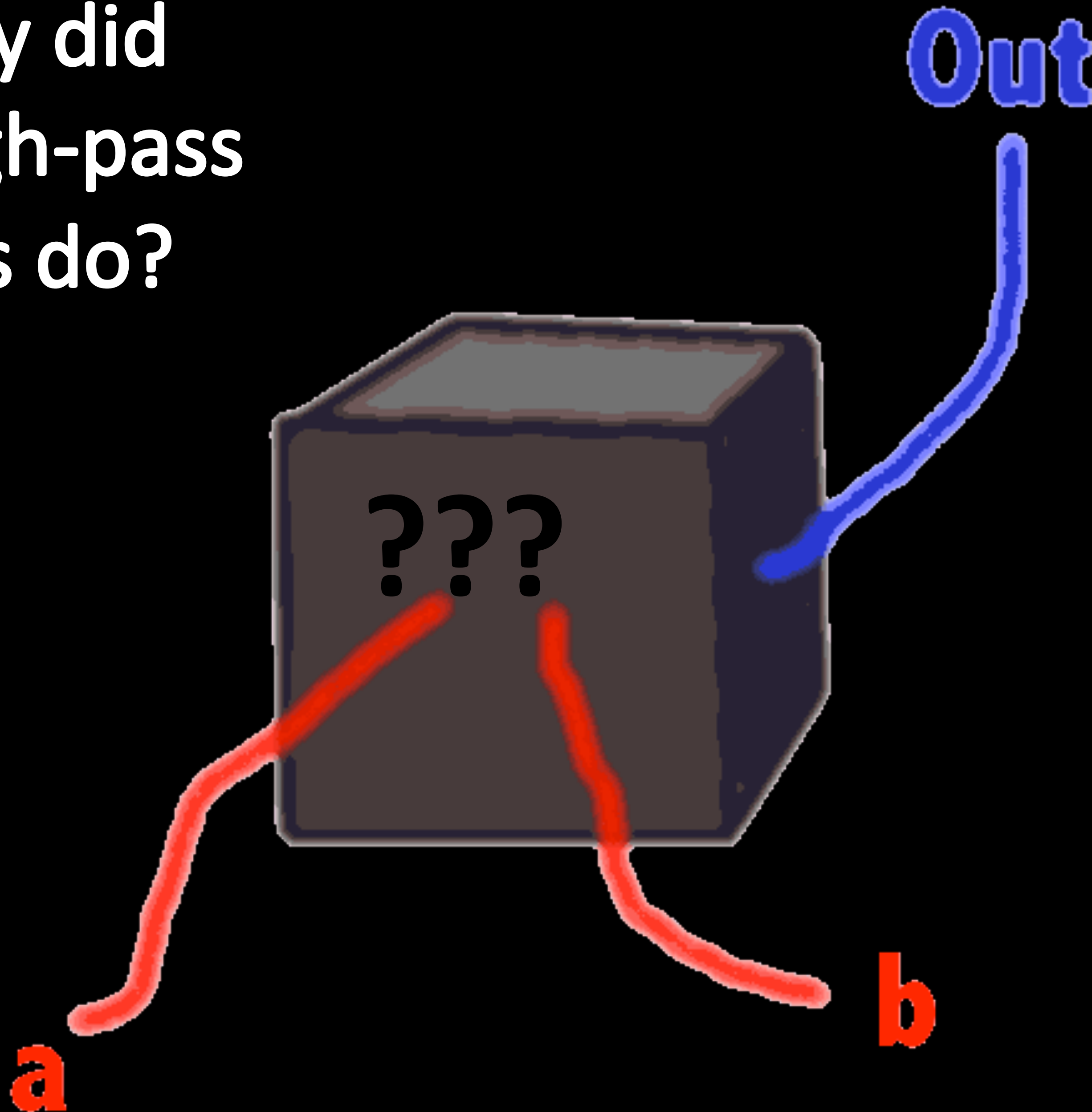
# Preprocessing fMRI Data

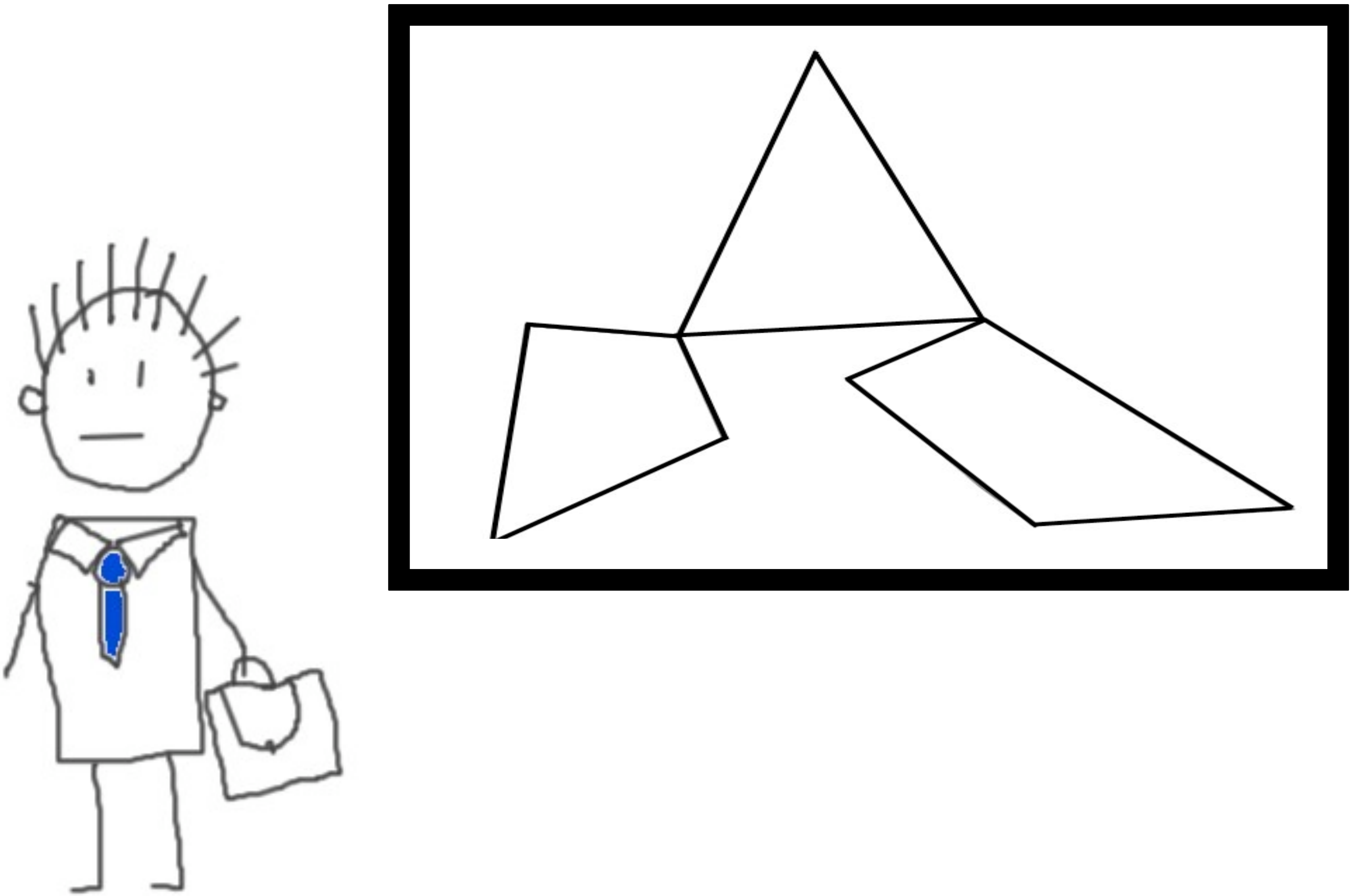
Lei Liew
USC Neuroimaging Workshop
08.11.2011

What exactly did choosing a high-pass filter of 100s do?

