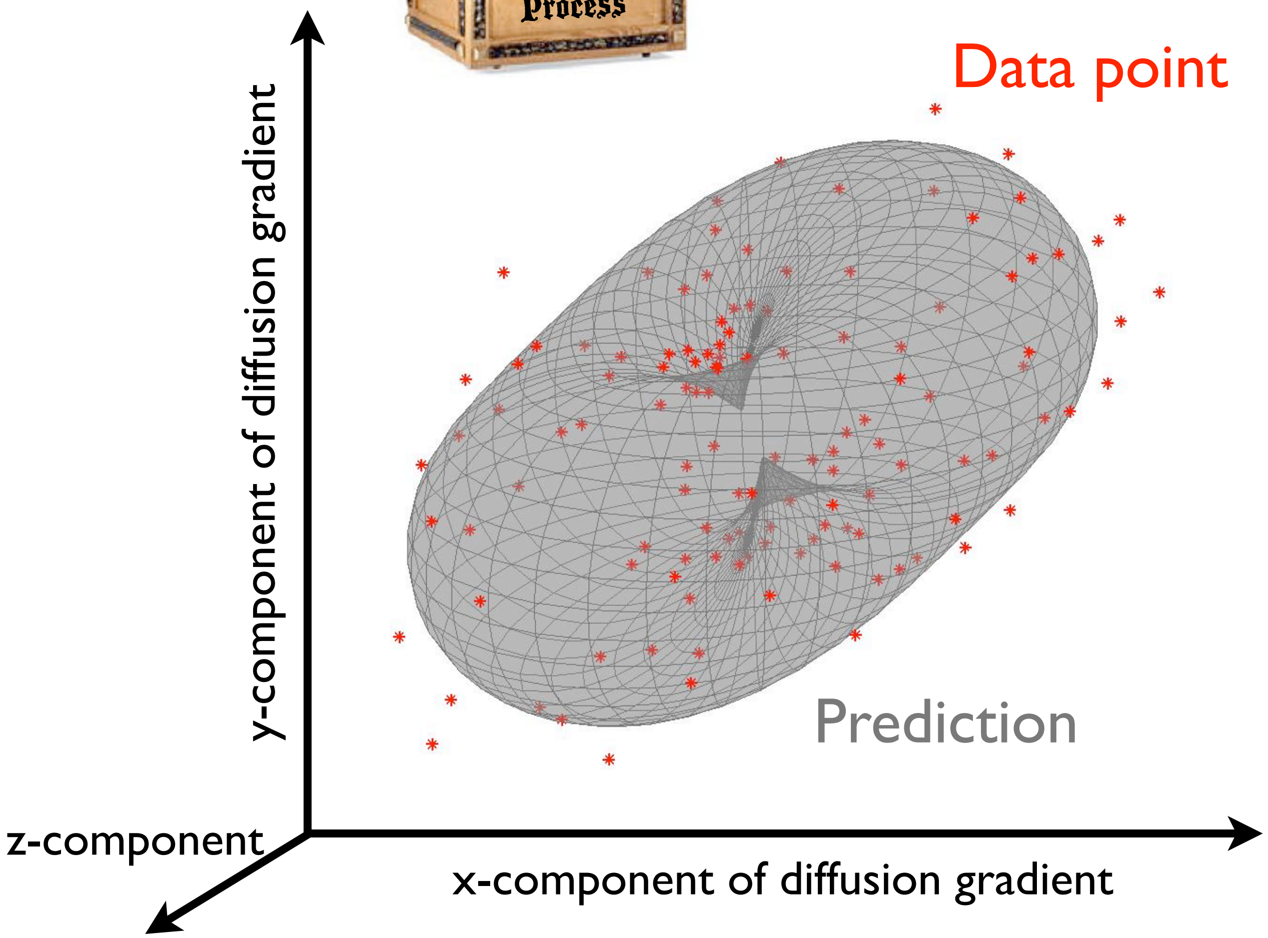
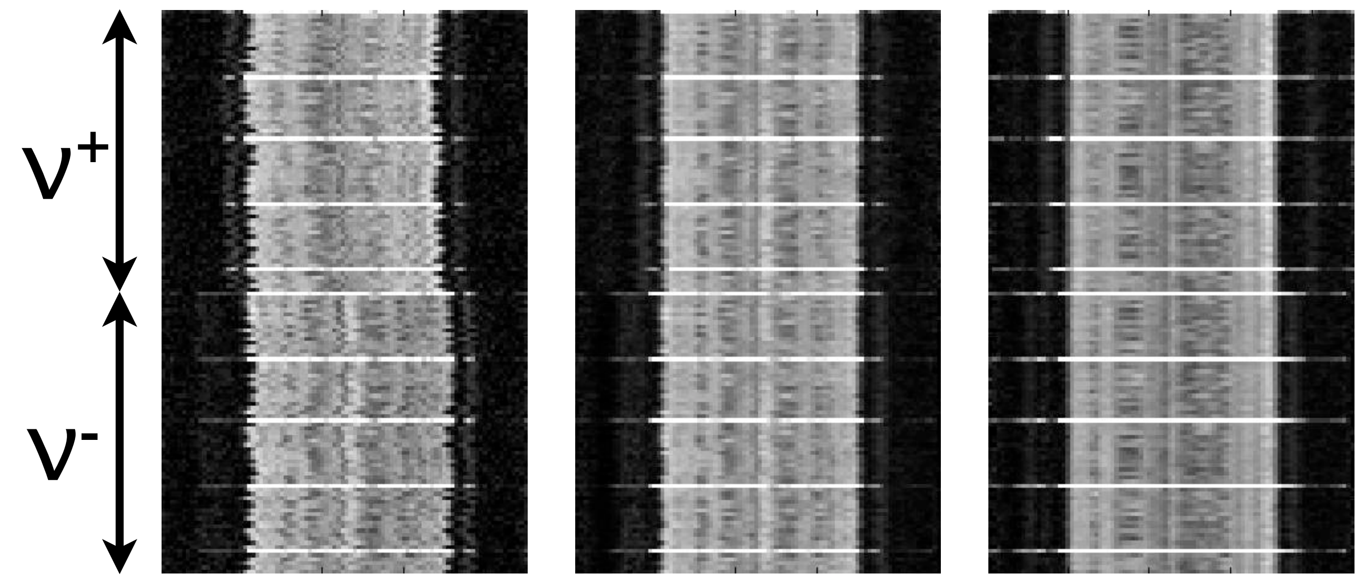
