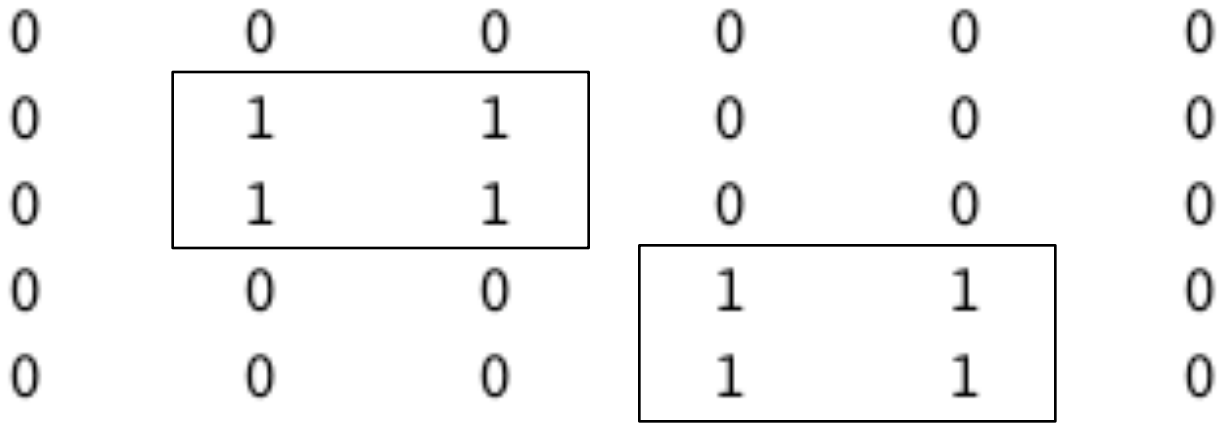
0 0 0 0 0 0
0 1 1 0 0 0
0 1 1 0 0 0
0 0 0 1 1 0
0 0 0 1 1 0

0 0 0 0 0 0
0 1 1 0 0 0
0 1 1 0 0 0
0 0 0 1 1 0
0 0 0 1 1 0

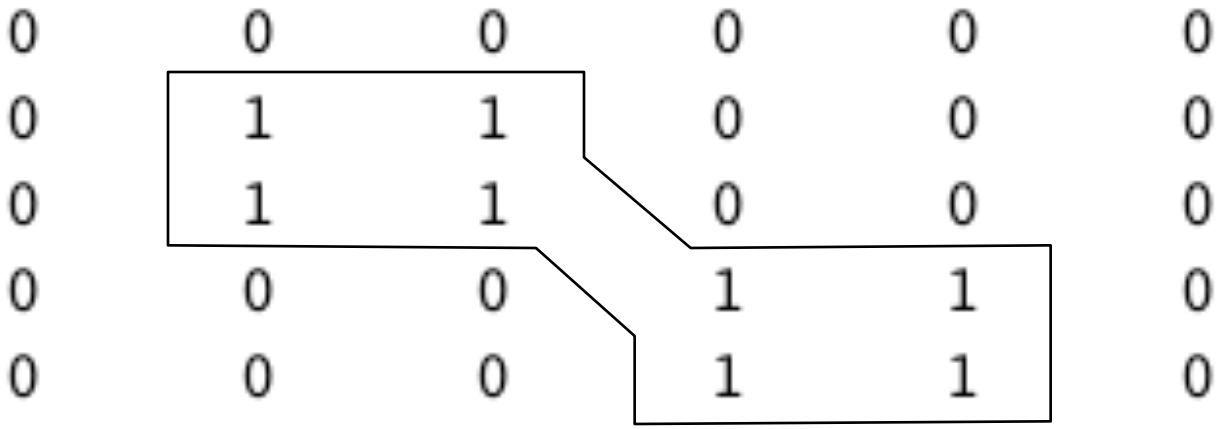
0 0 0 0 0 0
0 1 1 0 0 0
0 1 1 0 0 0
0 0 0 1 1 0
0 0 0 1 1 0

pixel connectivity





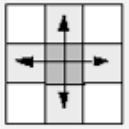
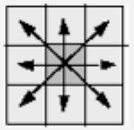
pixel connectivity



0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	1	1	0
0	0	0	1	1	0

pixel connectivity

0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	1	1	0
0	0	0	1	1	0

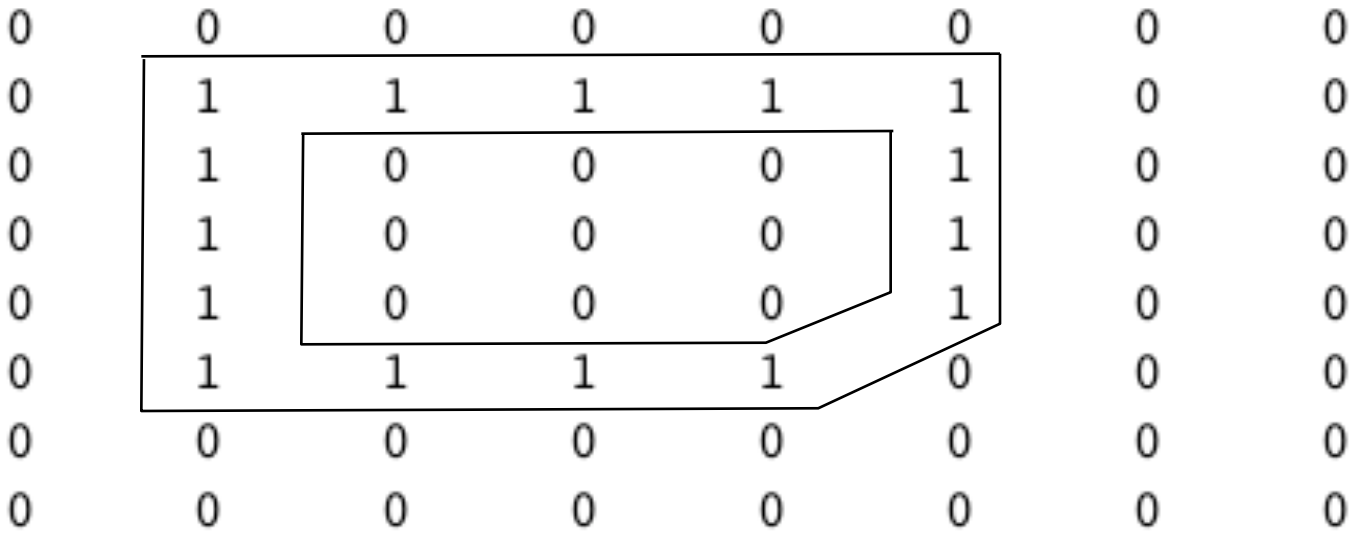
Two-Dimensional Connectivities		
4-connected	Pixels are connected if their edges touch. This means that a pair of adjoining pixels are part of the same object only if they are both on and are connected along the horizontal or vertical direction.	
8-connected	Pixels are connected if their edges or corners touch. This means that if two adjoining pixels are on, they are part of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.	
Three-Dimensional Connectivities		
6-connected	Pixels are connected if their faces touch.	
18-connected	Pixels are connected if their faces or edges touch.	
26-connected	Pixels are connected if their faces, edges, or corners touch.	

flood filling

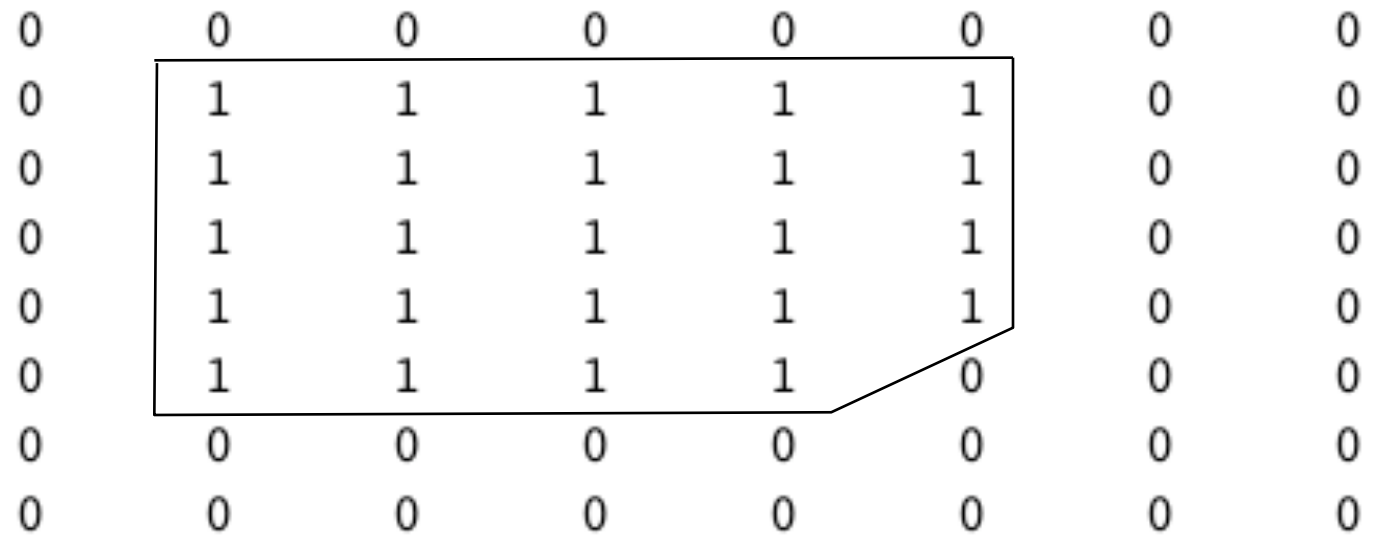
0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

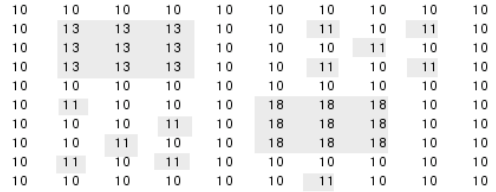
0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

flood filling

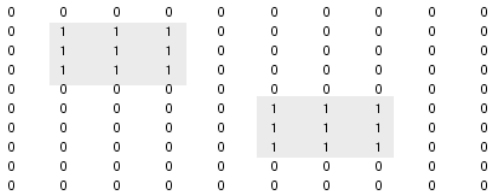


flood filling

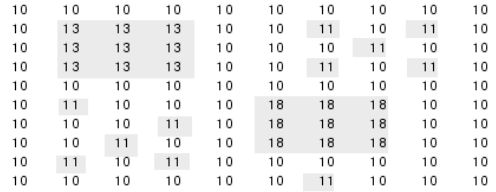




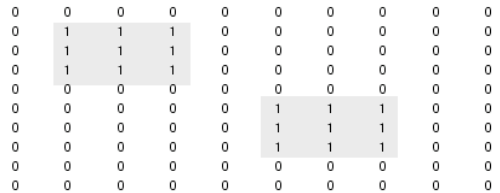
peaks & valleys



10	10	10	10	10	10	10	10	10	10
10	13	13	13	10	10	11	10	11	10
10	13	13	13	10	10	10	11	10	10
10	13	13	13	10	10	11	10	11	10
10	10	10	10	10	10	10	10	10	10
10	11	10	10	10	18	18	18	10	10
10	10	10	11	10	18	18	18	10	10
10	10	11	10	10	18	18	18	10	10
10	11	10	11	10	10	10	10	10	10
10	10	10	10	10	10	11	10	10	10

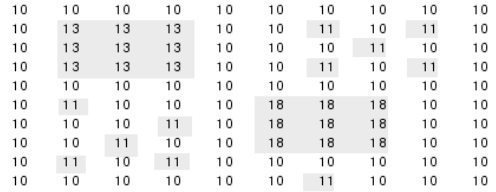


peaks & valleys

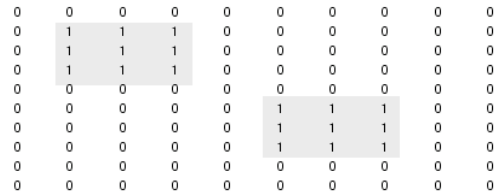


regional maxima

0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0	1	0
0	1	1	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1	0	0
0	0	0	1	0	1	1	1	0	0
0	0	1	0	0	1	1	1	0	0
0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0