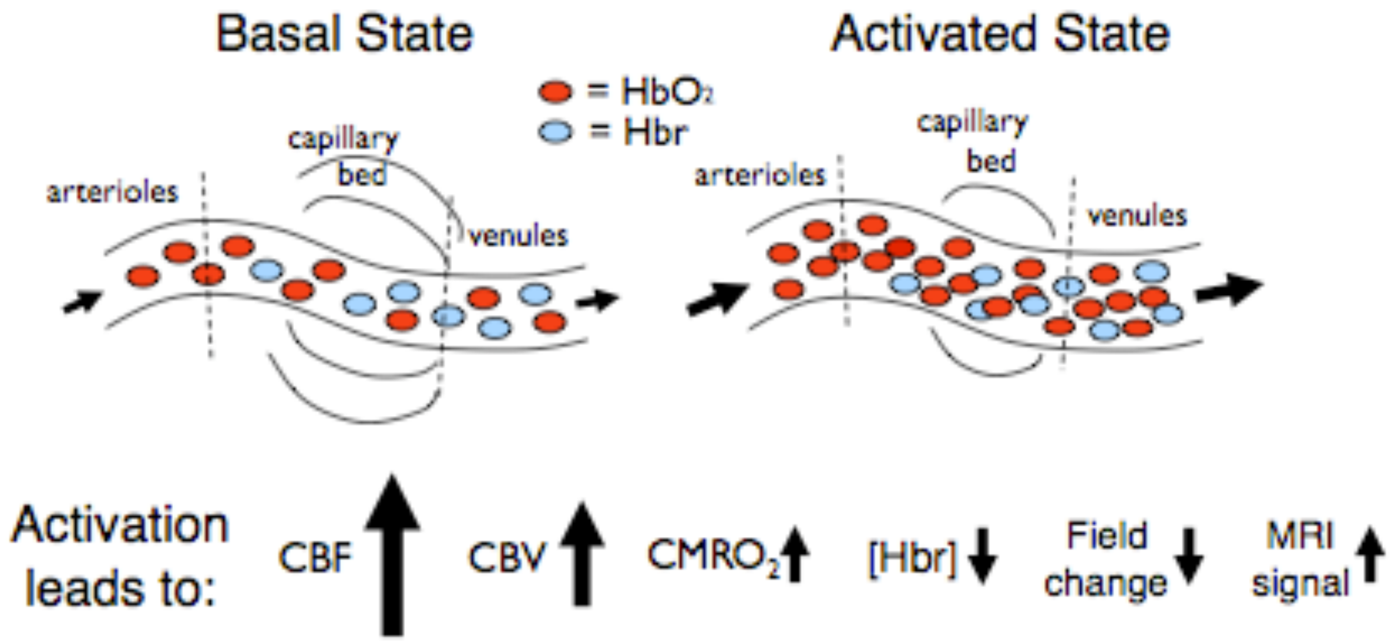


# Typical Process:

# Typical Process:



# Typical Process:



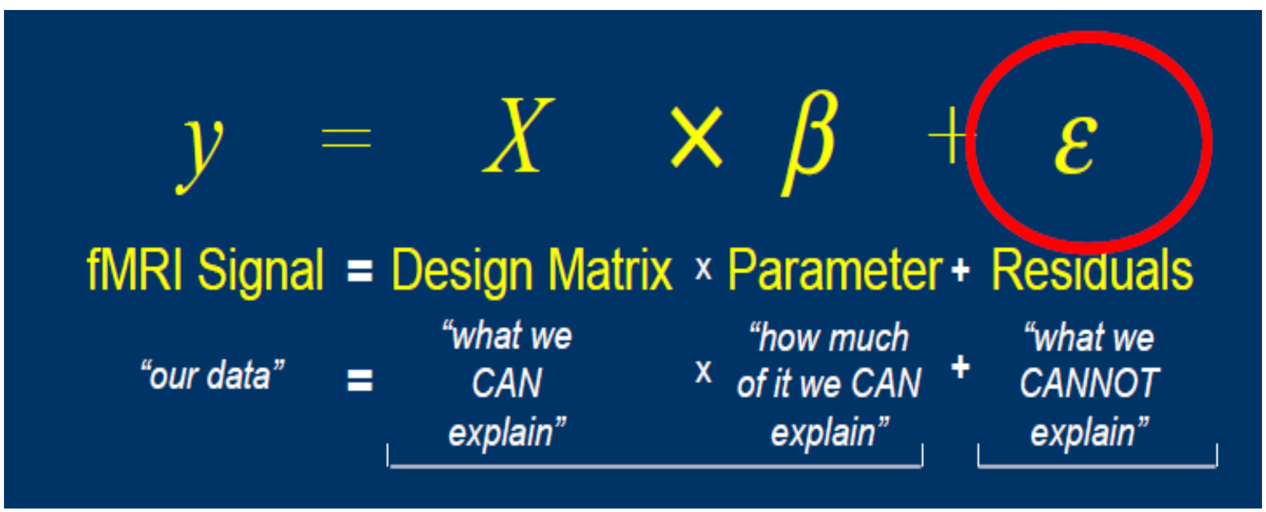

- Field changes (perturbations) --> dephasing --> $T_2$* effect
- BOLD-tuned MRI ($T_2$*-weighted) is sensitive to this effect

# Why Preprocess?

- Data is noisy!
- Subjects move!
- Things change over the time of your experiment (more if it’s long and boring)!
- There a giant bulldozers and excavators 10 feet away from your scanner! Wait, that’s not normal..?
- Etc etc etc…

# Why Preprocess?

- Preprocessing attempts to increase Signal (e.g., BOLD contrast) to Noise (variance from movement, subject, scanner artifacts, other uncontrollables) ratio
- Helps you meet assumptions so you can do stats on your data



# Preprocessing Resources

- Full credit for all the info and most of the slides (and nice pictures) goes to:
- FSL tutorials-
  - http://www.fmrib.ox.ac.uk/fslcourse/
- UCLA NITP Summer Course (See Monti 7/12 Preprocessing)
  - http://www.brainmapping.org/NITP/Summer2011.php

# Preprocessing Steps

- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Preprocessing Steps

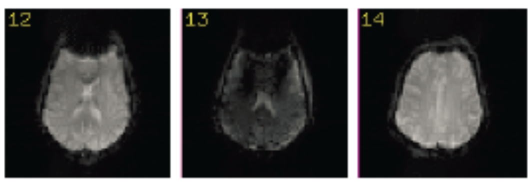
- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

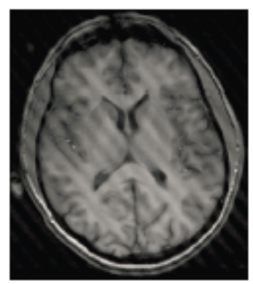
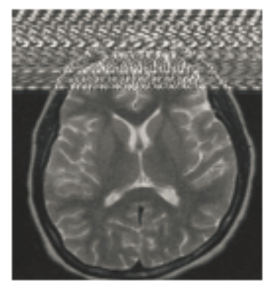
# Pre-Preprocessing

- Scanner output: DICOMS
- Analysis program input: Analyze (SPM), NiFTi (FSL, SPM), BrainVoyager
- SPM, BrainVoyager have built-in functions
- FSL – use MRICRON's dcm2nii convert
  - Can set up preferences on dcm2nii.ini file
  - Put all dcms in one folder
  - dcm2nii –b /Applications/mricronmac/dcm2nii.ini firstfile.dcm