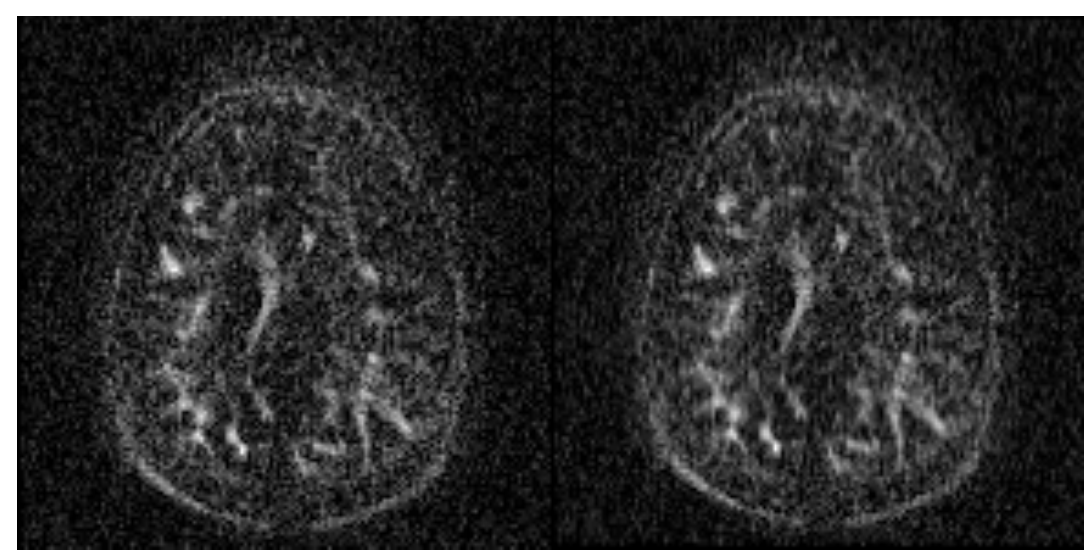
# eddy - An FSL tool for diffusion pre-processing