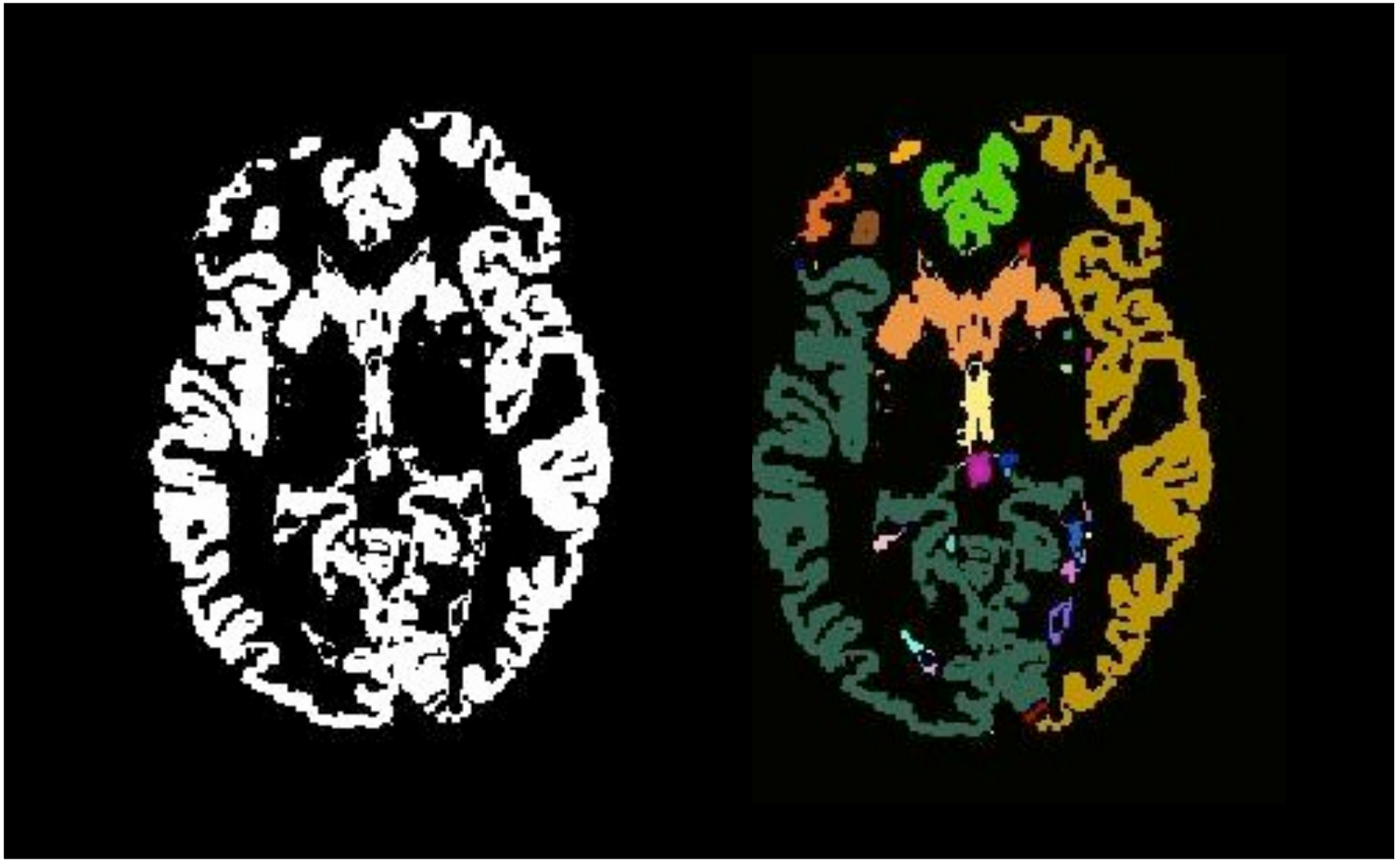
peaks & valleys



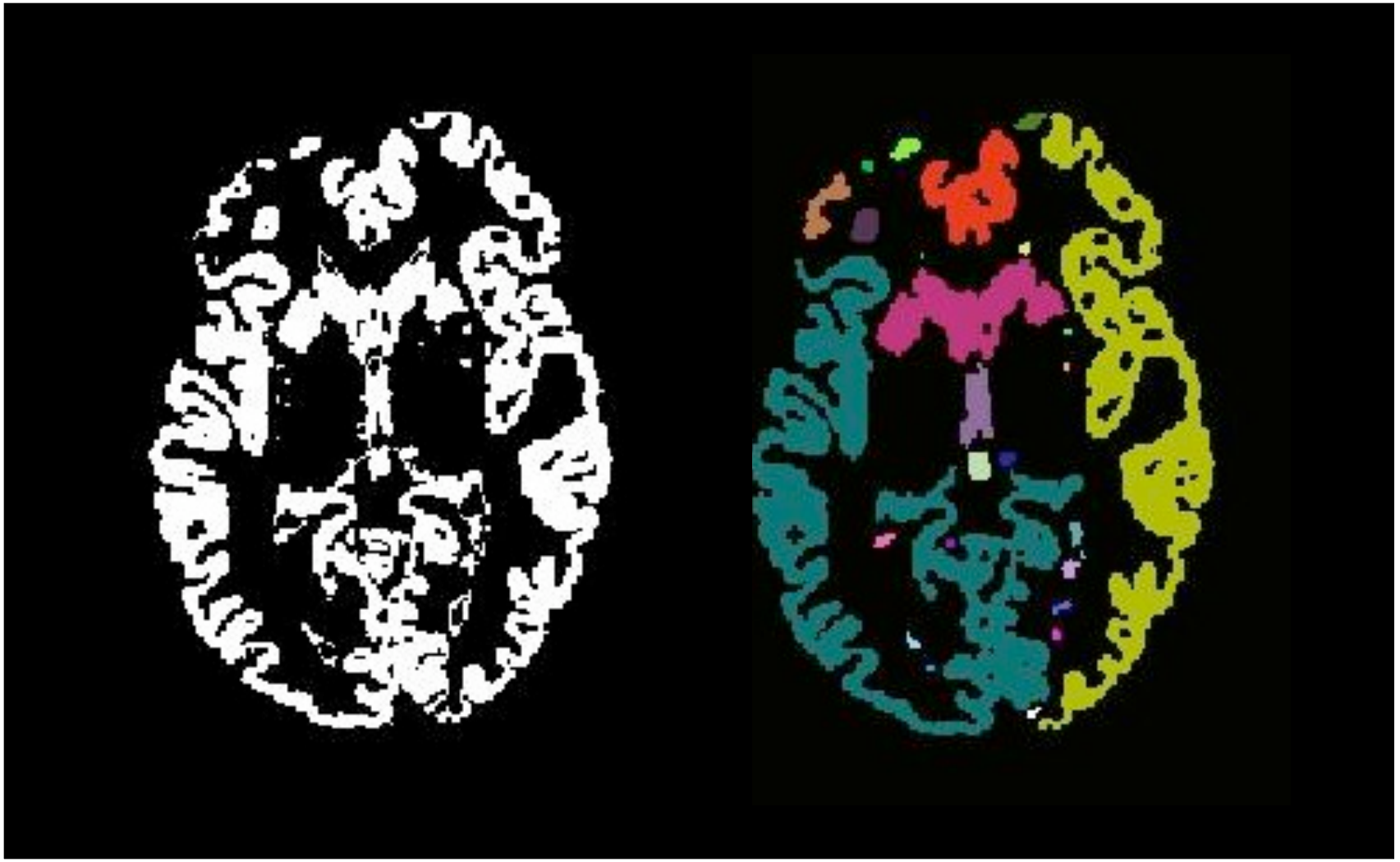
regional maxima, thresholded

0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

labeling connected objects

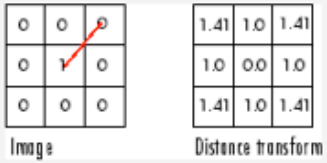
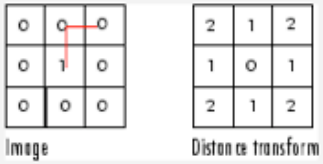
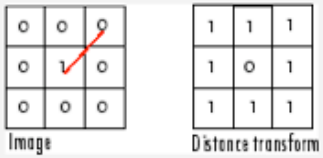
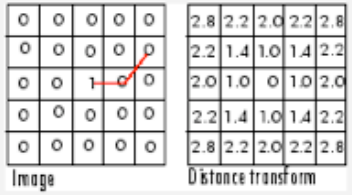


labeling connected objects

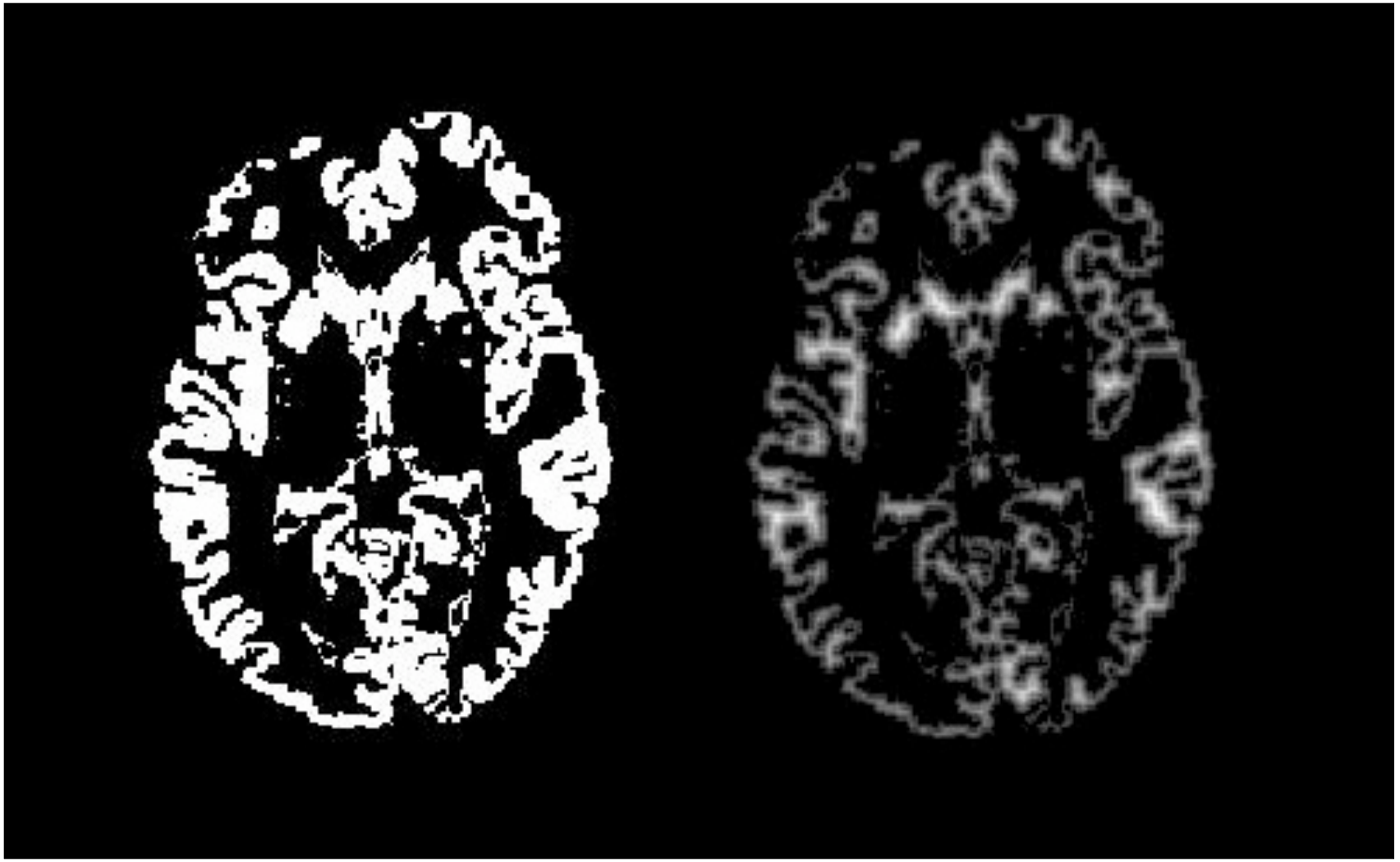


labeling connected objects

distance transforms

Distance Metric	Description	Illustration
Euclidean	The Euclidean distance is the straight-line distance between two pixels.	
City Block	The city block distance metric measures the path between the pixels based on a 4-connected neighborhood. Pixels whose edges touch are 1 unit apart; pixels diagonally touching are 2 units apart.	
Chessboard	The chessboard distance metric measures the path between the pixels based on an 8-connected neighborhood. Pixels whose edges or corners touch are 1 unit apart.	
Quasi-Euclidean	The quasi-Euclidean metric measures the total Euclidean distance along a set of horizontal, vertical, and diagonal line segments.	