# Pre-Preprocessing

- Image registration
  - Look at your data! Use FSL video mode.
  - Osirix Viewer (or any other DICOM viewer):
    - http://www.osirix-viewer.com/



- Brain extraction
  - Mostly in FSL, use BET toolbox

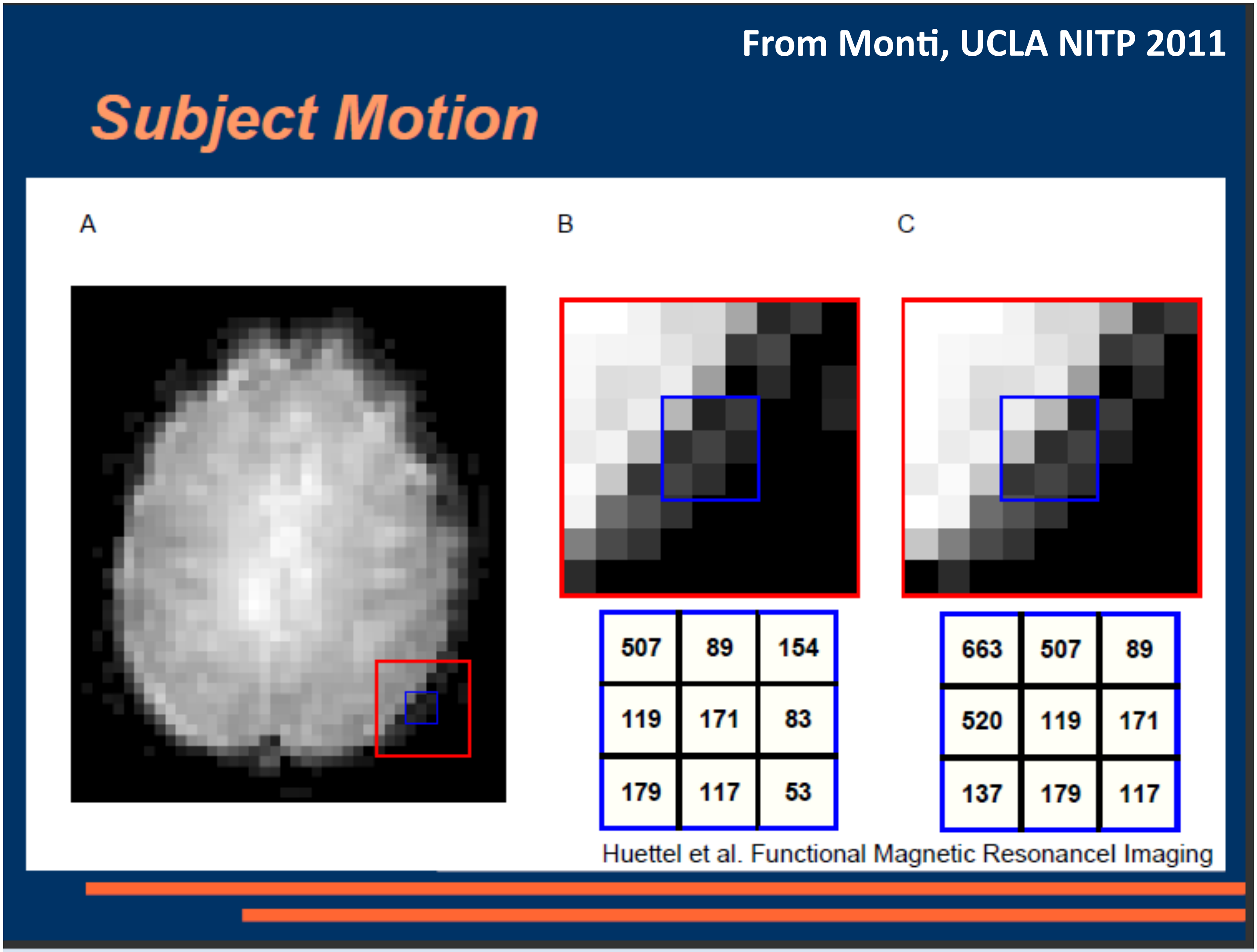
# Preprocessing Steps

- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Motion Correction

- “Please be very, very, very still….”
- Movement that’s 1% of voxel size can induce a 1% signal change, which is sometimes greater than your actual BOLD signal! (say the FSL gods)
- Also, your voxel is no longer the same voxel…

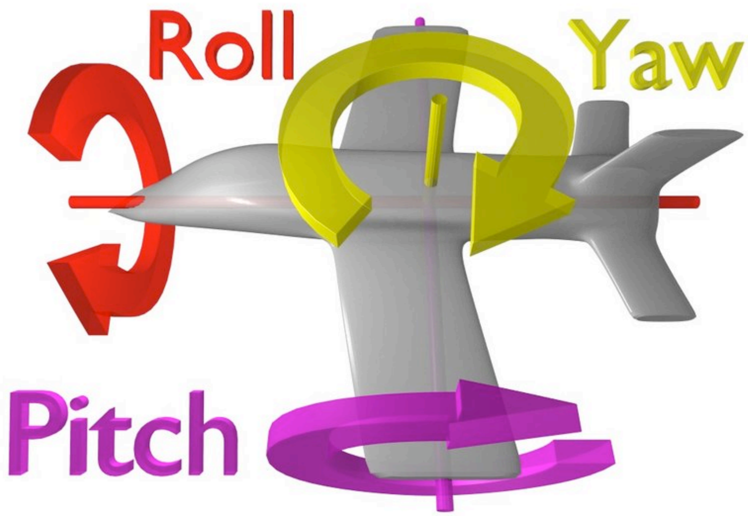
# Motion Correction



# Motion Correction

- Choose a volume to register all the other volumes to (e.g., first, middle, last, mean, standard space; in FSL its middle)
- Within subject: use 6 DOF rigid
- X, y, z, roll, pitch, yaw
- Realigns to reference to

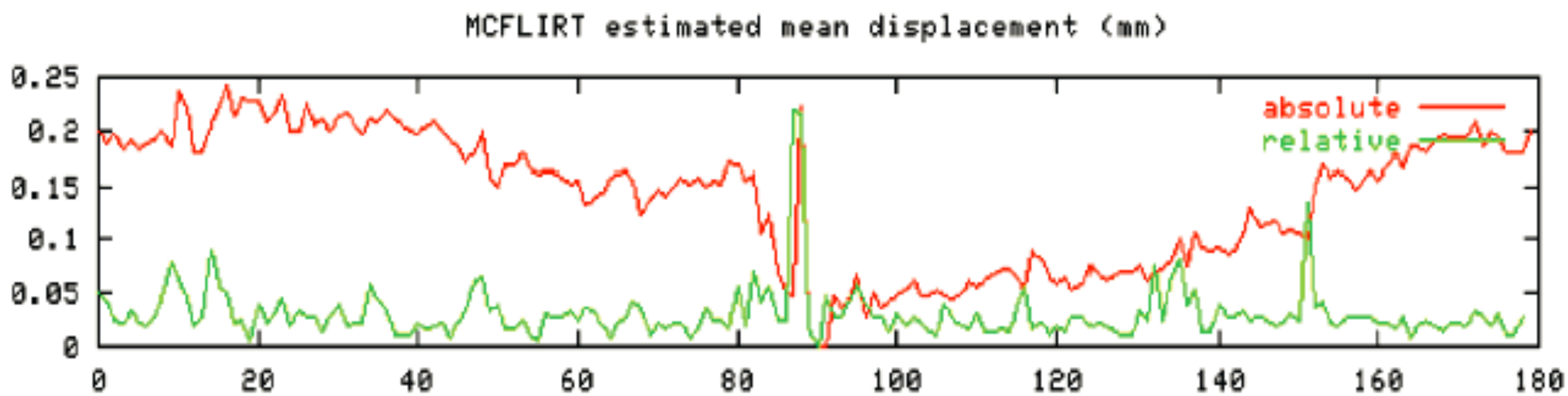
minimize variance



# Motion Correction

- Output (from FSL):