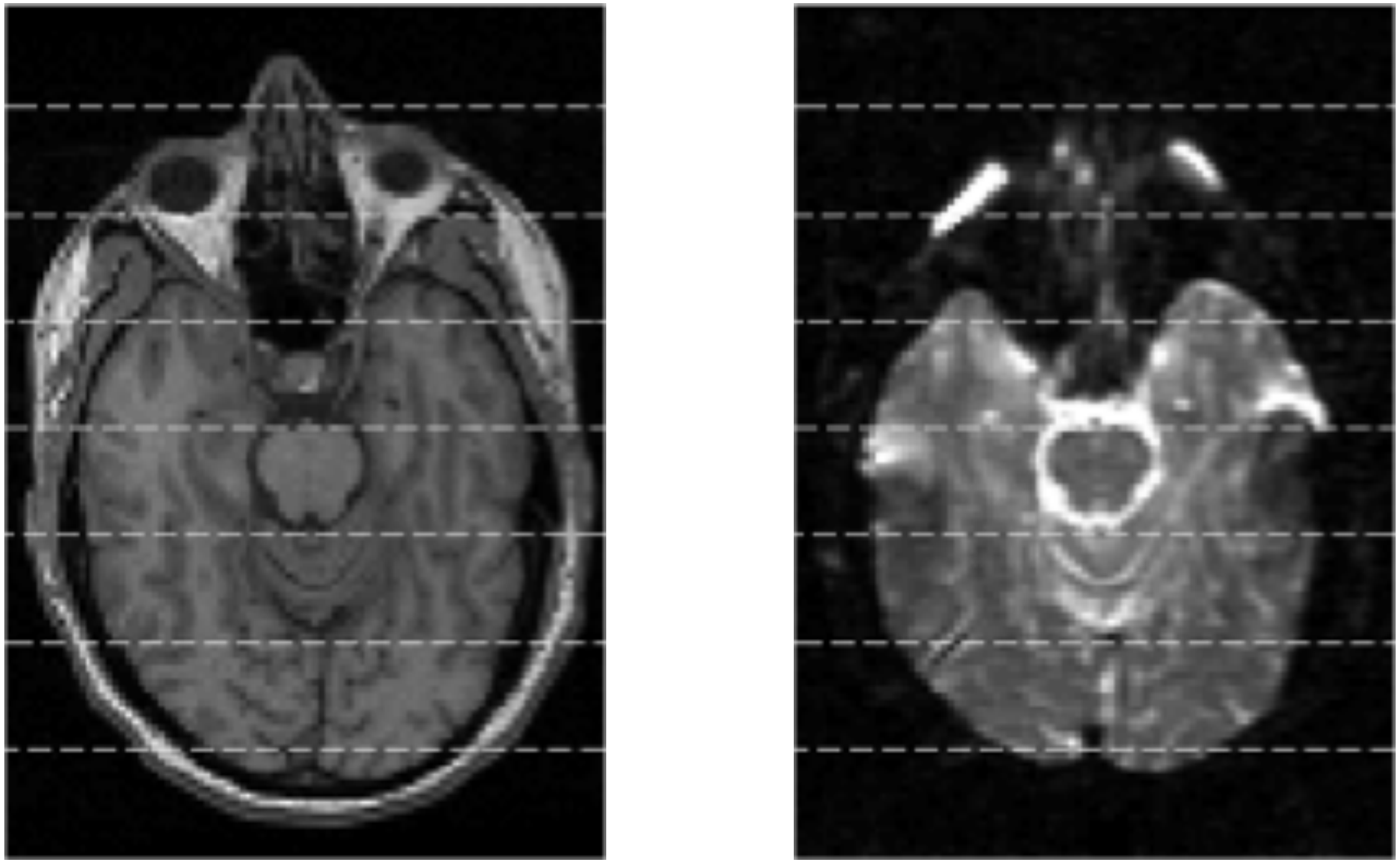



# Outline of the talk

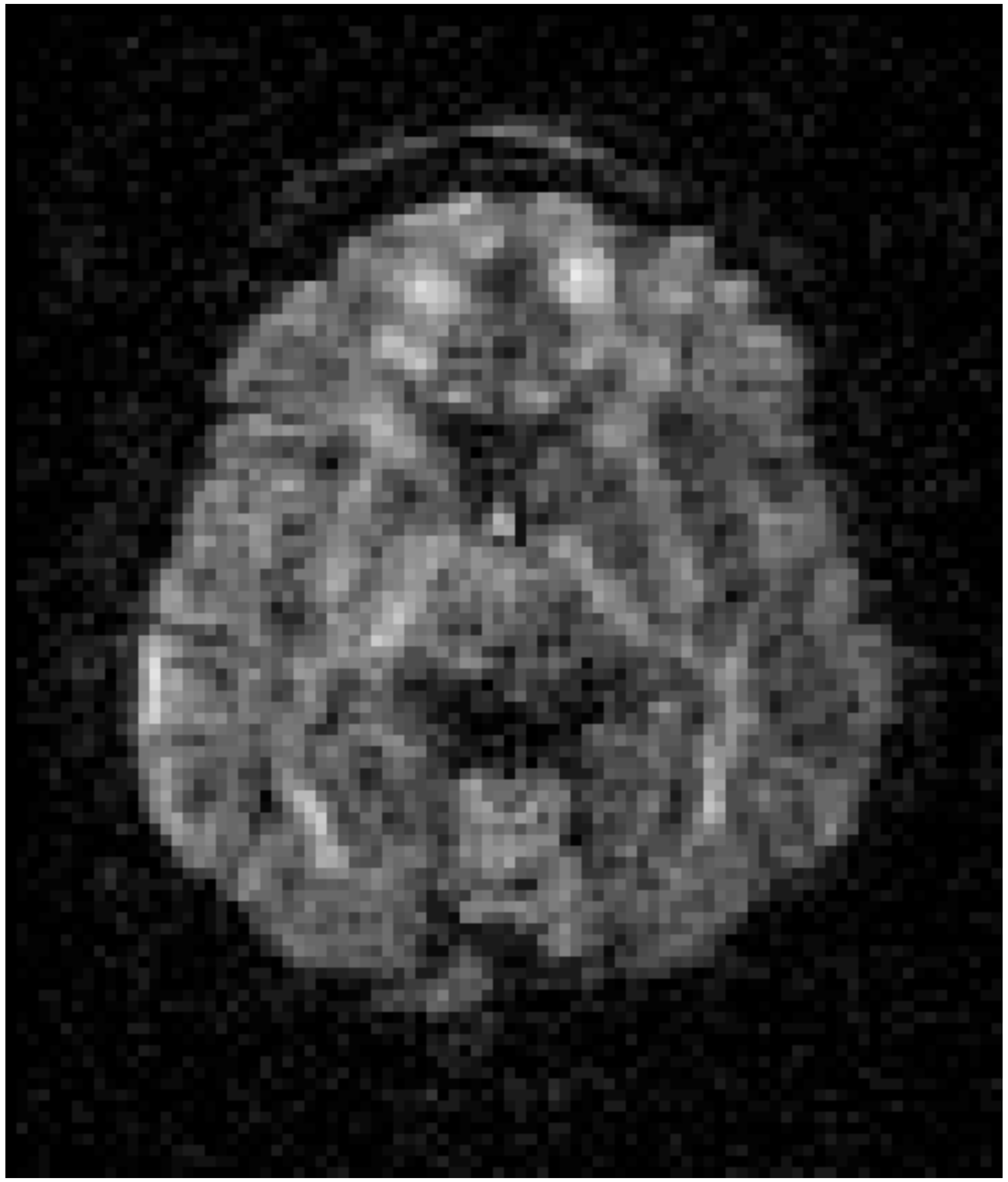
- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# What is the problem with diffusion data?