distance transforms



distance transforms

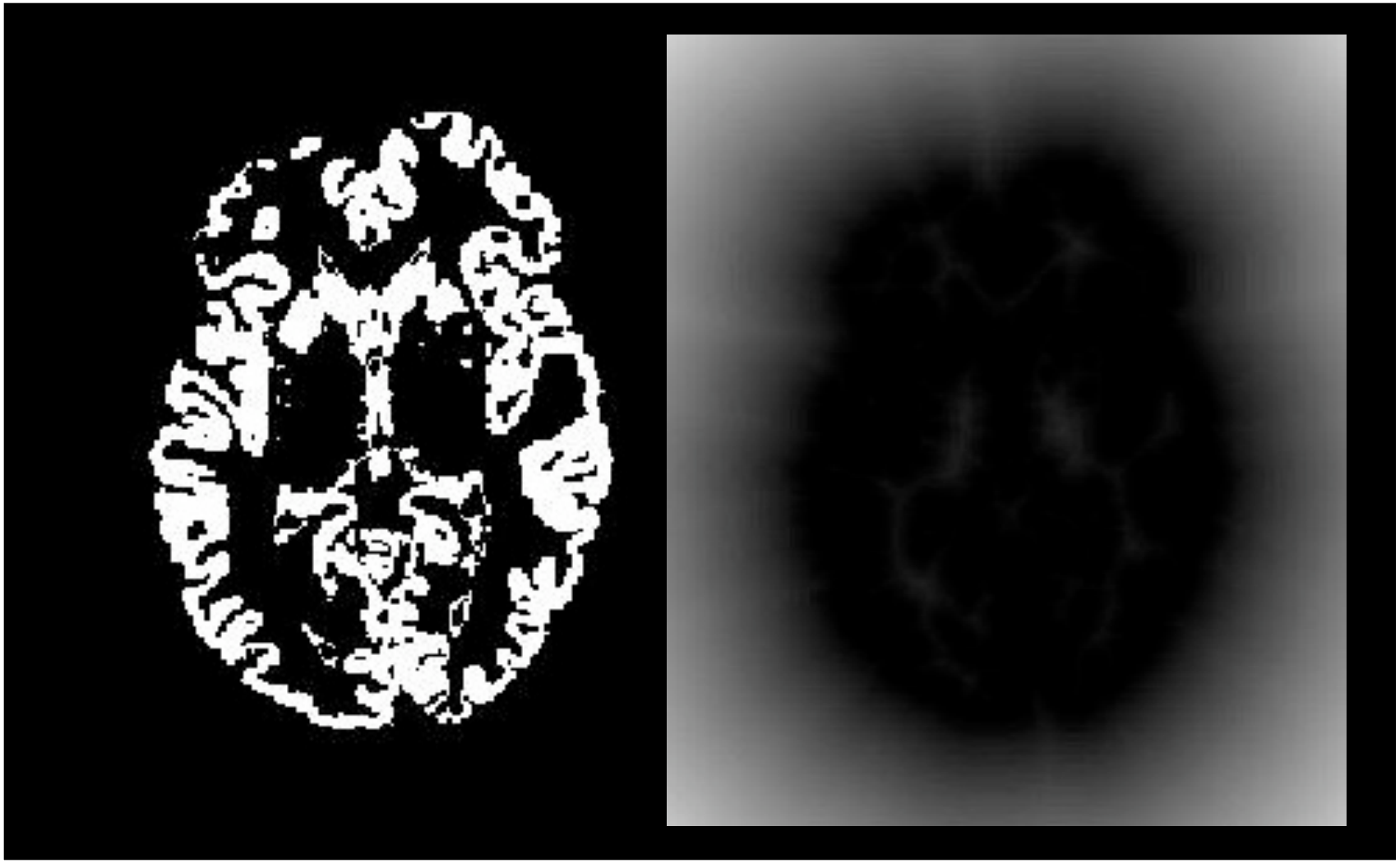
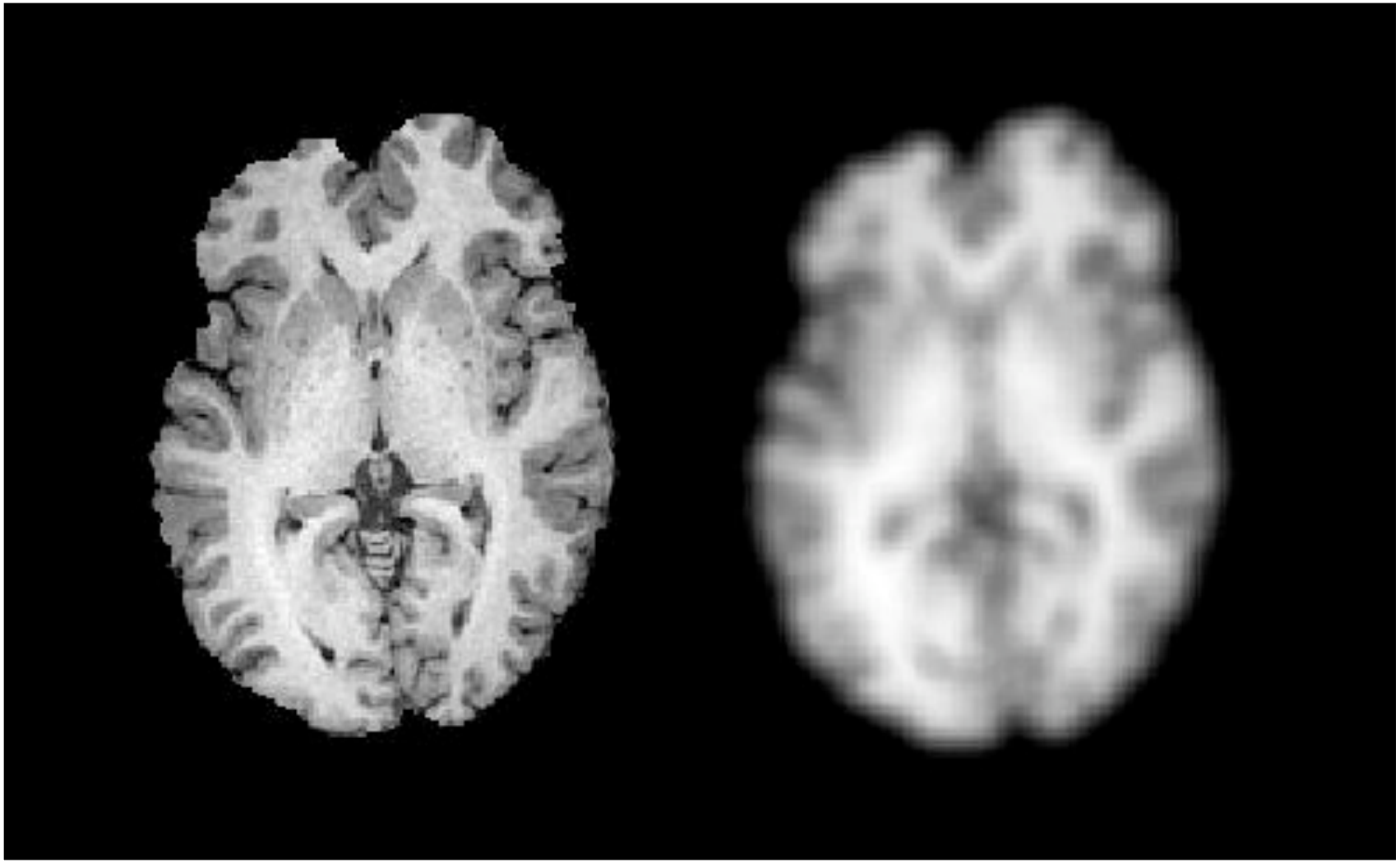
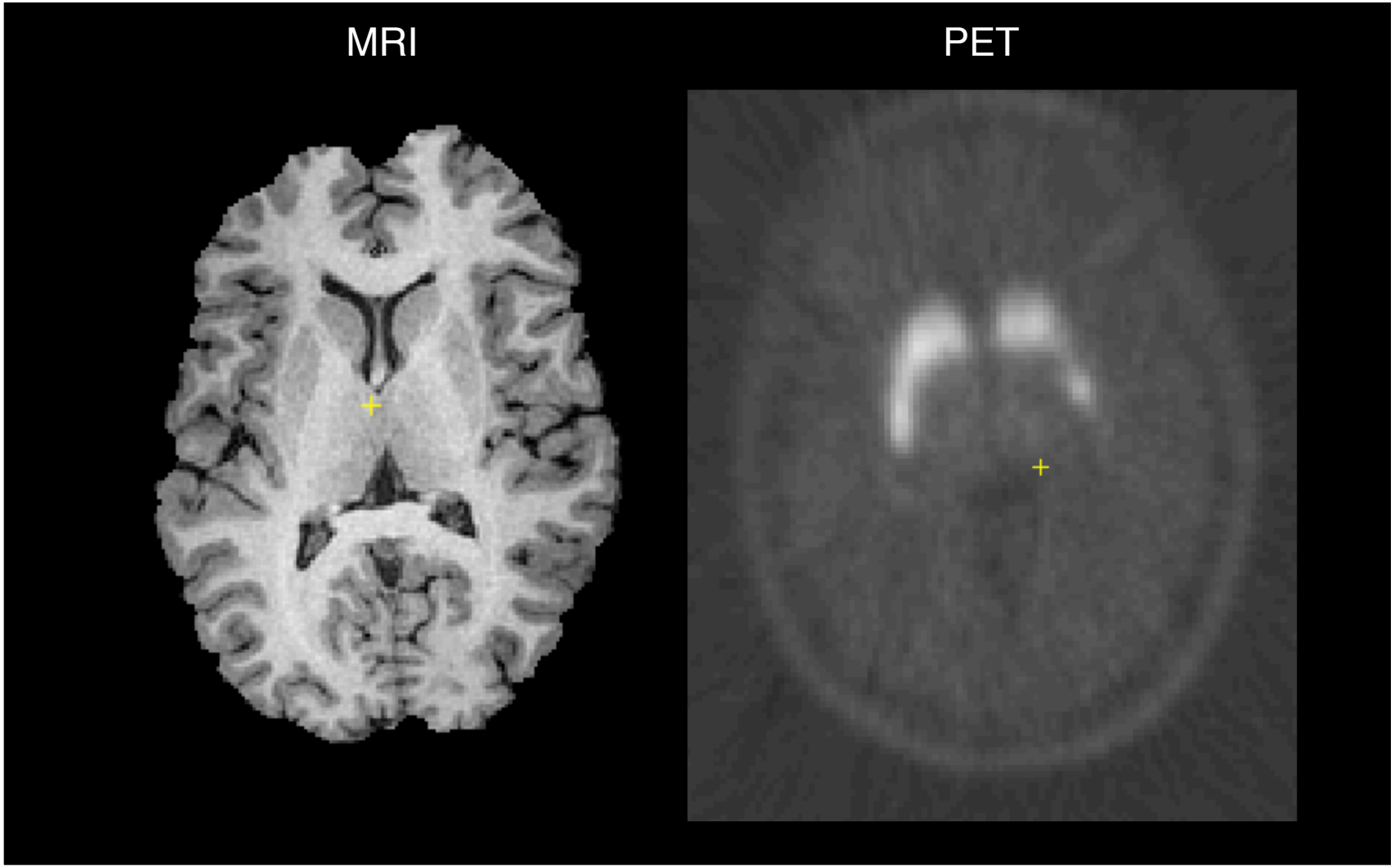