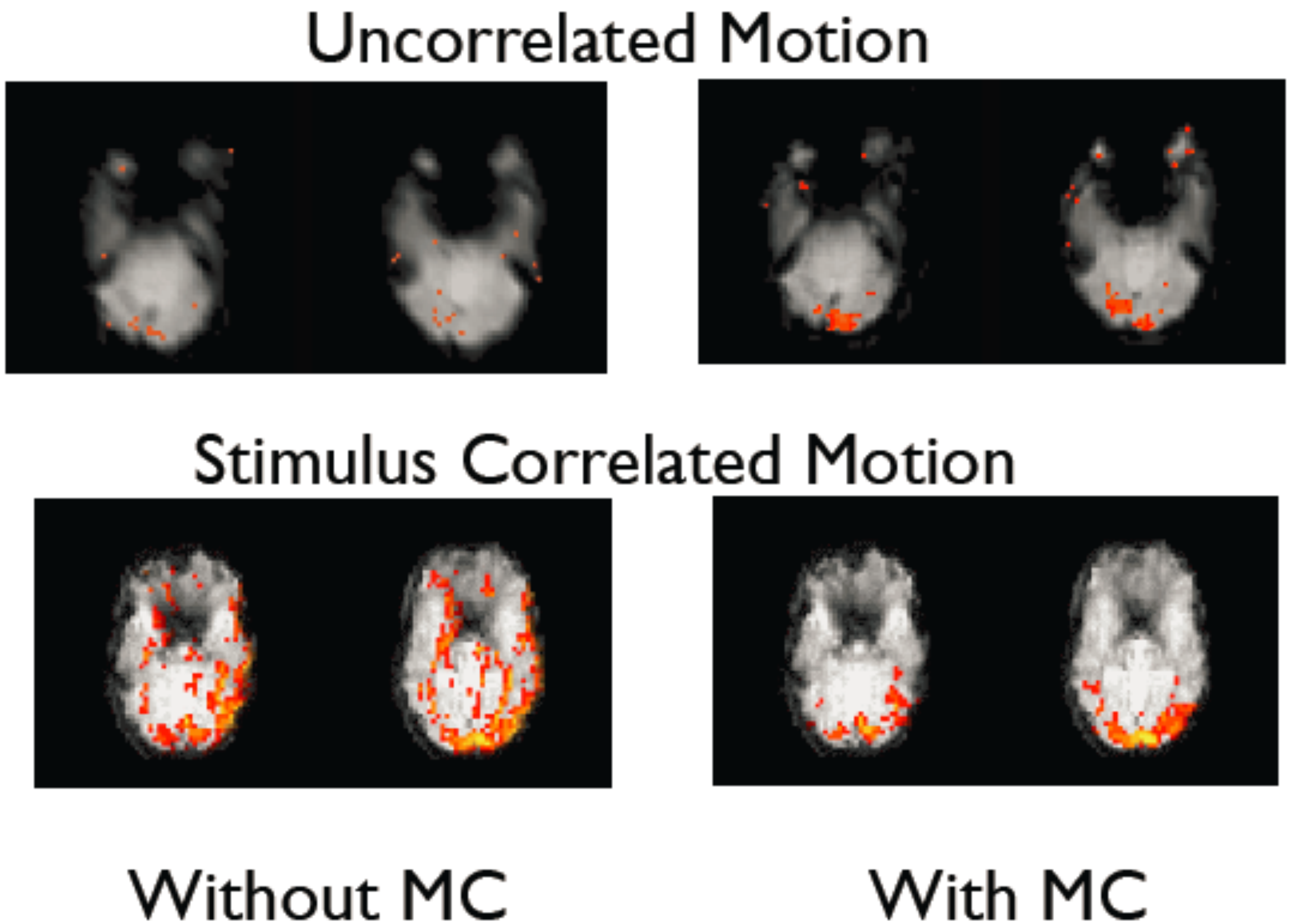
Summary of total motion (relative and absolute)




- Sudden spikes? Motion > 1 voxel size? Or .5?
- Model these 6 regressors!

# Motion Correction

- Results (from FSL):



# Motion Correction

- Other notes:
  - Can prevent with training, make subjects feel relaxed (saying “don’t move!” usually makes them more nevous!)
  - Give them a minute to get settled and comfortable
  - Make sure task isn’t correlated w/ motion
  - Too much motion? Throw out run/subject.. ☹

# Preprocessing Steps

- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Slice-Timing Correction

- Scanners (like our Siemens 3T) may acquire slices in an interleaved fashion (e.g., 0,2,4..1,3,5) to avoid contaminating neighboring slice
- Can correct and put slices back in order
- Current consensus is no need (FSL,Monti)
  - It's another interpolation (the less, the better)
  - It doesn't help that much, could make things worse
  - **Instead, add TEMPORAL DERIVATIVE to model**