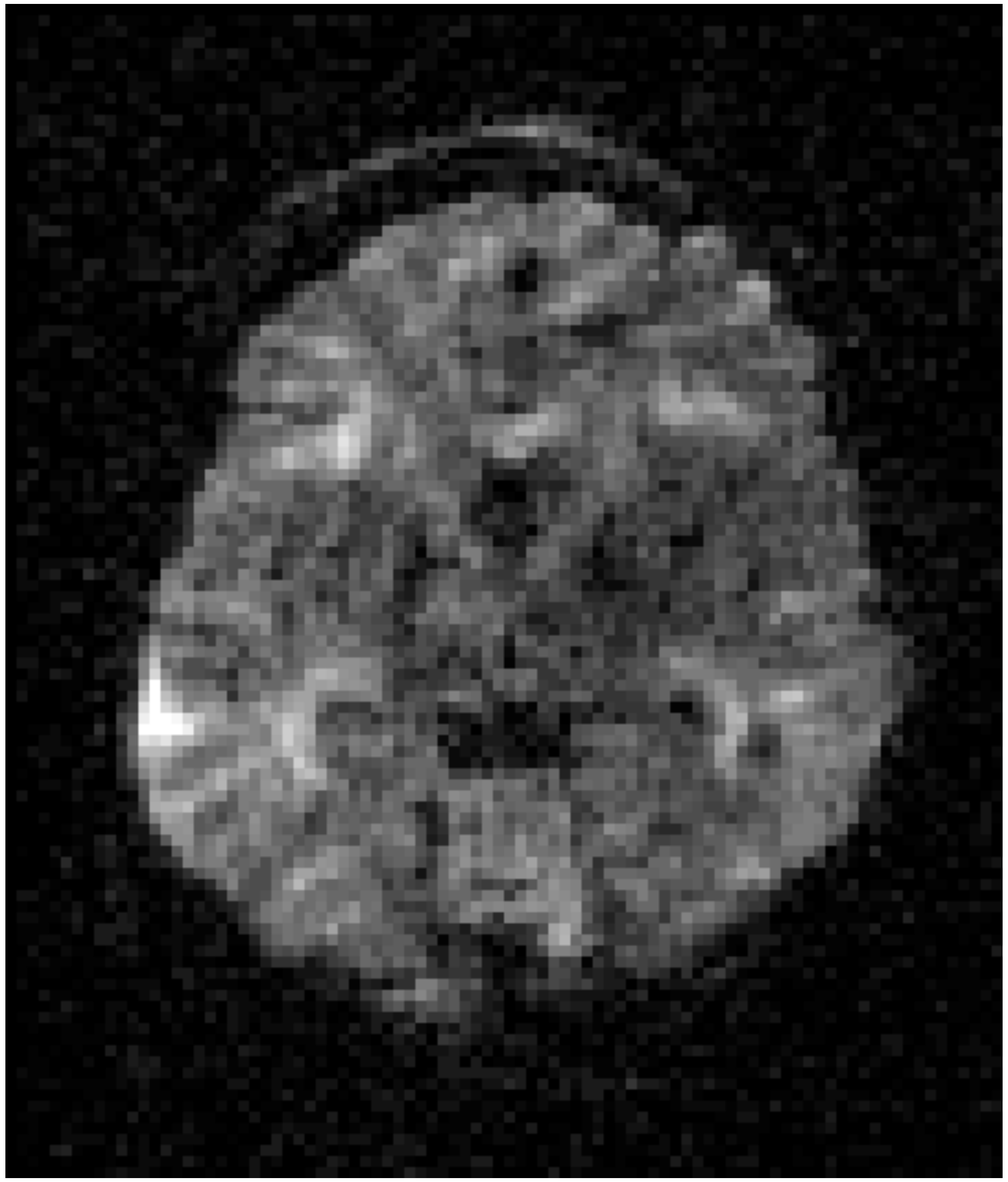
Well, it isn't very anatomically faithful

# What is the problem with diffusion data?



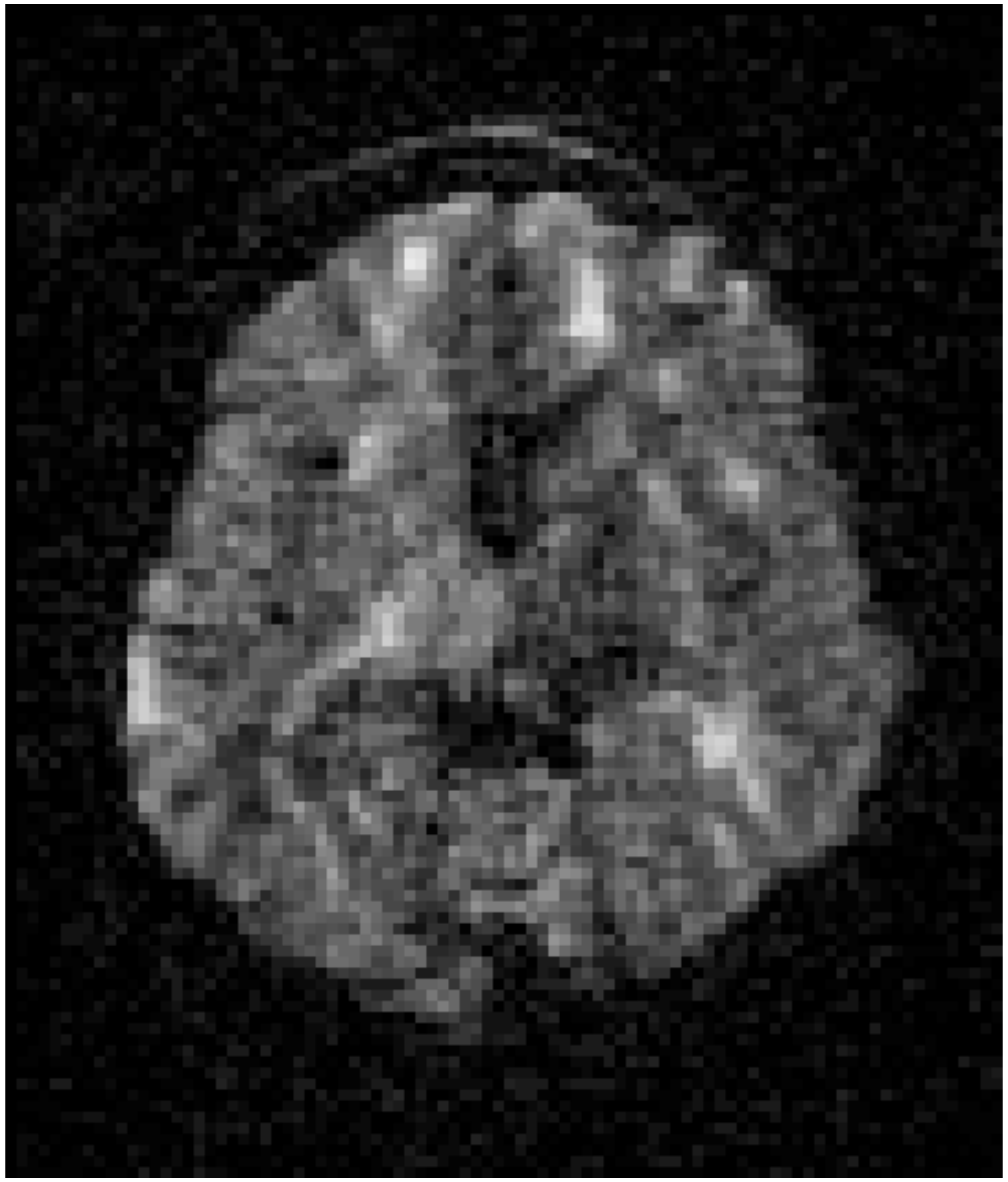
In fact, it isn't even internally consistent