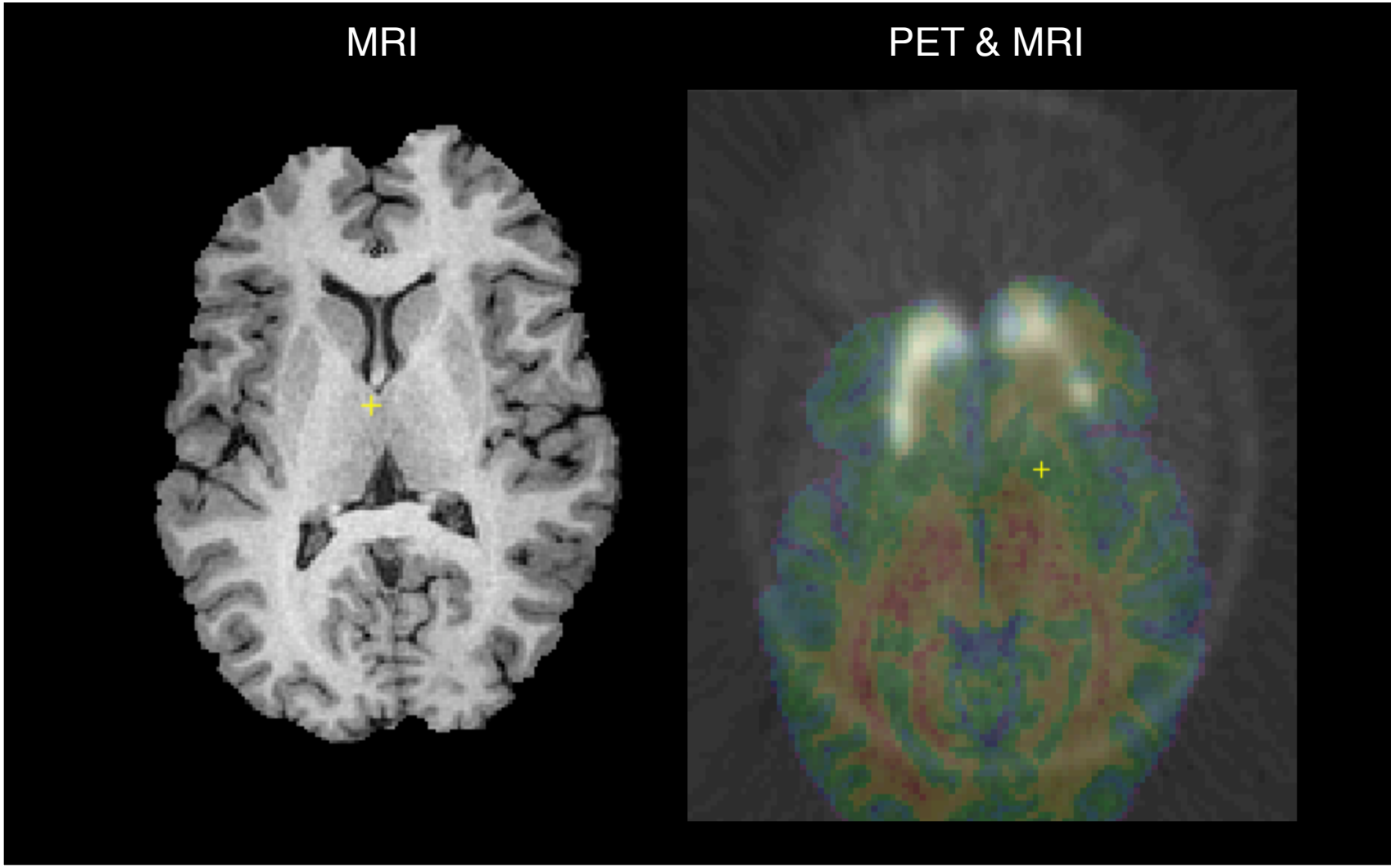
image filters



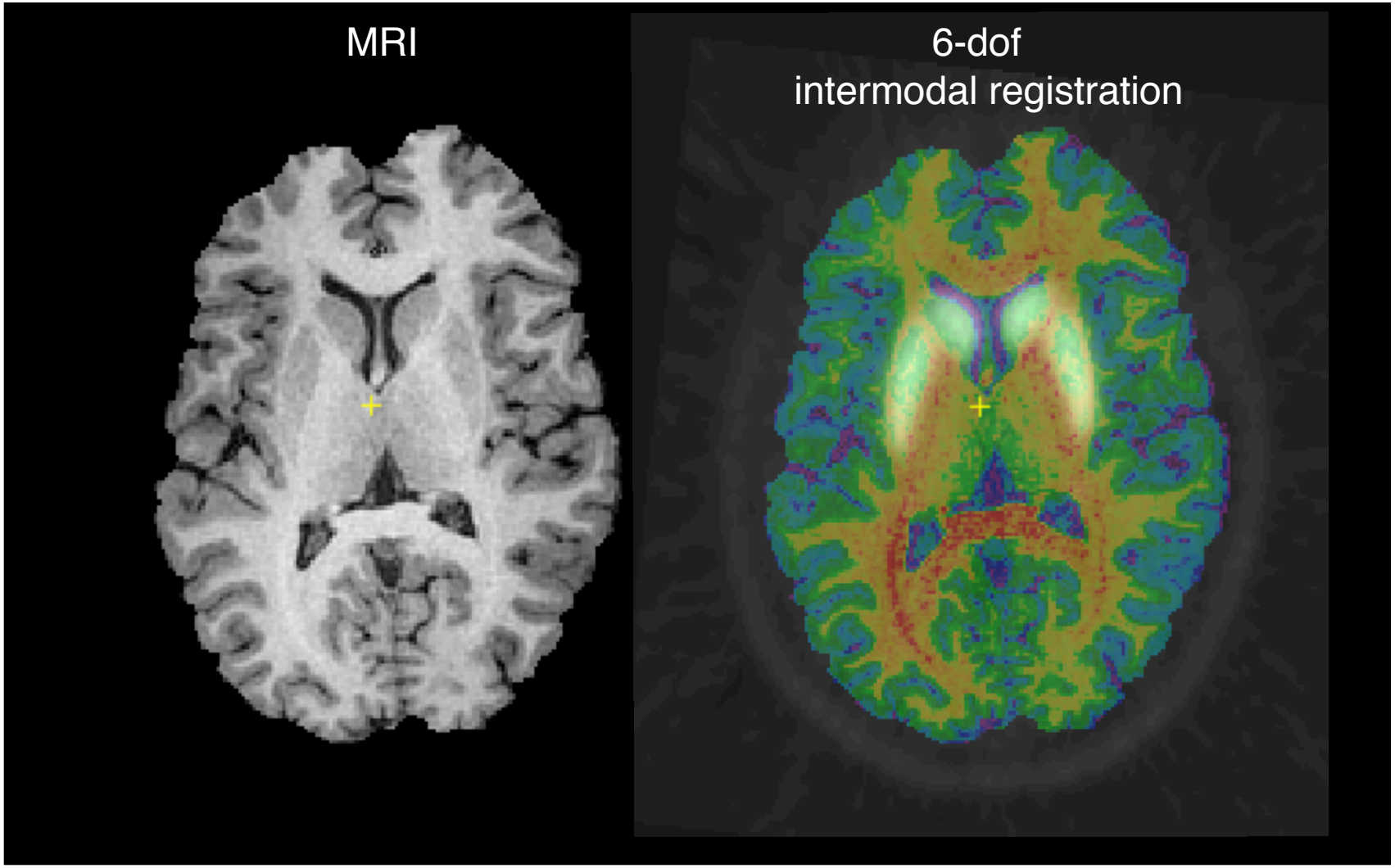
registration



registration



registration



registration

Applications:

intramodality (T1-to-T1, PET-to-PET,...)

- motion correction
- group analysis
- comparative morphometry
- labeling

intermodality (T2-to-T1, PET-to-T1,...)

- group analysis
- multispectral analysis
- labeling

registration

Registration involves:

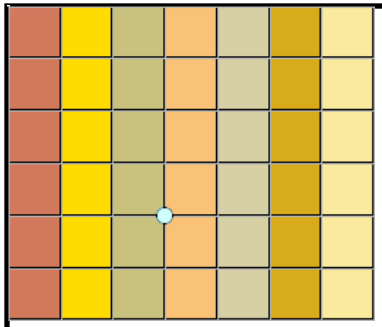
1. Similarity metric: (n)CR, SSD, MSD, (n)CC, MI,...
2. **Transformation** model: affine, piecewise linear, nonlinear,...
3. Regularization method: multi-resolution/scale, Gaussian blur,...
4. Optimization strategy: simplex, gradient descent,...
5. Interpolation type: nearest-neighbor, trilinear, cubic, sinc,...

transformation matrices

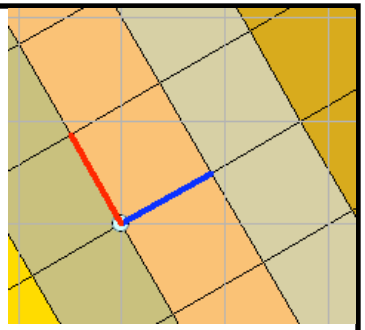
Scale

For scaling (that is, enlarging or shrinking), we have $x' = s_x \cdot x$ and $y' = s_y \cdot y$. The matrix form is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



transformation matrices



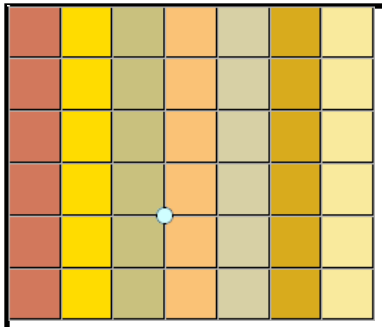
Rotation

For rotation by an angle θ counterclockwise about the origin, the functional form is x' = xcosθ – ysinθ and y' = xsinθ + ycosθ. Written in matrix form, this becomes:

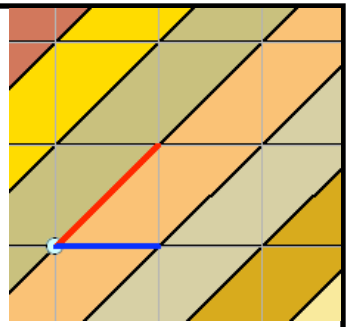
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Similarly, for a rotation clockwise about the origin, the functional form is x' = xcosθ + ysinθ and y' = – xsinθ + ycosθ and the matrix form is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



transformation matrices



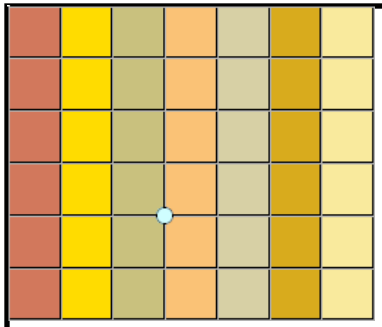
Shear

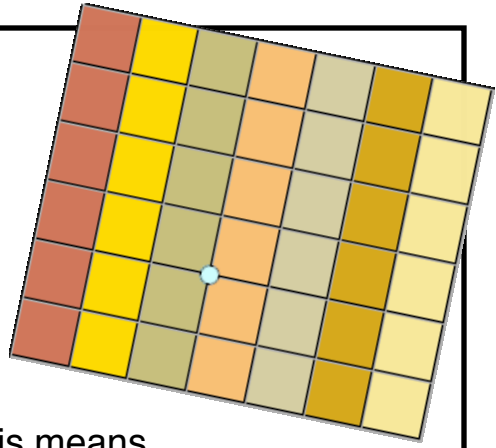
For shear mapping (visually similar to slanting), there are two possibilities.
For a shear parallel to the x axis has x' = x + ky and y' = y;
the shear matrix, applied to column vectors, is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

A shear parallel to the y axis has x' = x and y' = y + kx,
which has matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$





transformation matrices

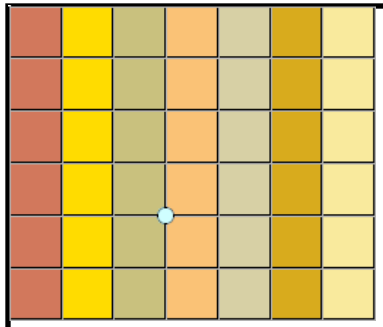
Affine: Translation

To represent affine transformations (linear + translation) with matrices, we must use homogeneous coordinates. This means representing a 2-vector (x, y) as a 3-vector (x, y, 1). The functional form x' = x + tx; y' = y + ty of translation is:

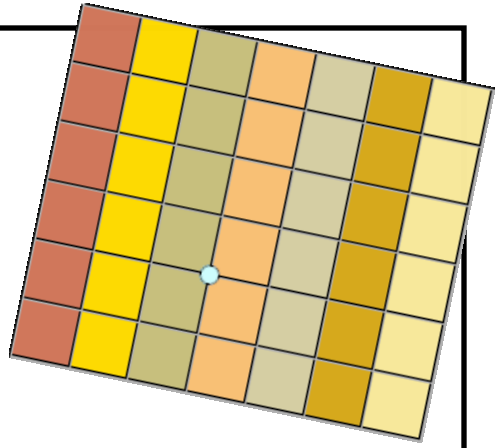
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Affine: Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$



transformation matrices



degrees of freedom

6 dof: 3 translations, 3 rotations (x,y,z) “rigid-body”

7 dof: 3 translations, 3 rotations, 1 global scale

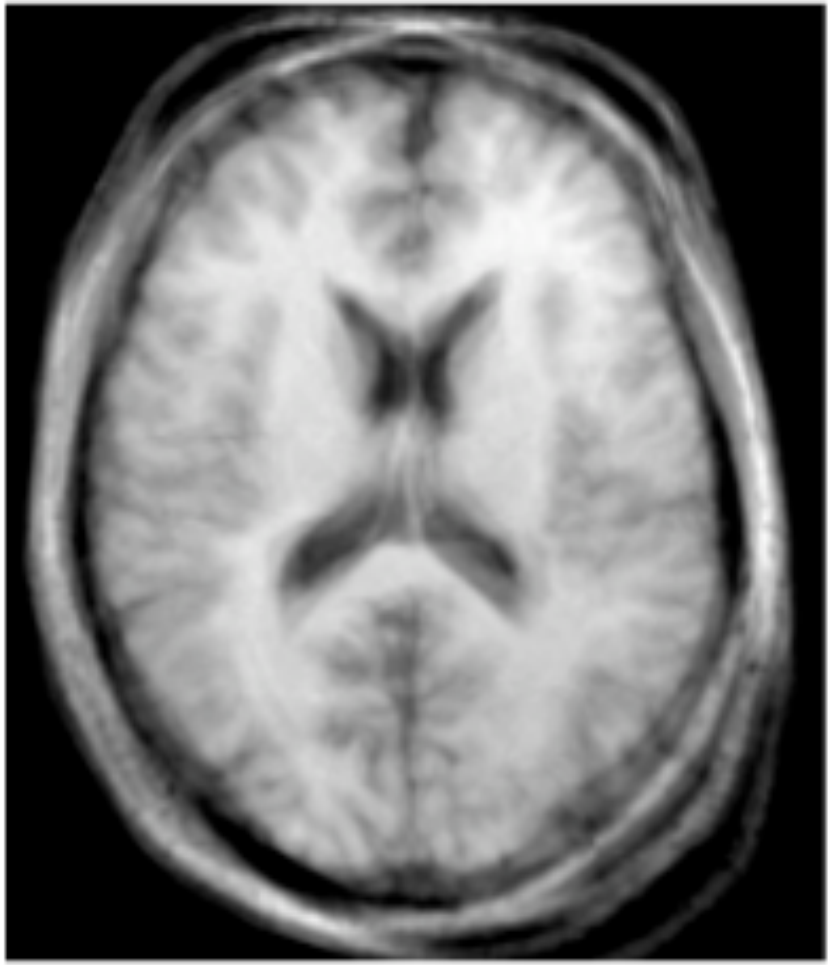
9 dof: 3 translations, 3 rotations, 3 scales