# Slice-Timing Correction

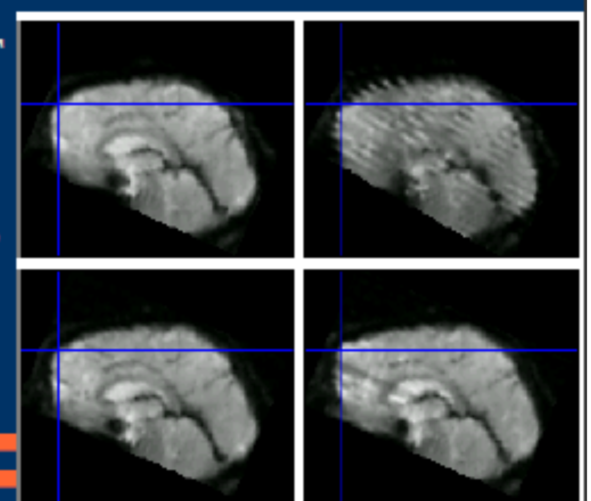
From Monti, UCLA NITP 2011

## *Slice timing correction*

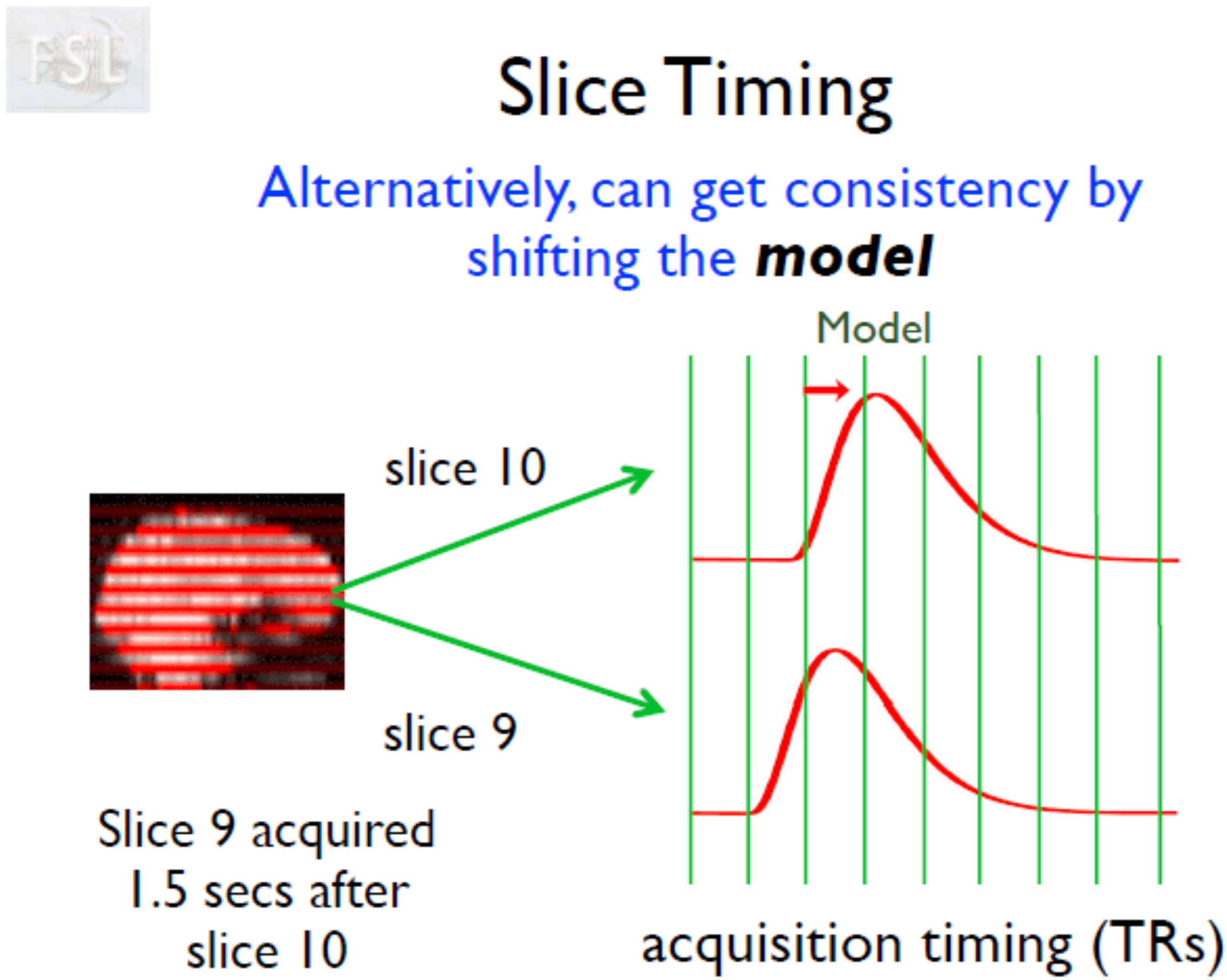
- Most people now suggest not to do it

i. Not all that helpful & requires interpolation
ii. It may worsen artefacts (e.g., smearing spikes)
iii. Interacts in unpredictable ways with motion correction
iv. We spatially smooth across proximal slices
v. Mismatching TR and task
vi. Include temporal derivative of HRF

- What order? Ascending, descending, contiguous, interleaved.



# Slice-Timing Correction



# Slice-Timing Correction

## Slice Timing

One way to shift the model is to use the ***temporal derivative in the GLM***

Based on Taylor approx:
$m(t+a) = m(t) + a.m'(t)$

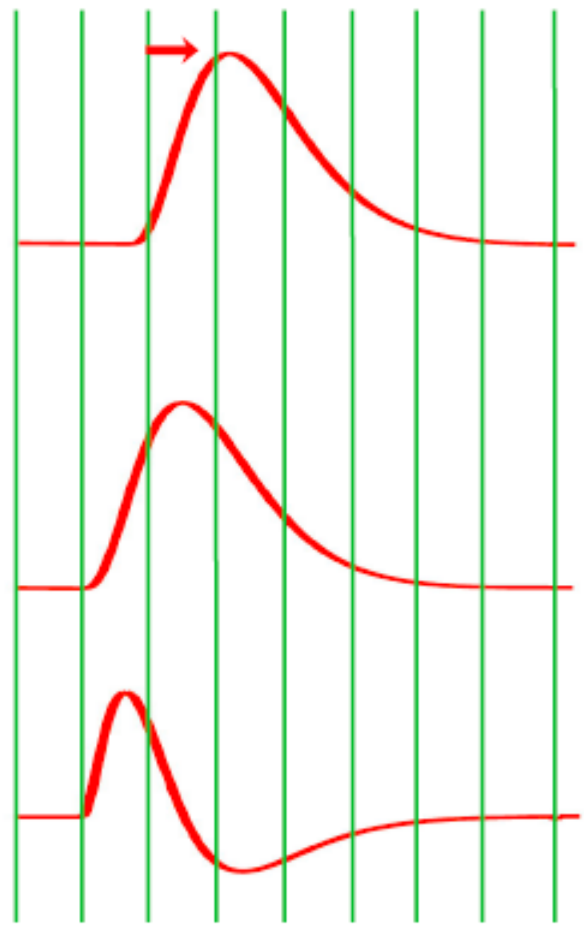
Shifted Model

=

Original Model

-

Temporal Derivative



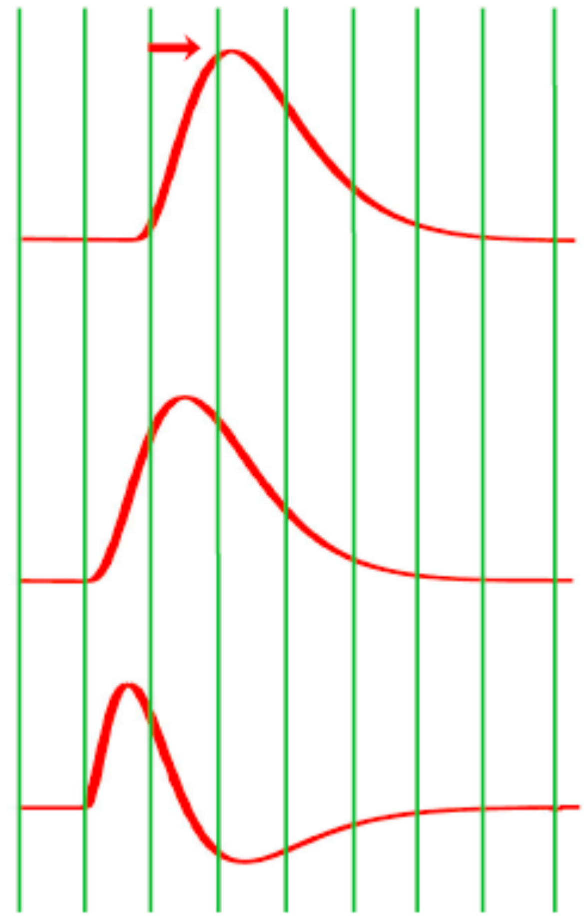
# Slice-Timing Correction

## Slice Timing

Shifting the model also accounts for variations in the HRF delay

- as the HRF is known to vary between subjects, sessions, etc.