# What is the problem with diffusion data?



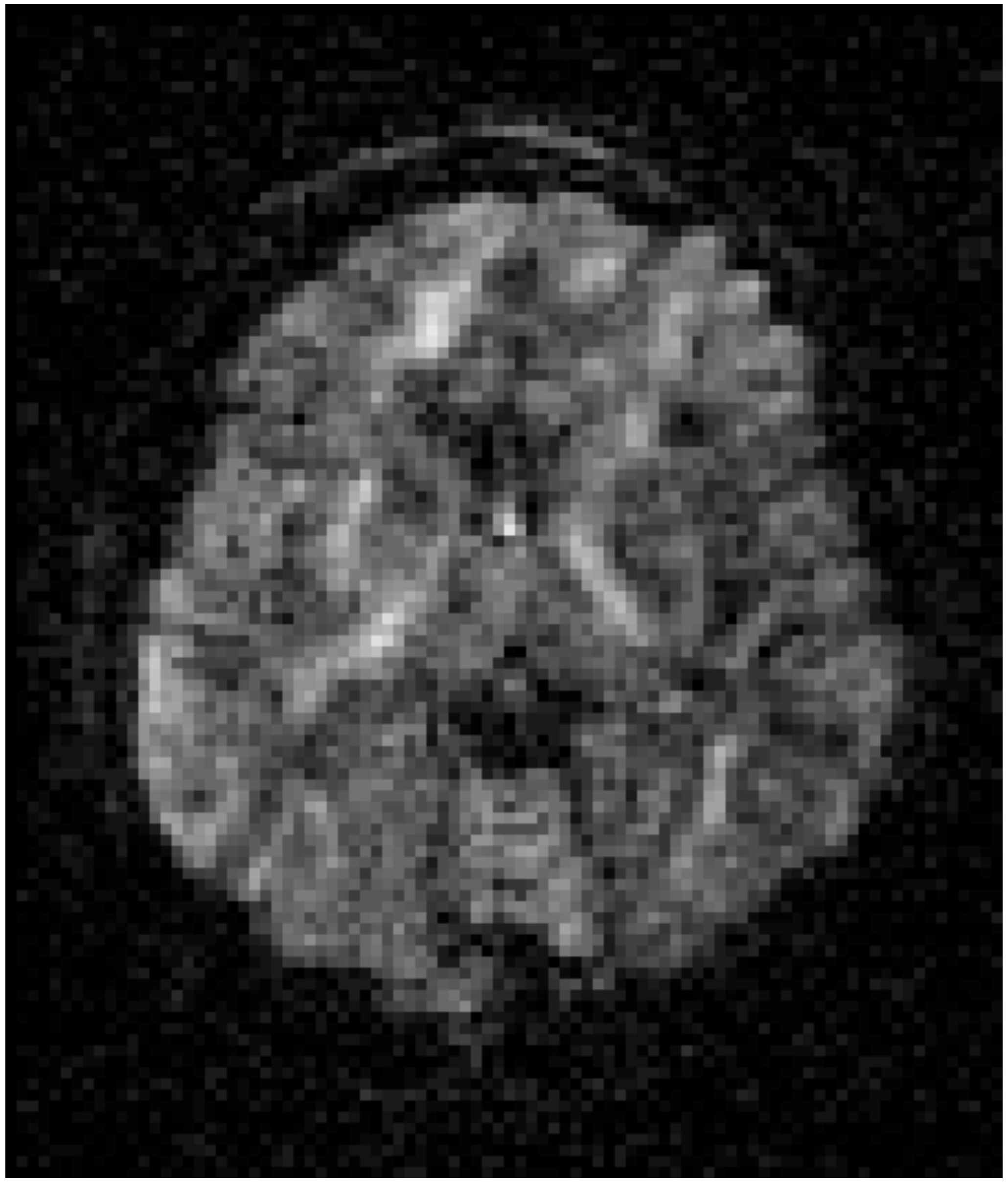
In fact, it isn't even internally consistent