12 dof: 3 translations, 3 rotations, 3 scales, 3 shears

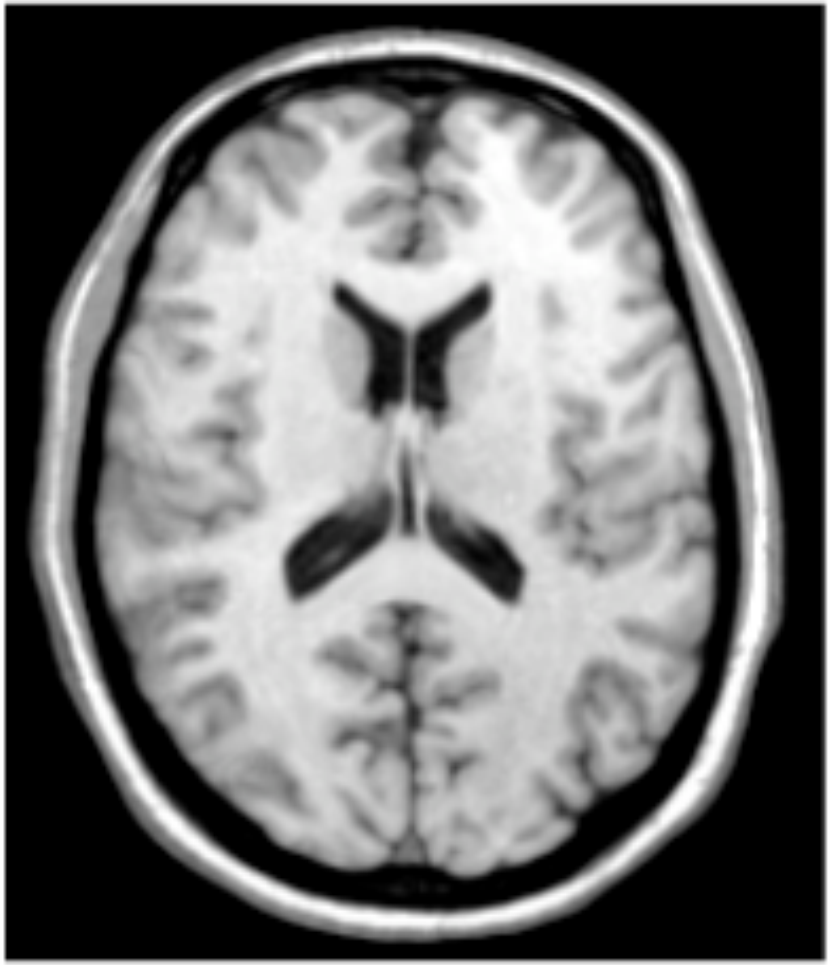
>12: nonlinear transformation

linear vs. nonlinear

different subjects, same image type

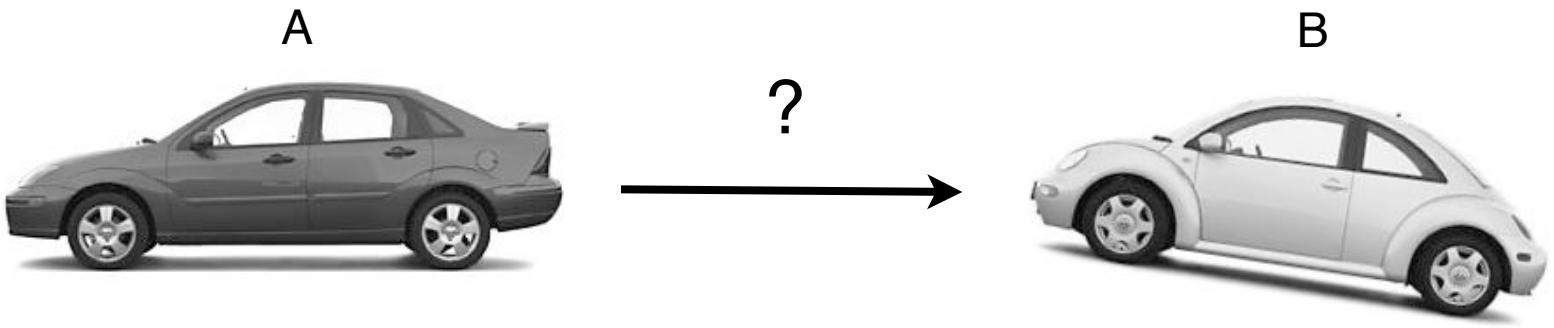


linear

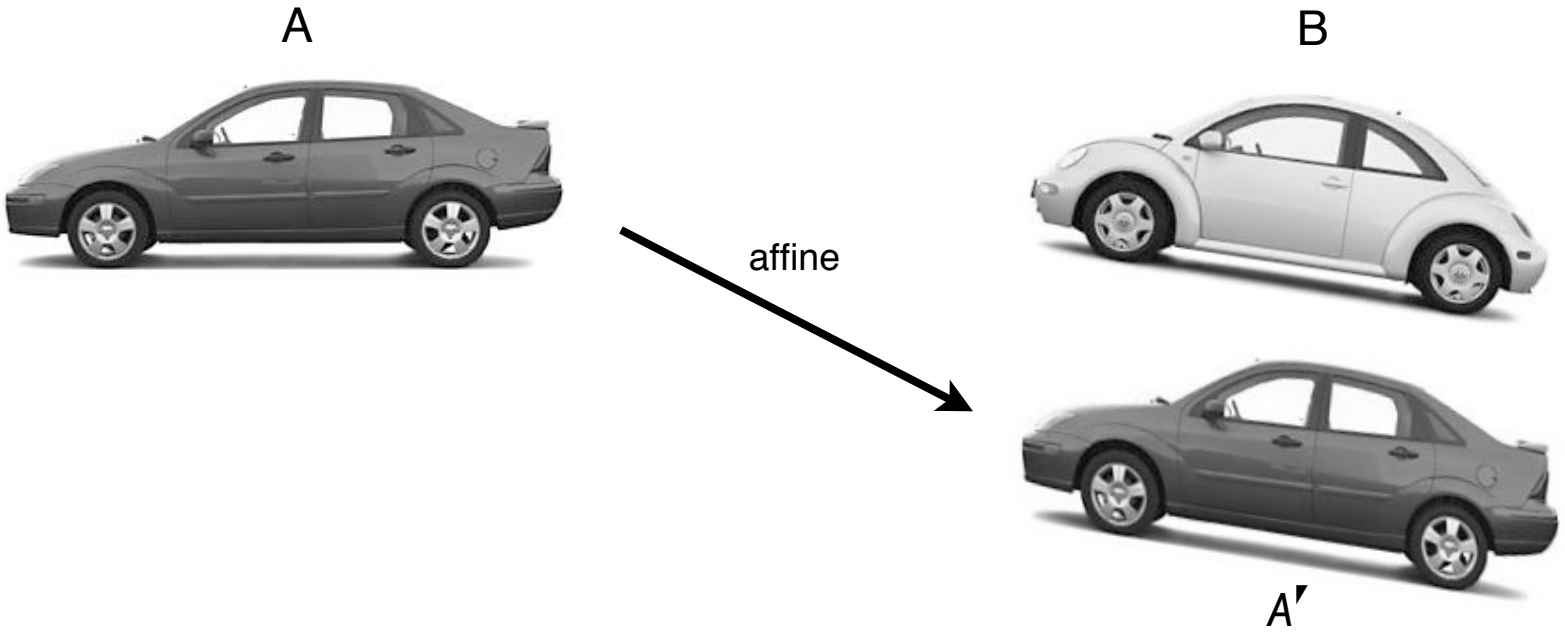


nonlinear

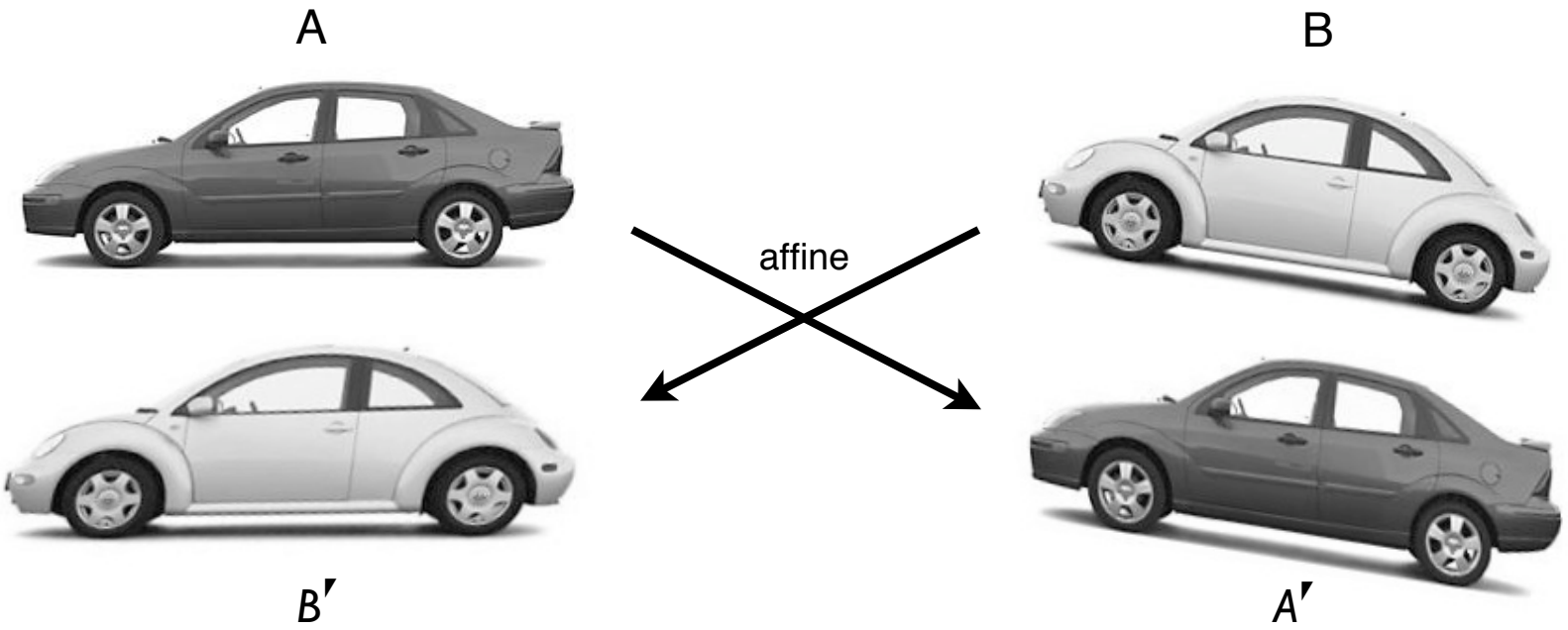
affine transformation



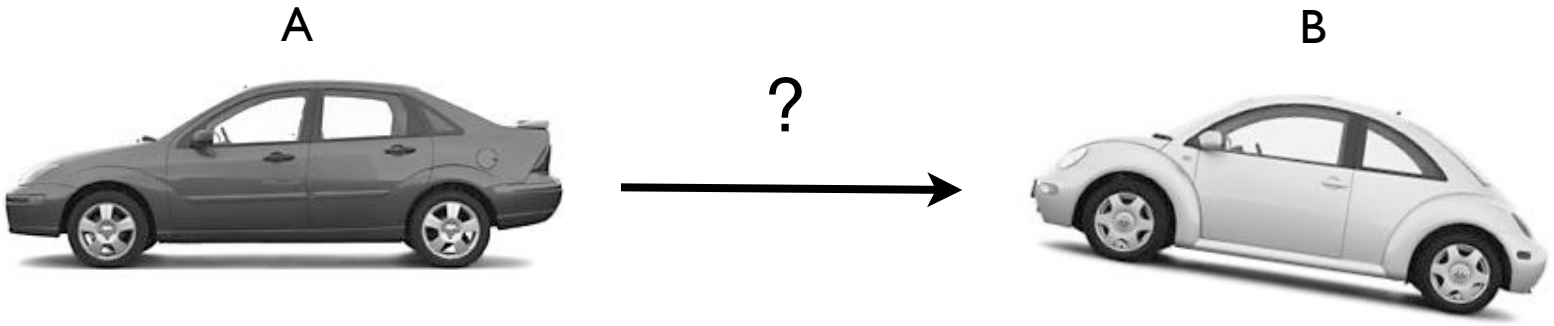
affine transformation



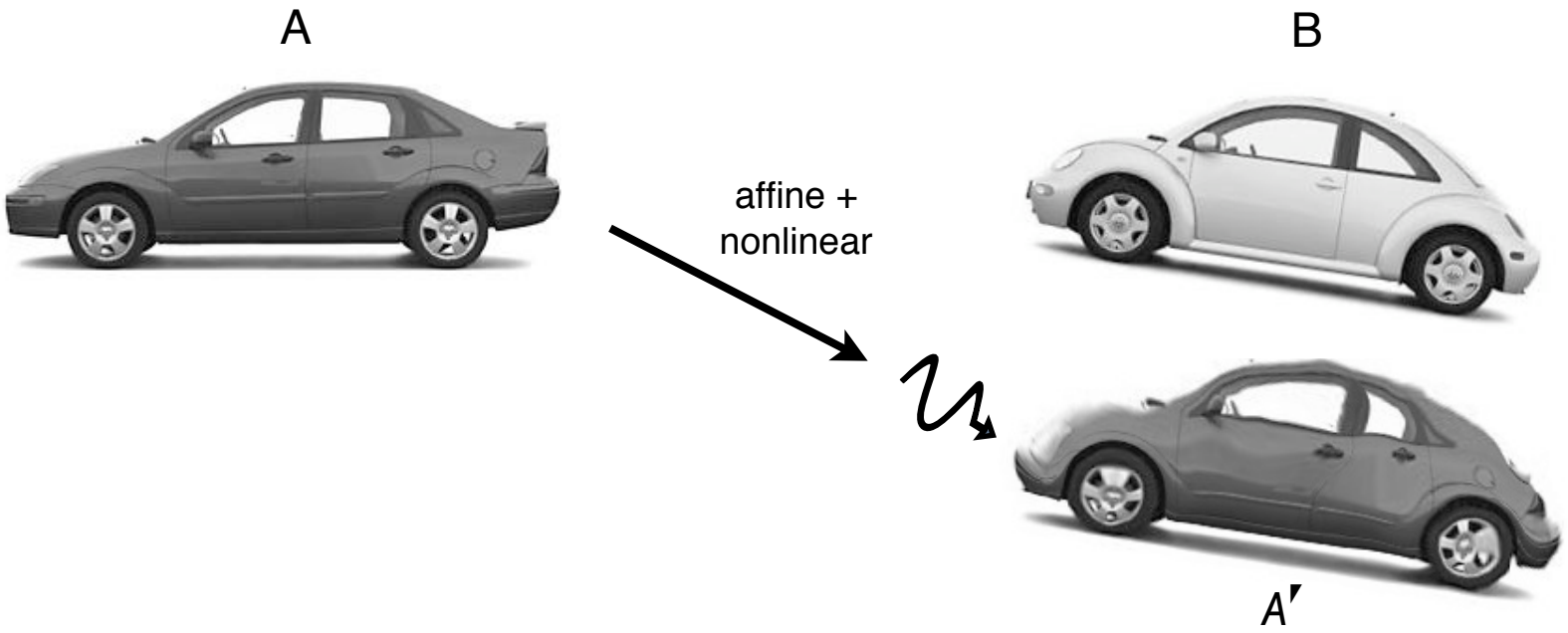
affine transformation



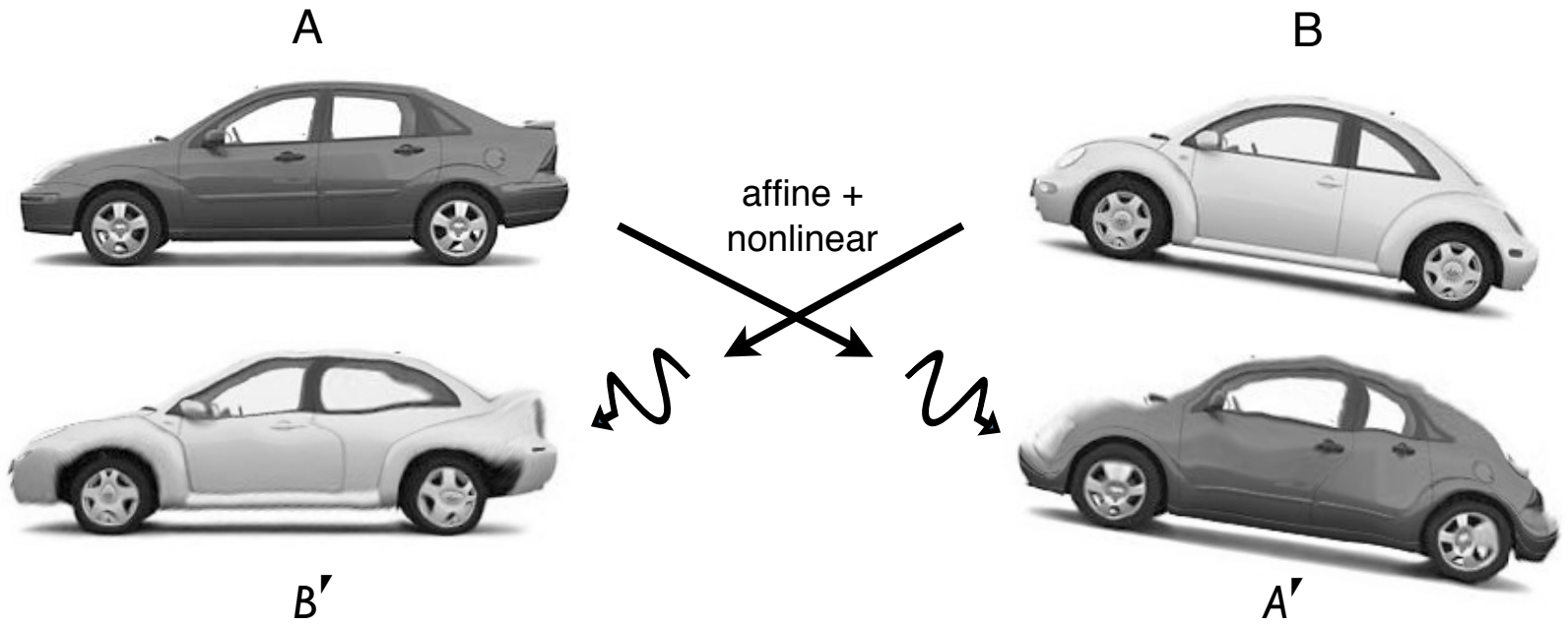
nonlinear transformation



nonlinear transformation

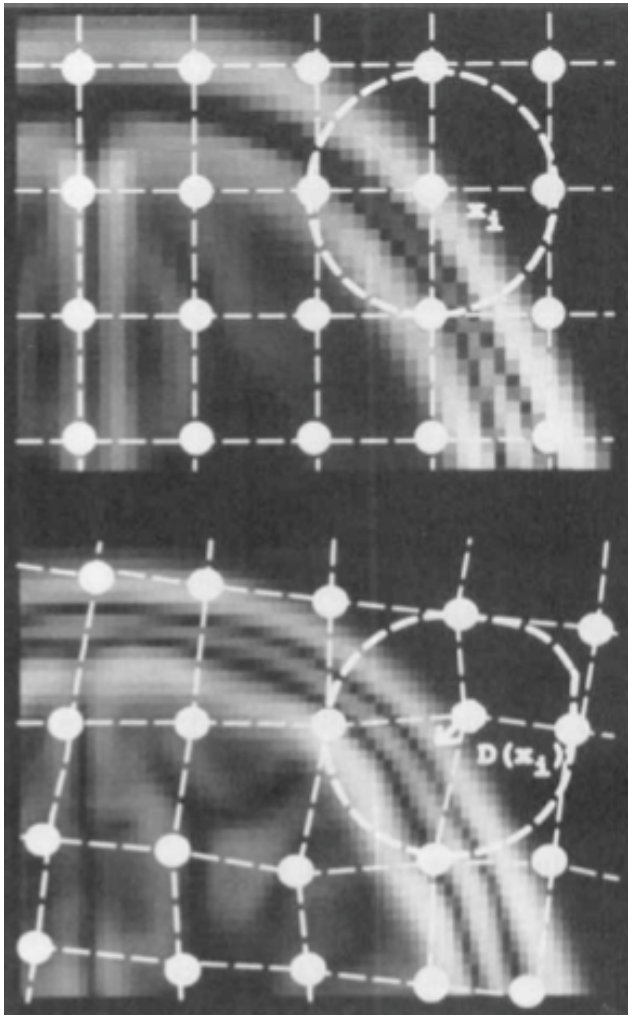


nonlinear transformation

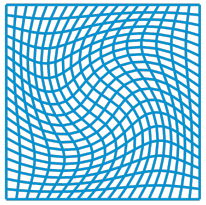