This is the recommended solution for slice timing



# Preprocessing Steps

- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Spatial Filtering (Smoothing)

- Averages one voxel's values with its neighbors
- Minimum "smoothness" needed if using Gaussian random field theory (FSL)
- PROS: can increase SNR by decreasing variance
- CONS: may reduce signal if small activations
  - If you expect this, use a smaller kernel
  - Also reduces spatial resolution

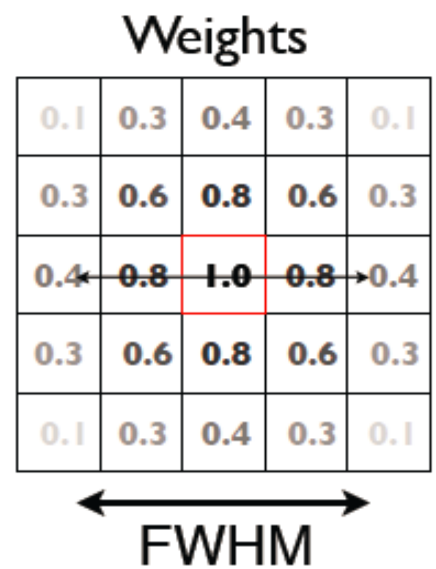
# Spatial Filtering (Smoothing)

- Gaussian Full Width Half Maximum (FWHM) kernel (from FSL tutorial)

Spatial filtering done by a 3D convolution with a Gaussian (cf. 1D convolution with HRF for model)

Each voxel intensity is replaced by a *weighted* average of neighbouring intensities

A Gaussian function in 3D specifies weightings and neighbourhood size




Specify amount by Full Width Half Maximum (FWHM) = distance between 0.5 values

# Spatial Filtering (Smoothing)

- On average, a kernel of 5 mm is adequate
  - Pick something about 2.5x FWHM
  - Can go up to 10-15 mm if expecting large activations
  - Or not use (other options for thresholding)

# Preprocessing Steps

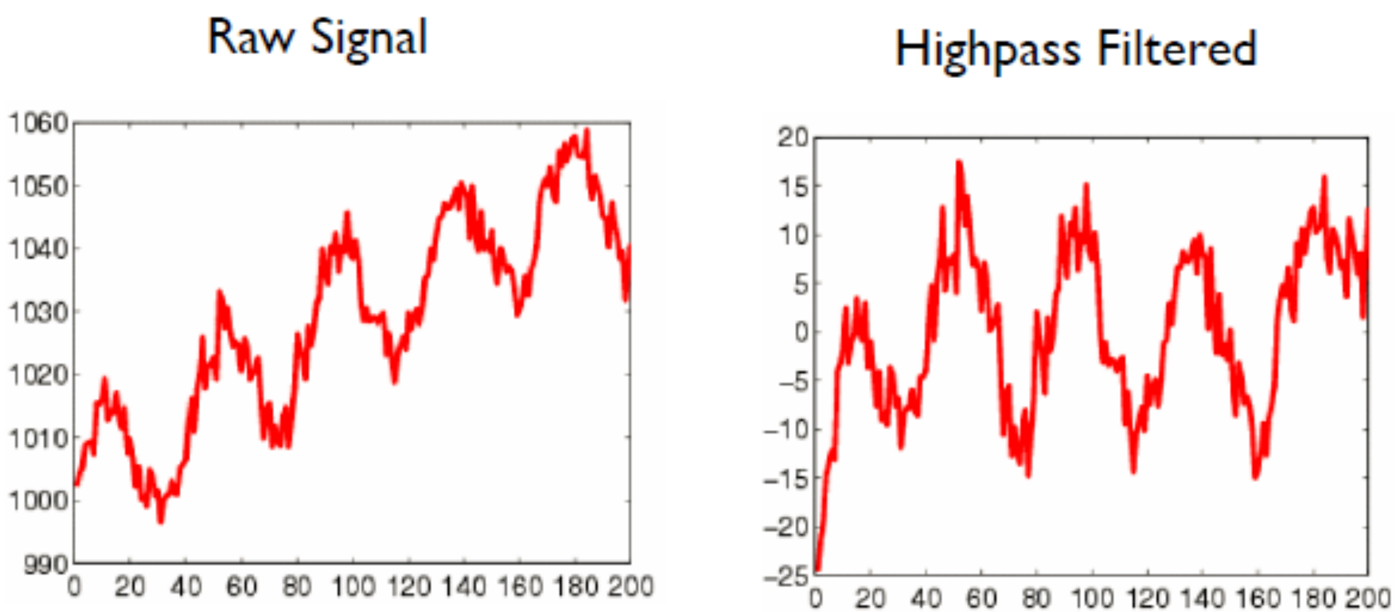
- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Temporal Filtering

- Temporal noise:
  - Temporal drift from scanner; physiological cycles (cardiac, repiratory)
- These can mask your actual signal!
- High pass filter lets high frequencies through
- Low pass filter lets low frequencies through (removes high-frequency fluctuations) but can remove signal, esp if event-related.
- Usually just use high pass – FSL uses prewhitening to avoid low-pass

# Temporal Filtering

- Use the high-pass filter to remove low-frequency (e.g., long, slow) noise (from FSL):




Removes low frequency signals, including linear trend

# Temporal Filtering

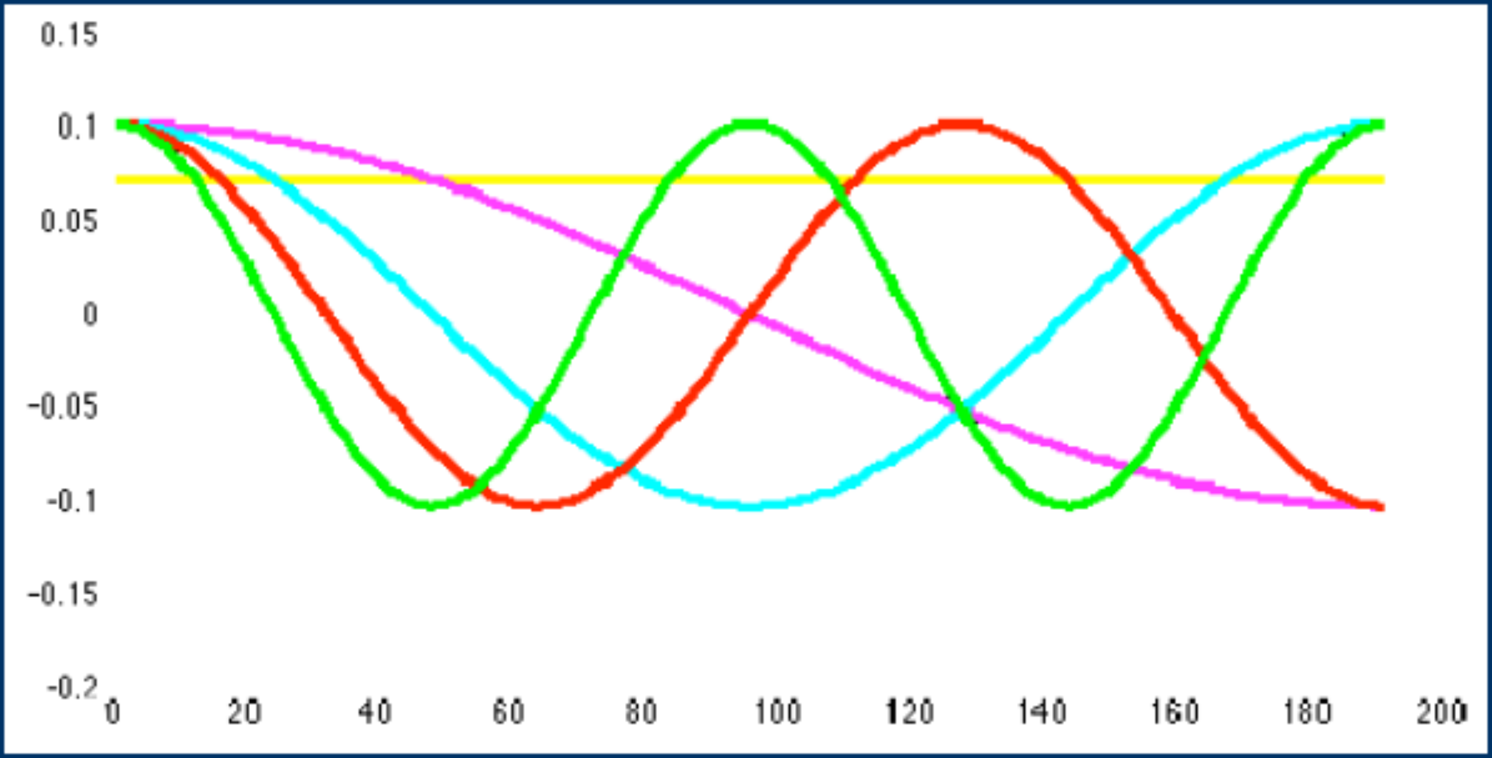
- Generally a HP Filter of 100 s is adquate
  - Change if your model is different (eg very long event)
- SPM: models low drifts to catch variance; uses cosine basis set
- FSL: removes low drifts and then convolves signal with gaussian-weighted running line

# Temporal Filtering

From Monti, UCLA NITP 2011

## *HP Filtering Strategy I: SPM*

- Model low drifts to "soak up" their variance (using a discrete cosine transform basis set).