# What is the problem with diffusion data?



In fact, it isn't even internally consistent

# What is the problem with diffusion data?



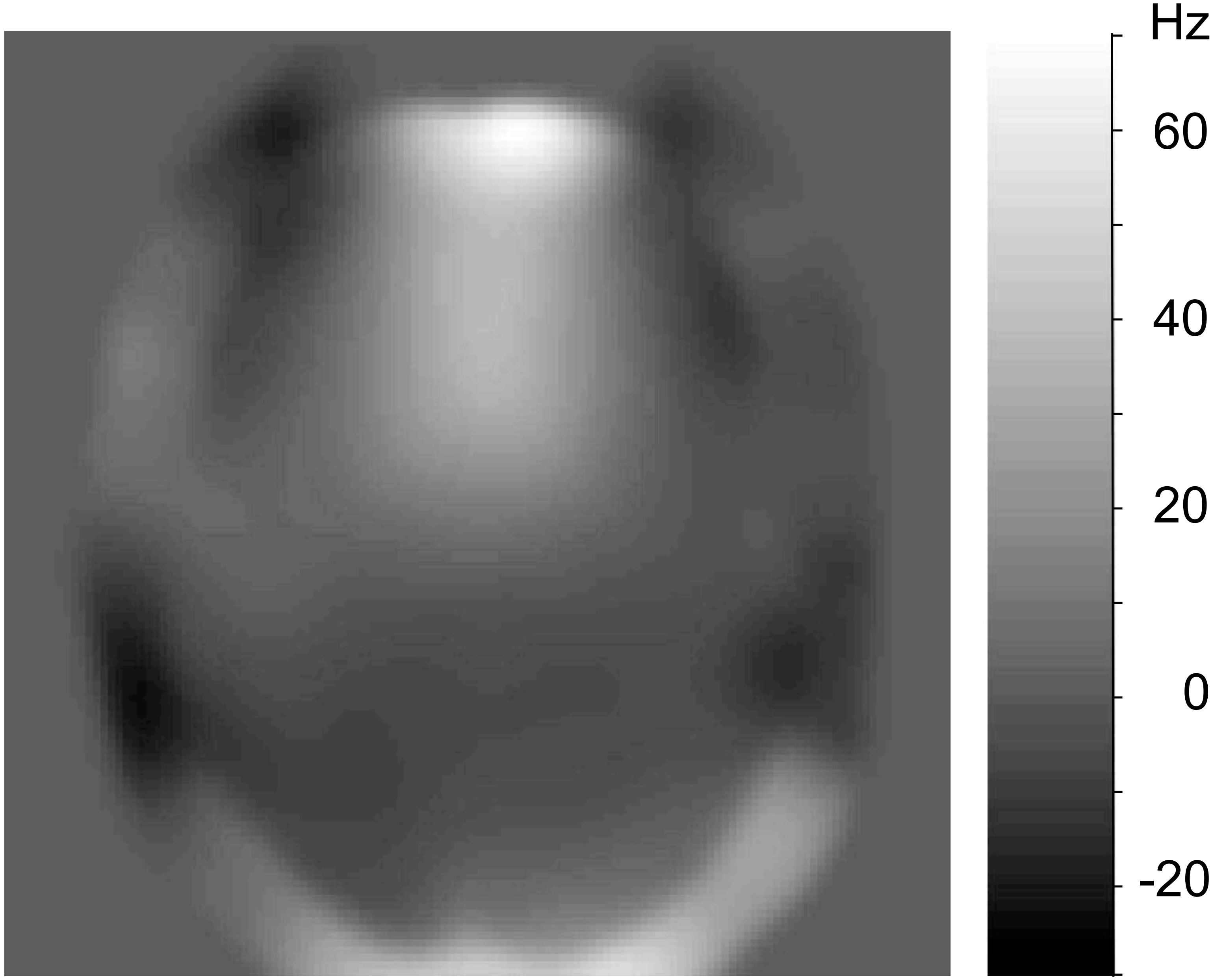
In fact, it isn't even internally consistent

# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

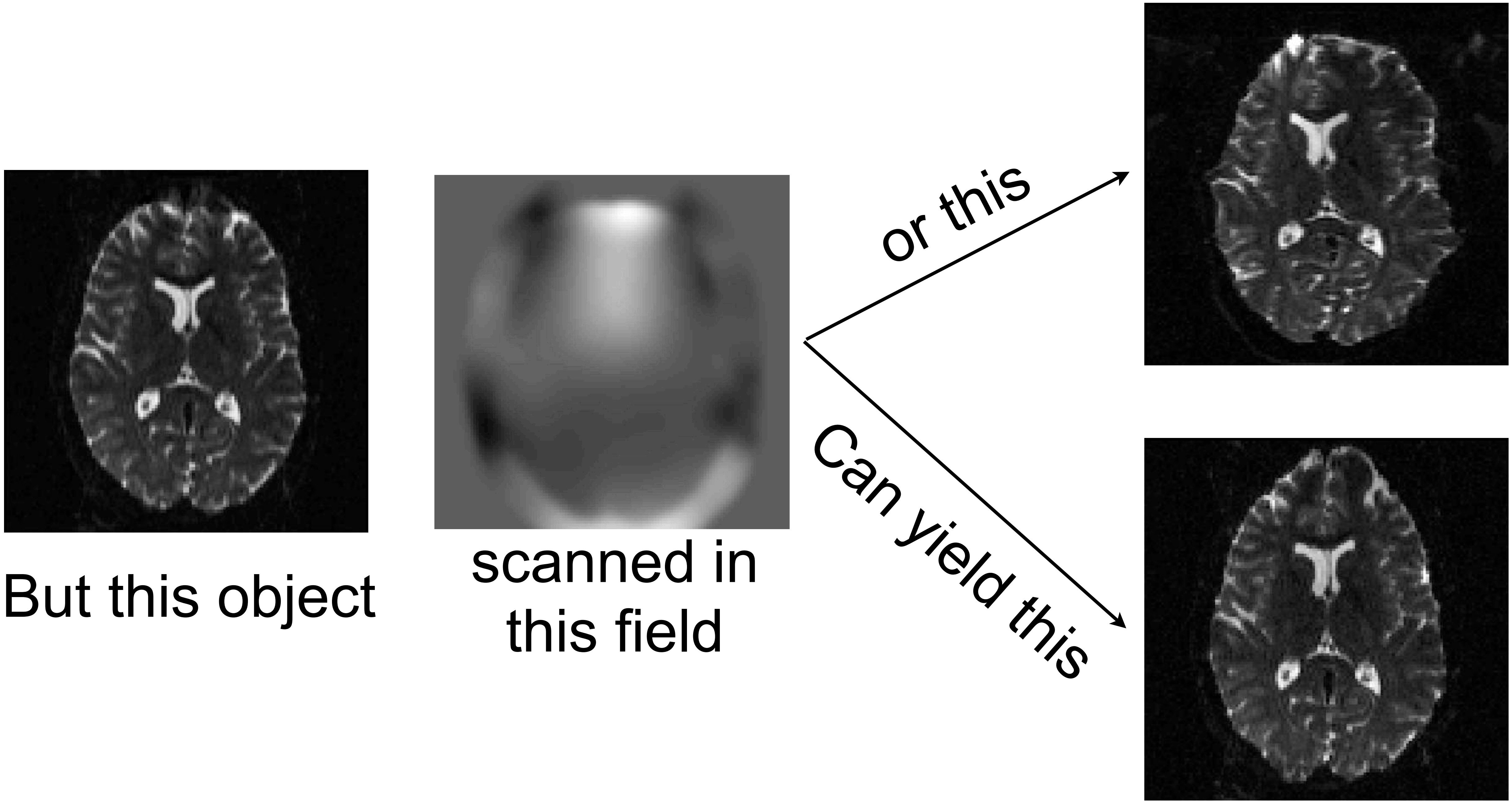
# Off-resonance field ⇒ Distortions

An "off-resonance" field is a map of the difference between what we think the field is and what it really is.