nonlinear brain image registration

example: local translations (ANIMAL), FFD (ART, IRTK, FNIRT, etc.)



ANIMAL:
optimization of three translational parameters to maximize the neighborhood correlation for each node



nonlinear brain image registration

example of bidirectional diffeomorphism: symmetric normalization (SyN)

transformation model

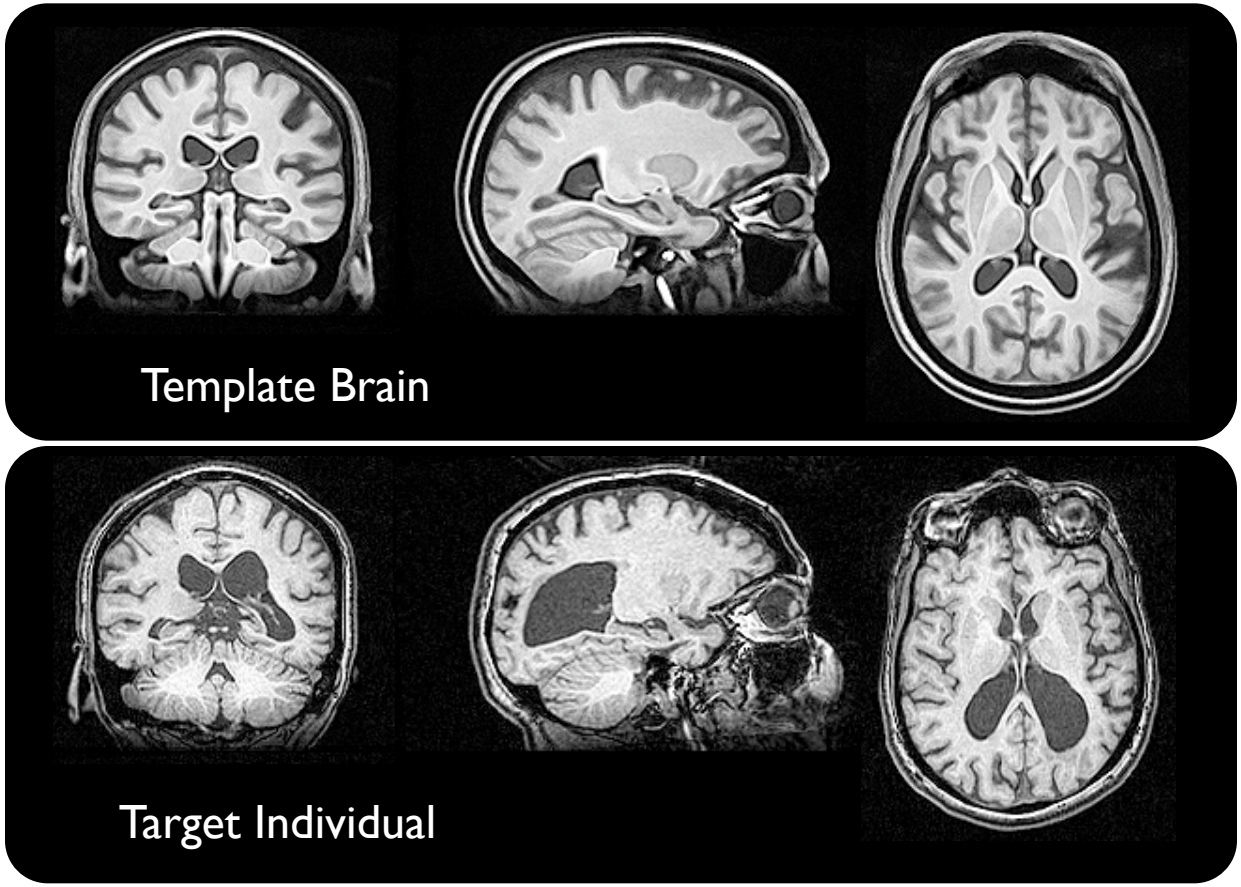
similarity metrics

$$\int_0^1 < Lv(x,t), v(x,t) > dt + w_1 SSD(I,J) + w_2 MI(I,J) + w_3 \sum_i LM_i(I,J)$$

diffeomorphic regularization + intensity difference + mutual information + landmark guidance

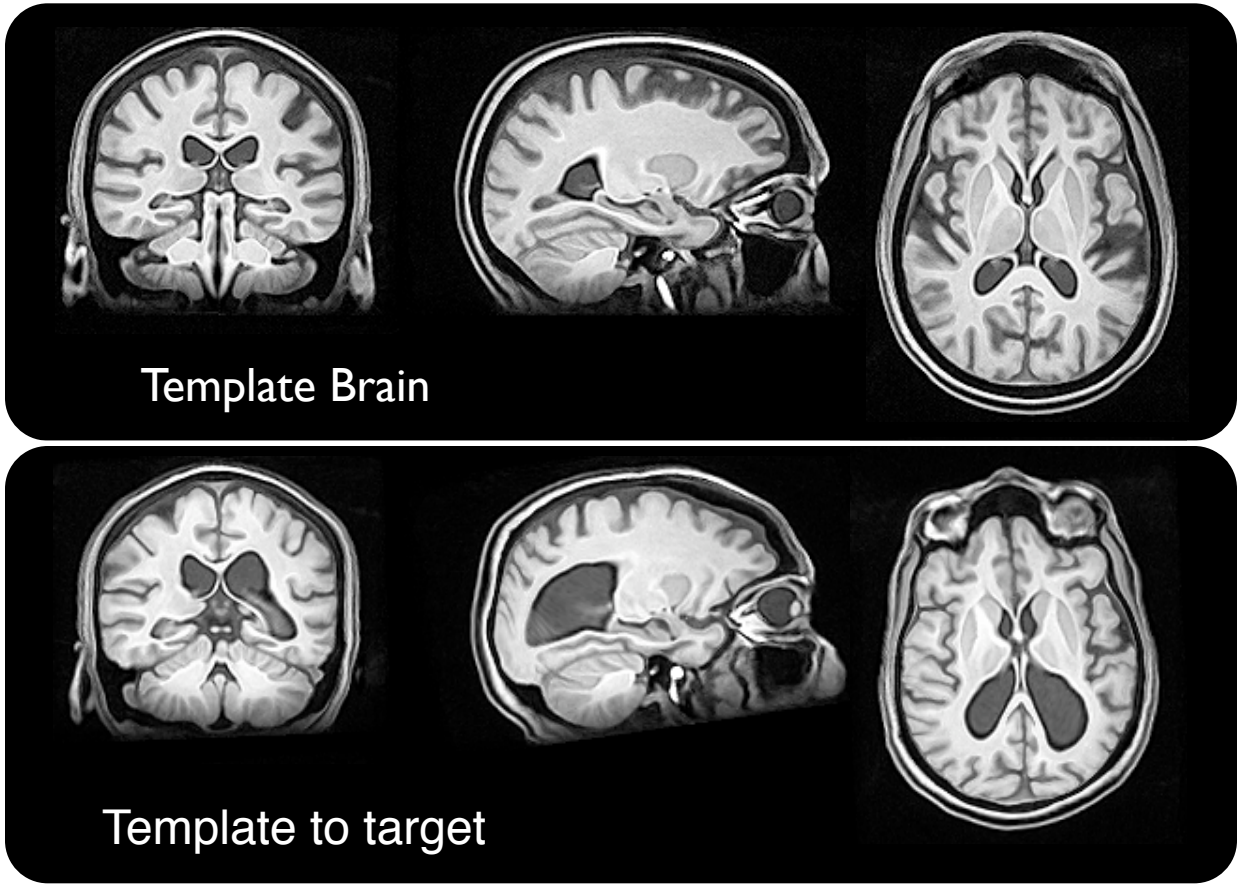
(*L* is the linear operator regularizing the velocity.)