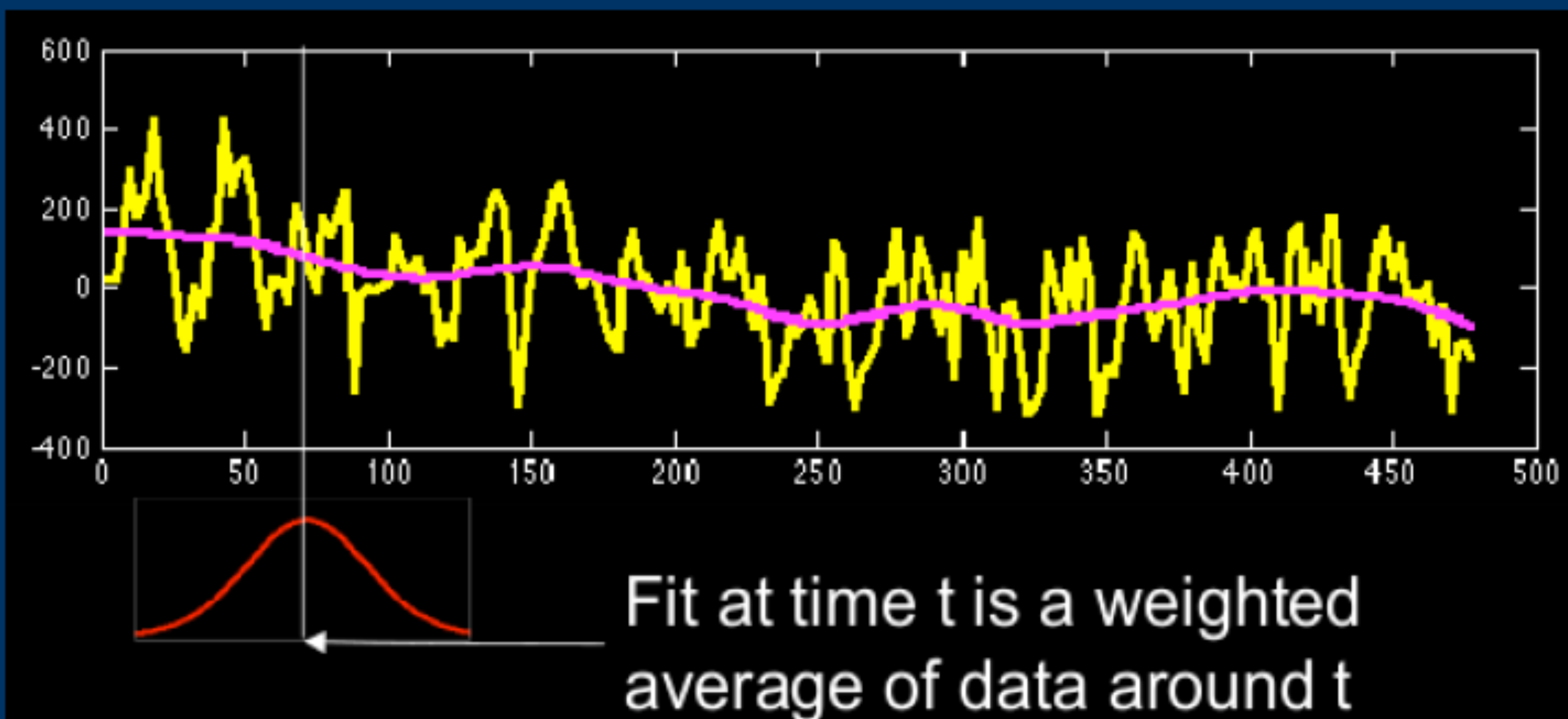
# Temporal Filtering

**From Monti, UCLA NITP 2011**

## *HP Filtering Strategy II: FSL*

- Remove low drifts from the signal:

  i. Fit a Gaussian-weighted running line



# Preprocessing Steps

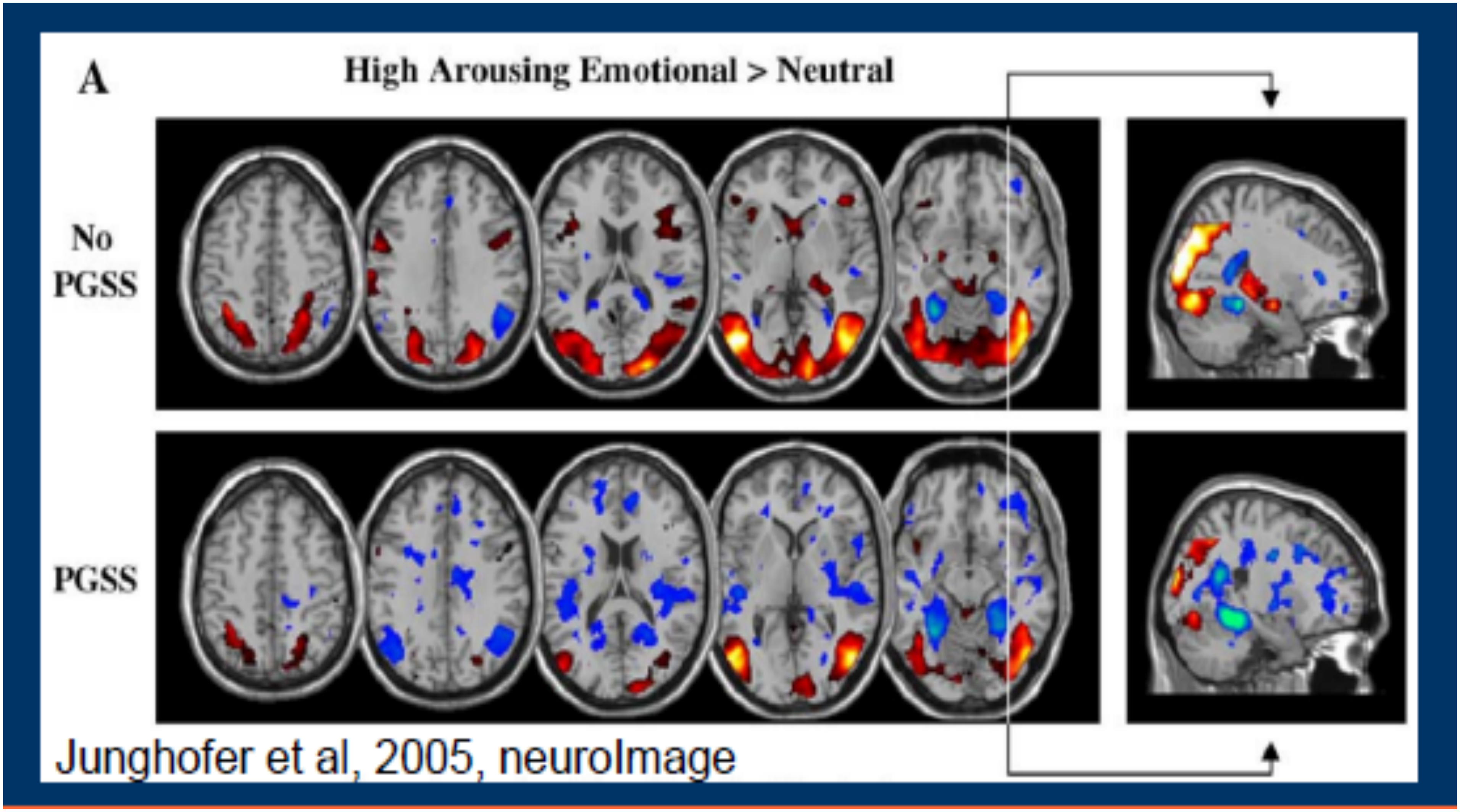
- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- Registration/Normalizing (technically post-preprocessing)

# Global Intensity Normalization

- From Monti 2011 slides:
- GOOD:
  - Between-sessions (grand mean scaling) so you are comparing all runs by a single factor (can be arbitrary) centered on the same mean
- BAD:
  - Within-session (global scaling), forces each VOLUME to have same mean intensity, that's silly

# Global Intensity Normalization

- Don't do global scaling!