It is all caused by an "off-resonance" field

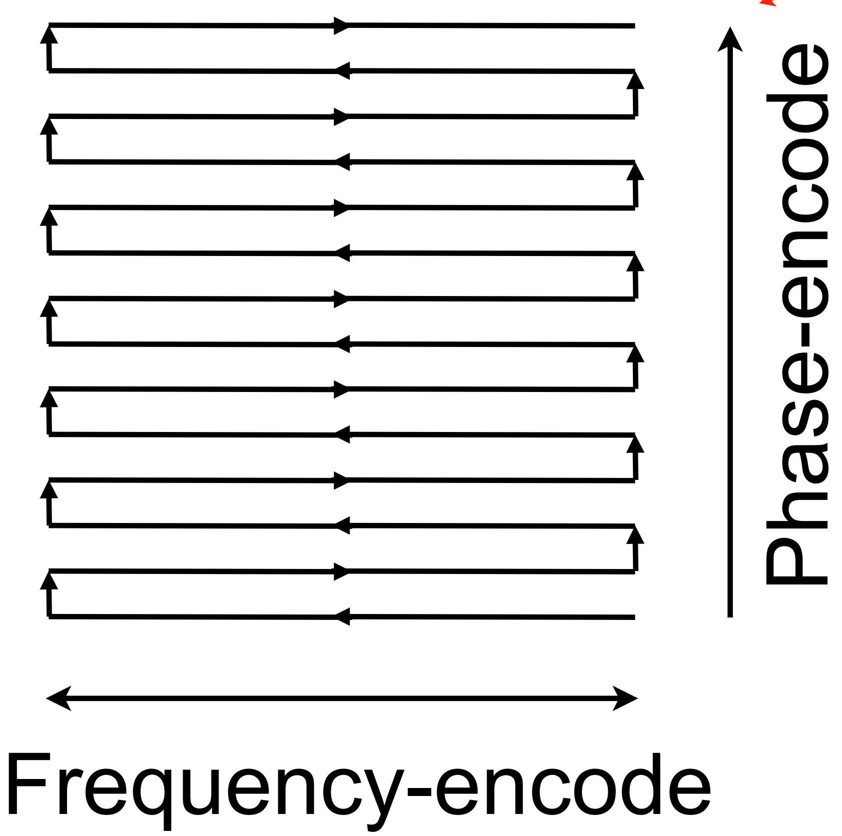
# Off-resonance field ⇒ Distortions




So there is clearly more to this story...

# Off-resonance field ⇒ Distortions

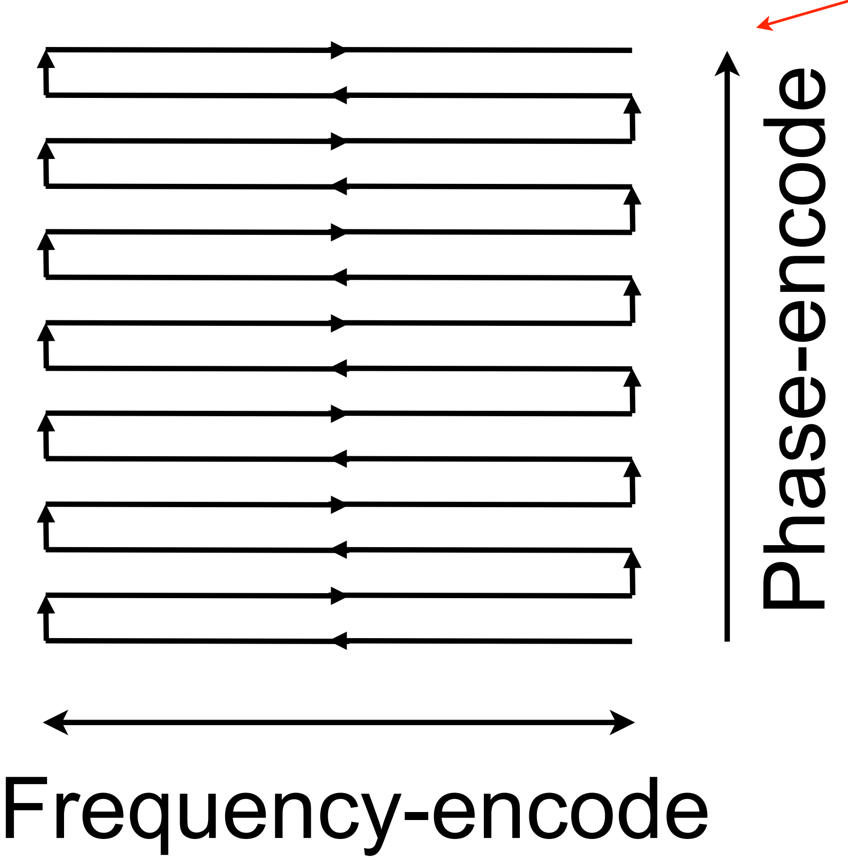
But first, what does this mean?

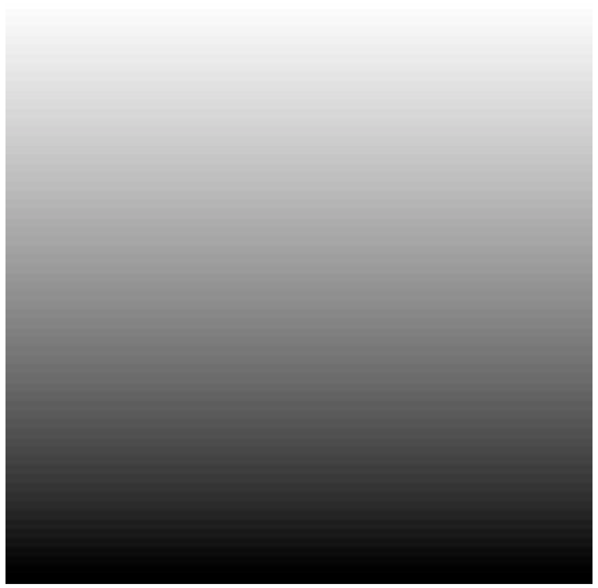



In echo-planar imaging we traverse one direction fast and the other very slowly.