template registration



Avants, et al. 2008

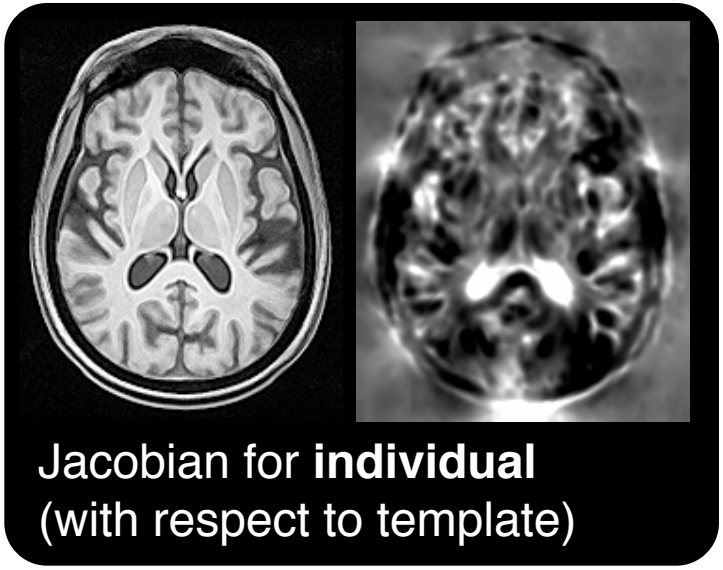
template registration



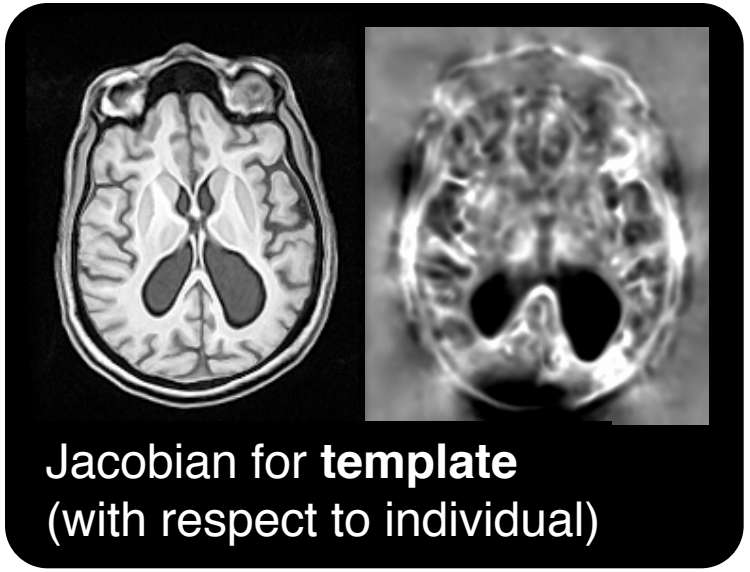
Avants, et al. 2008

precise, but accurate?

template registration
application 1: deformation-based morphometry



Jacobian for **individual**
(with respect to template)

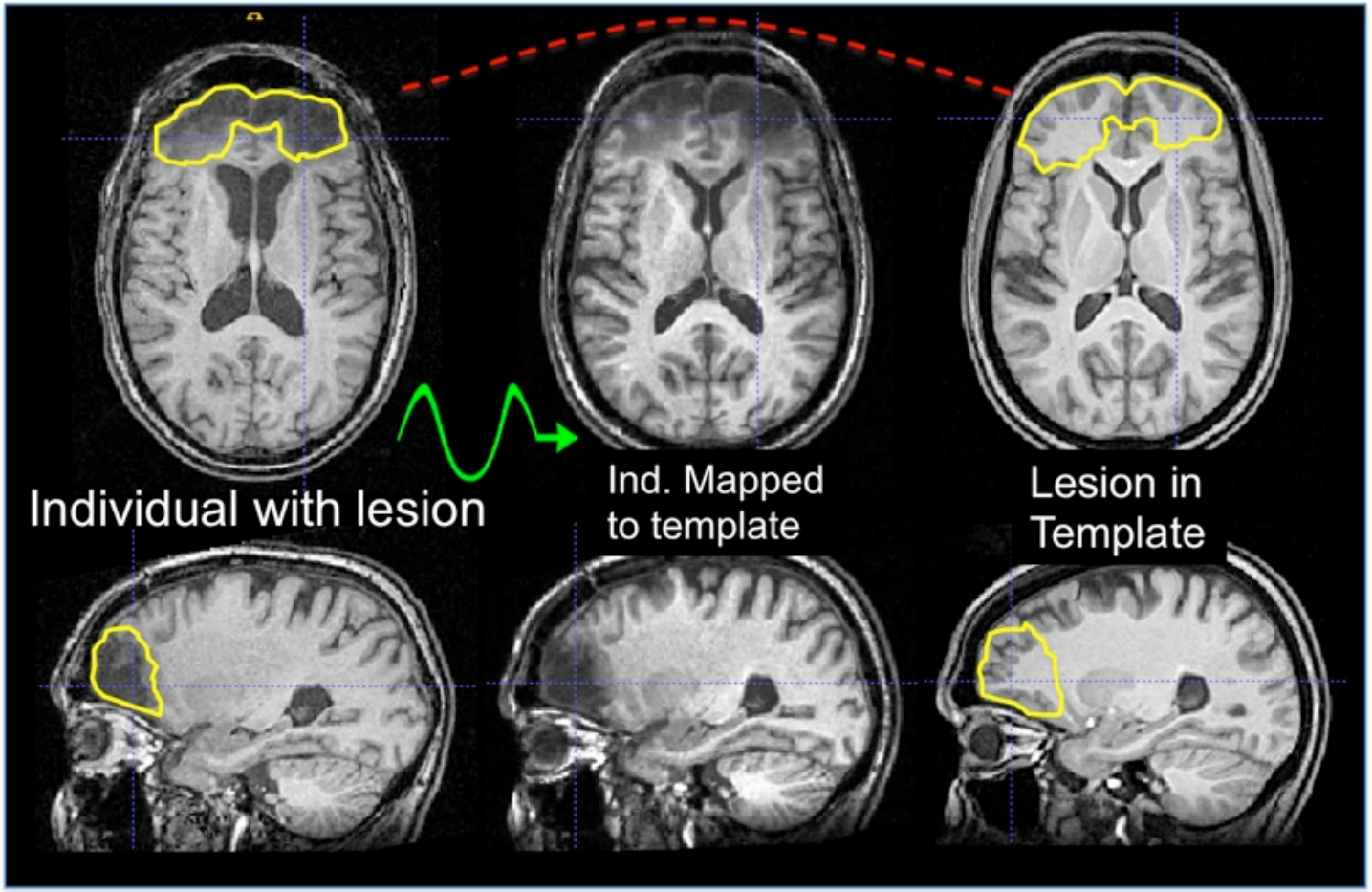


Jacobian for **template**
(with respect to individual)

Avants, et al. 2008

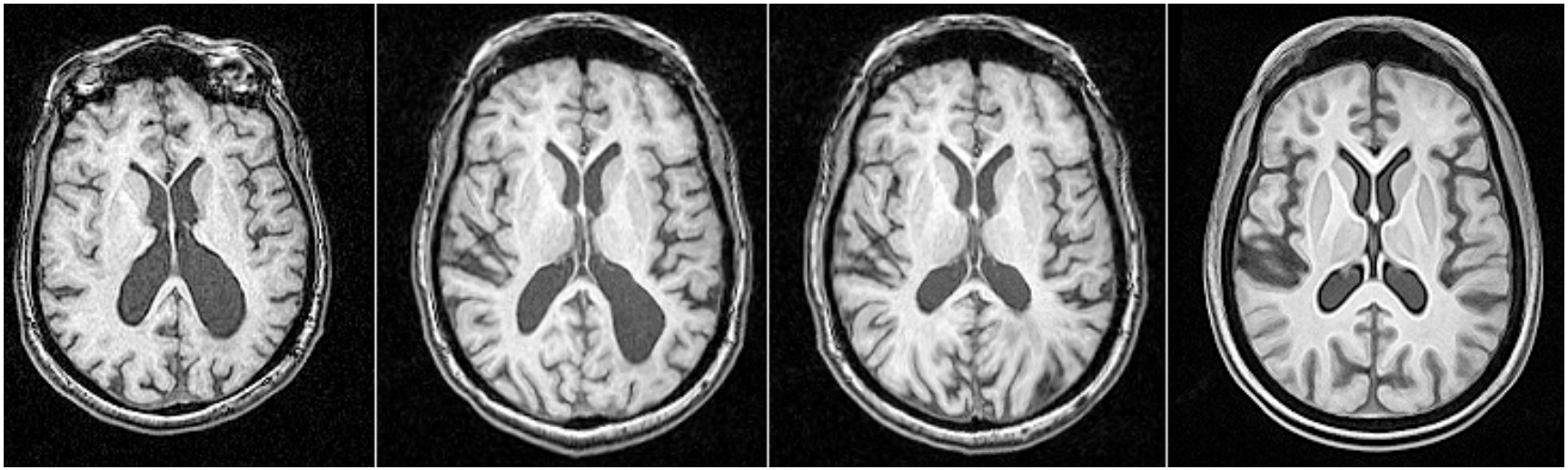
template registration

application 2: anatomical localization



Avants, et al. 2008

template registration
application 3: spatial normalization

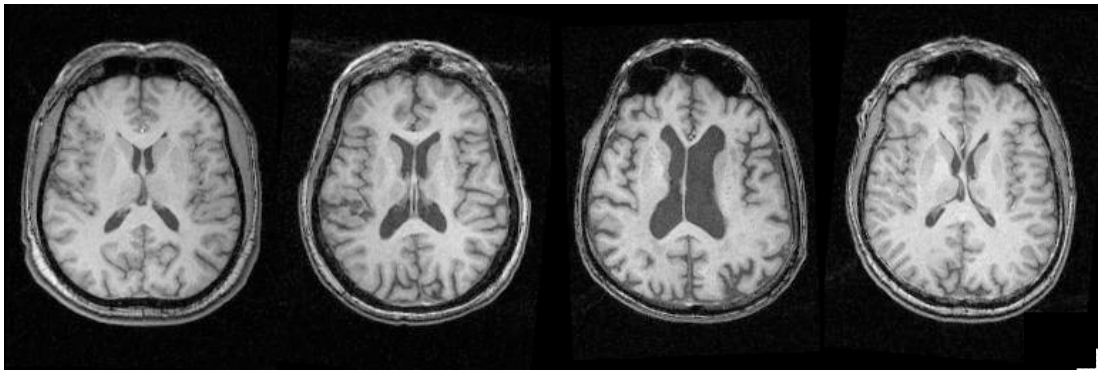


individual | default mapping to template | adjusted mapping to template | template

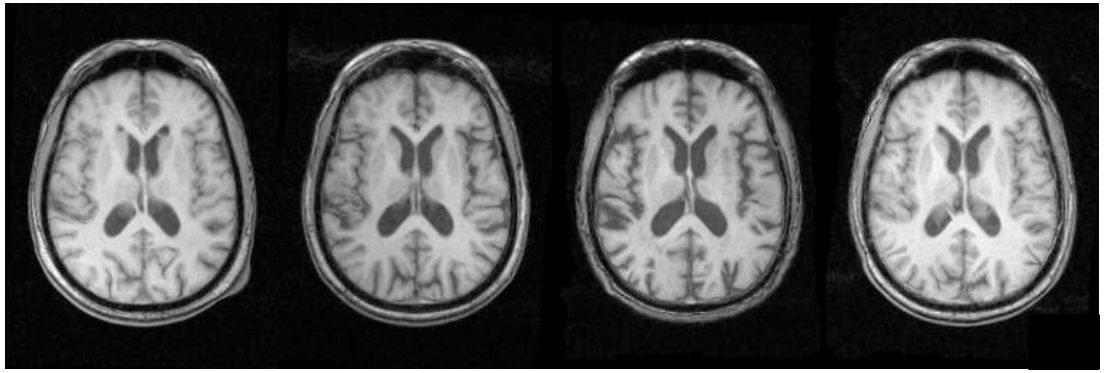
template registration

application 4: group analysis of data

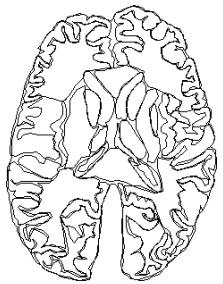
data (structure, function, etc.) in individual space



template image



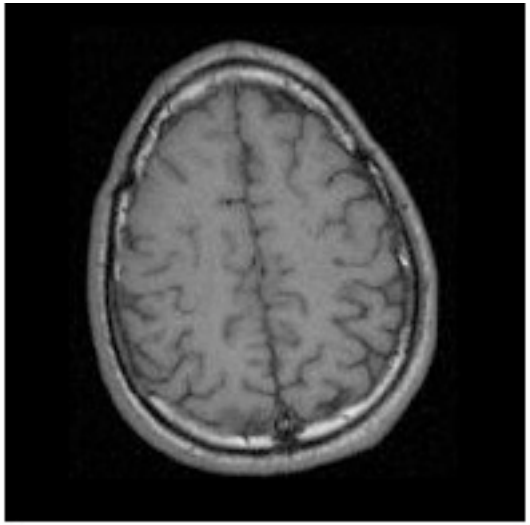
individuals normalized to template space