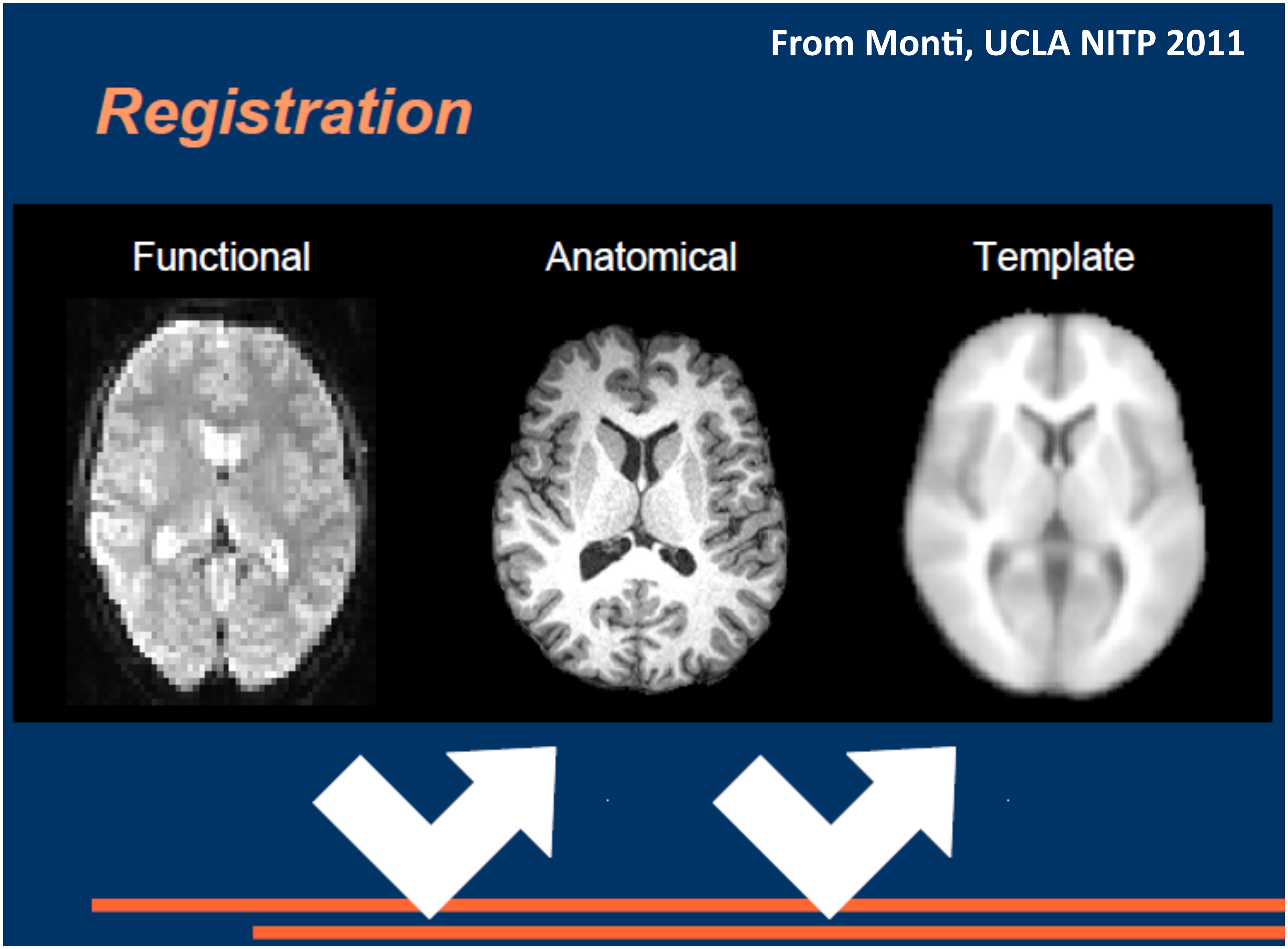
# Preprocessing Steps

- Pre-Preprocessing
  - DICOM transformation, Image reconstruction, BET
- Motion correction
- Slice-timing correction
- Spatial filtering
- Temporal filtering
- Global intensity normalization
- **ICA Denoising
- Registration/Normalizing (technically post-preprocessing)

# ICA Denoising

- Manually remove noise components
- Feed raw data into MELODIC (FSL; Multivariate Exploratory Linear Optimised Decomposition into Independent Components)
- Model free – pulls out all the “components”
- Can manually go through components and identify noise! [demo]
- See Kelly 2010 JNeurosci Methods  for details on what to remove
- Especially good if resting state data; or odd movements! Bad if noise relates to signal…

# Registration/Normalization



# Registration

- Allows comparing between individuals by mapping all to a template brain (from Monti 2011 slides)
- Rigid body:
  - 6 DOF – 3 rotations, 3 translations, typically within subj
  - 7 DOF – add global scaling, good for func to anatomical
- Affine:
  - 12 DOF – 3 rotations, 3 translations, 3 scalings, 3 skewings
  - Subject to template rotation
- Non-linear:
  - Represented by a deformation field (not a matrix)

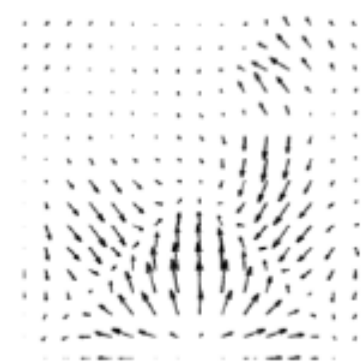
# Registration

From Monti, UCLA NITP 2011

- Non-linear (> 12dof)
  - Can be local
  - Can be constrained (e.g.,, regularization, topology preservation)

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

An affine transformation is represented by these 12 numbers.
This matrix multiplies coordinate vectors to define the transformed coordinates.



A non-linear transformation is represented by a ***deformation field***.

# Registration

- Re: Non-linear transformations
- See FSL specifically on registration and Monti for more detailed information (FSL FLIRT, FNIRT)
- SPM DARTEL is also good (so I hear)
- Lots of options overall!

# Registration

- When registering, you have to interpolate again – what to do with new empty spaces?
  - Nearest neighbor? Trilinear (4 neighbors)? Sinc (10)?

- FSL: analyzes in native space, then normalizes
- SPM: normalizes first, then GLM, decreases variance & SD
- Can account for some differences in results!

# Questions/Discussion?

sliew@usc.edu

# Thanks to Preprocessing Resources

- Full credit for all the info and most of the slides (and nice pictures) goes to:

- FSL tutorials-
  - http://www.fmrib.ox.ac.uk/fslcourse/

- UCLA NITP Summer Course (See Monti 7/12 Preprocessing)
  - http://www.brainmapping.org/NITP/Summer2011.php