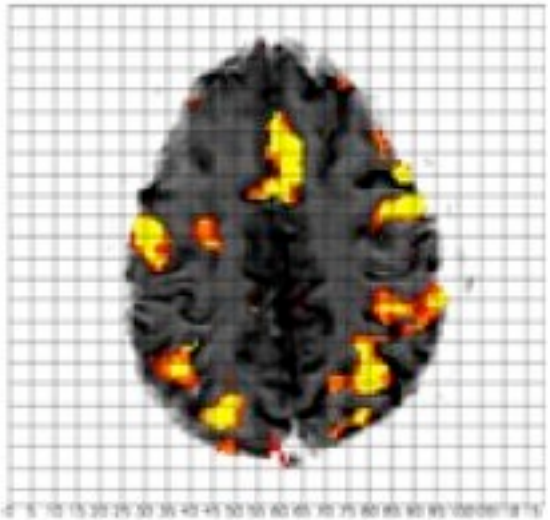
template labels

template registration
application 5: atlas-based anatomical labeling

manual labeling:



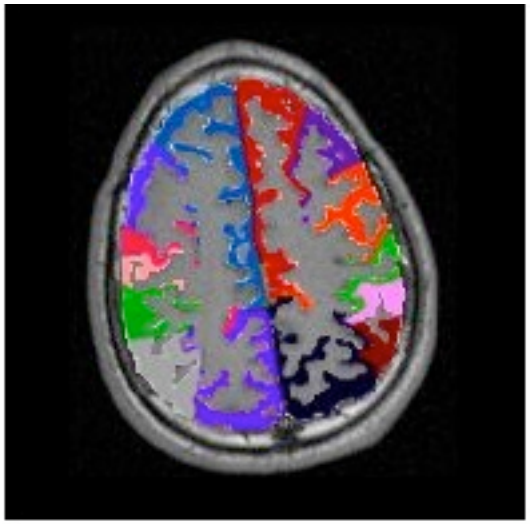
MRI



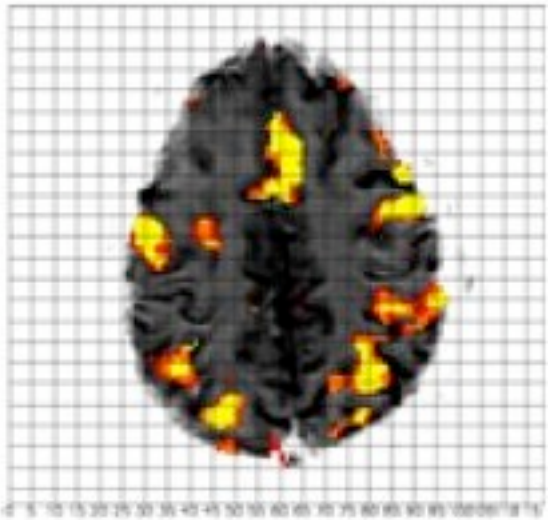
fMRI

template registration
application 5: atlas-based anatomical labeling

manual labeling:



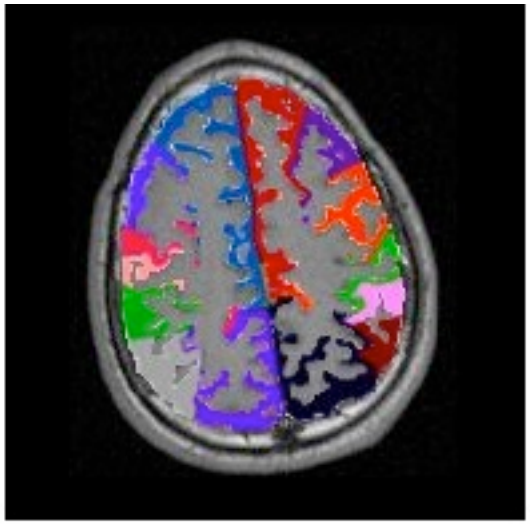
labeled MRI



fMRI

template registration
application 5: atlas-based anatomical labeling

manual labeling:

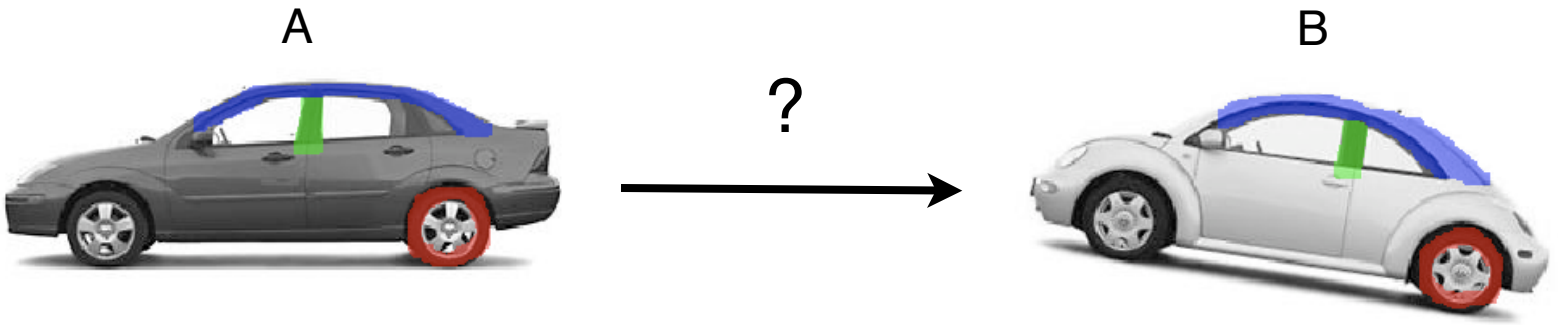


labeled MRI

labeled fMRI

template registration
application 5: atlas-based anatomical labeling

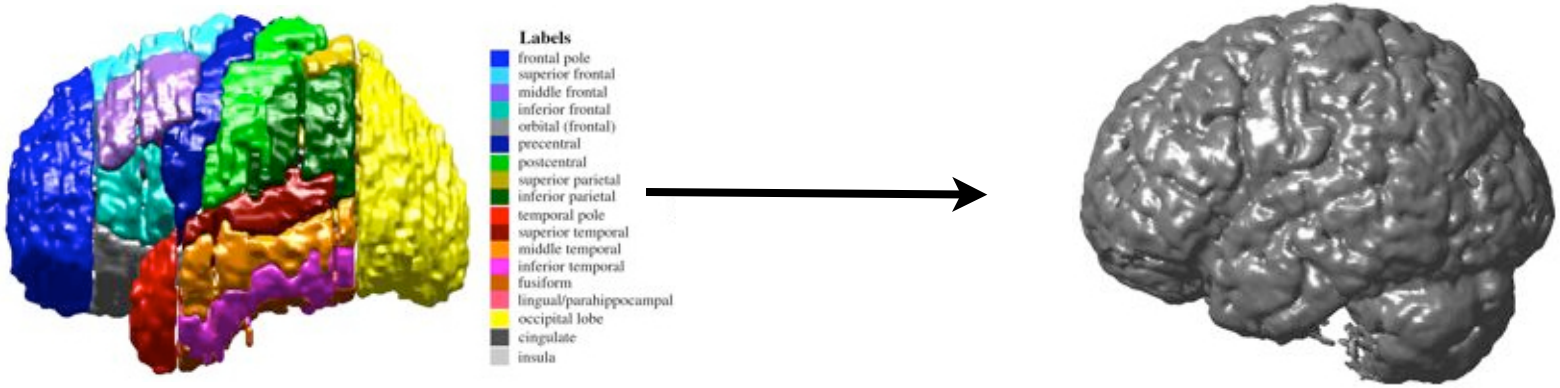
automated labeling:



template registration

application 5: atlas-based anatomical labeling

automated labeling:



References

http://www.mathworks.com (Matlab Image Processing Toolbox documentation)

http://mathworld.wolfram.com/AffineTransformation.html

http://www.wikipedia.org

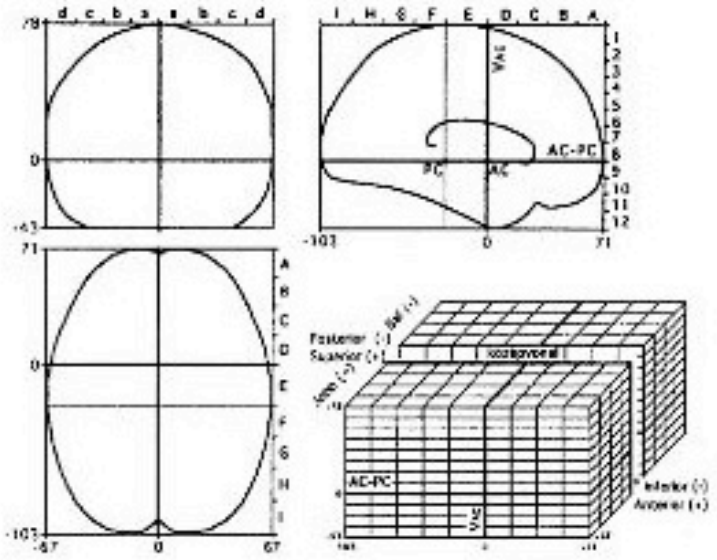
http://www.quantdec.com/GIS/affine.htm

http://www.picsl.upenn.edu/ANTS/

SyN: “ANTS Anatomy: Overview of ANTS Toolkit” keynote presentation
Brian Avants, et al., Penn Image Computing & Science Laboratory
University of Pennsylvania

ANIMAL: http://dx.doi.org/10.1002/hbm.460030304 and chapter 9

individual atlases

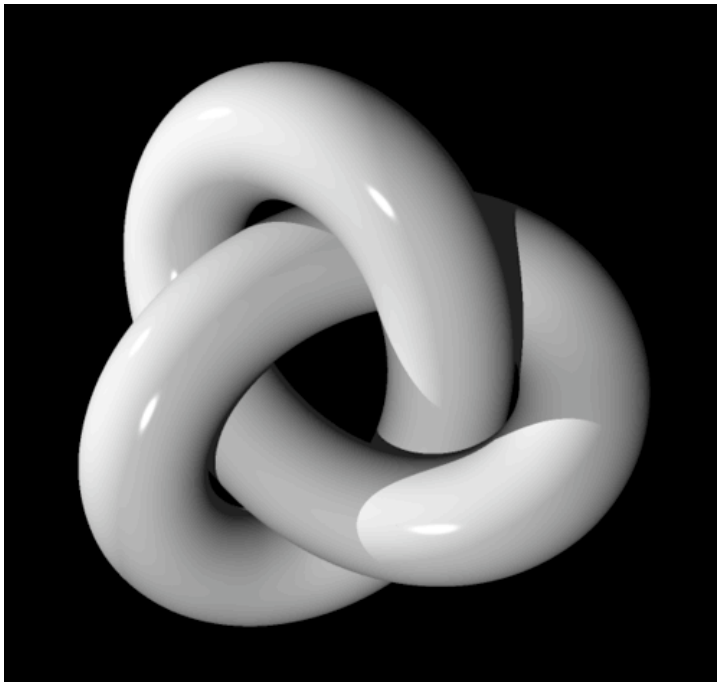


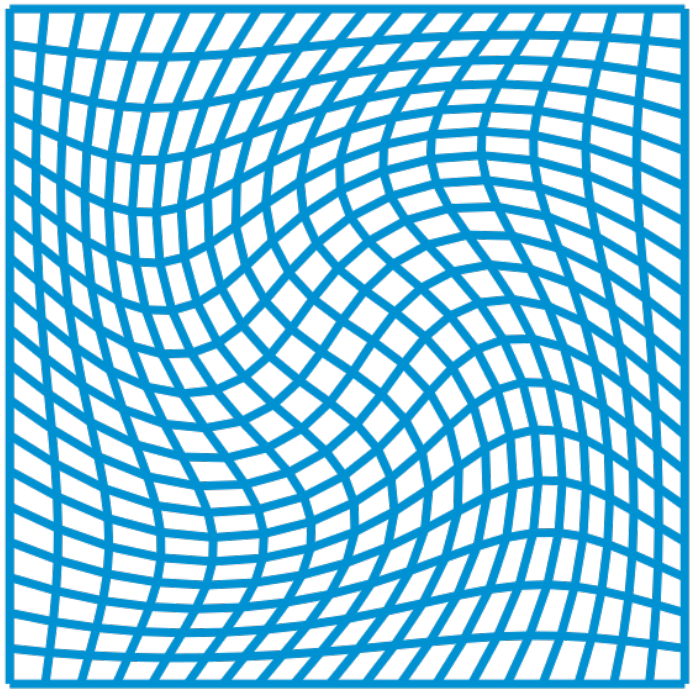
Talairach-Tournoux coordinate frame and variants (1967, 1988,...)

individual atlases

NIH Visible Human Male, Harvard's Whole Brain and SPL Atlases, vanEssen's CARET Atlas (top)

nonlinear brain image registration
example: bidirectional diffeomorphism





homeomorphism: continuous mapping and inverse, but not deformation (trefoil knot, circle)

diffeomorphism: bijective map from manifold M to N and its inverse are differentiable

Jacobian matrix: representation of a differential, an $n \times n$ matrix of first order partial derivatives whose entry in the i-th row and j-th column is $\partial f_i / \partial x_j$. The Jacobian can be thought of as describing the amount of "stretching" that a transformation imposes.