# Off-resonance field ⇒ Distortions

But first, what does this mean?

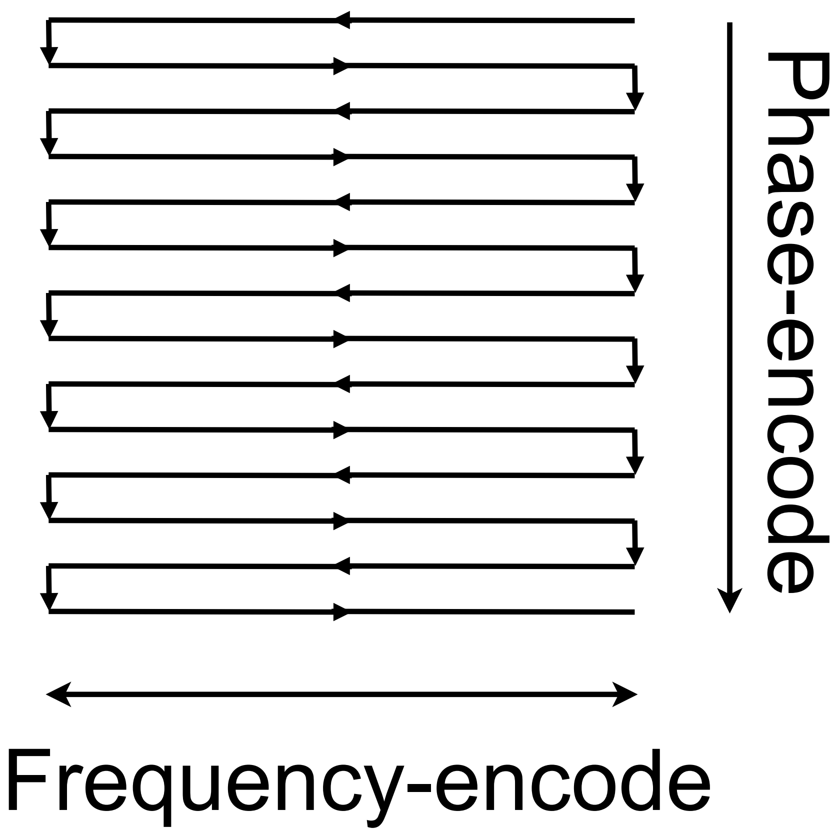


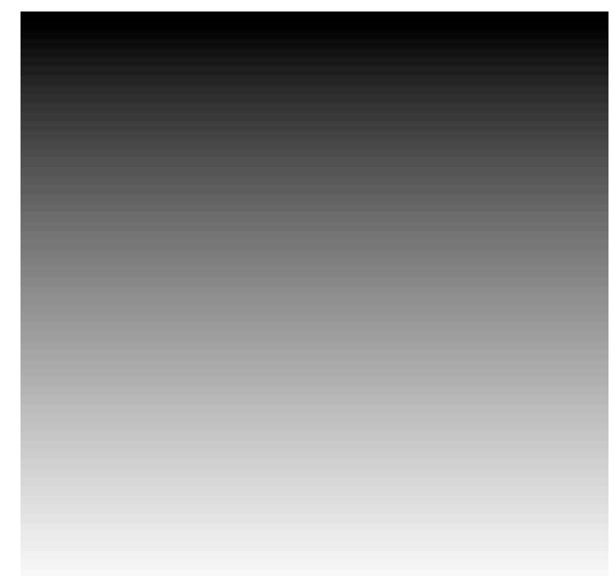
It means that for each PE-step we switch “this” on.

So when we “see” a high frequency, what do we infer?

# Off-resonance field ⇒ Distortions



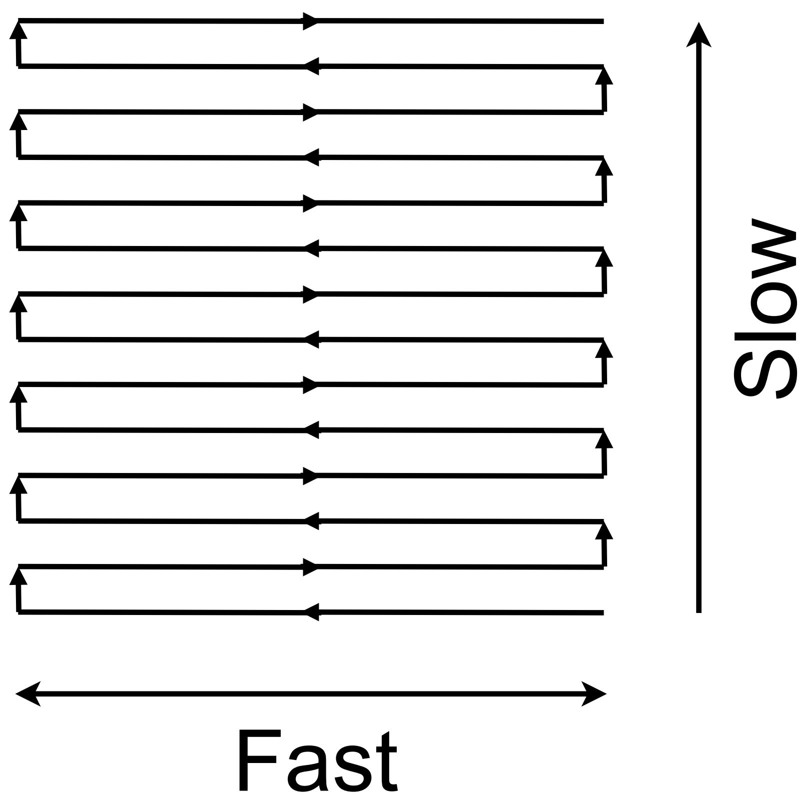

Conversely we could

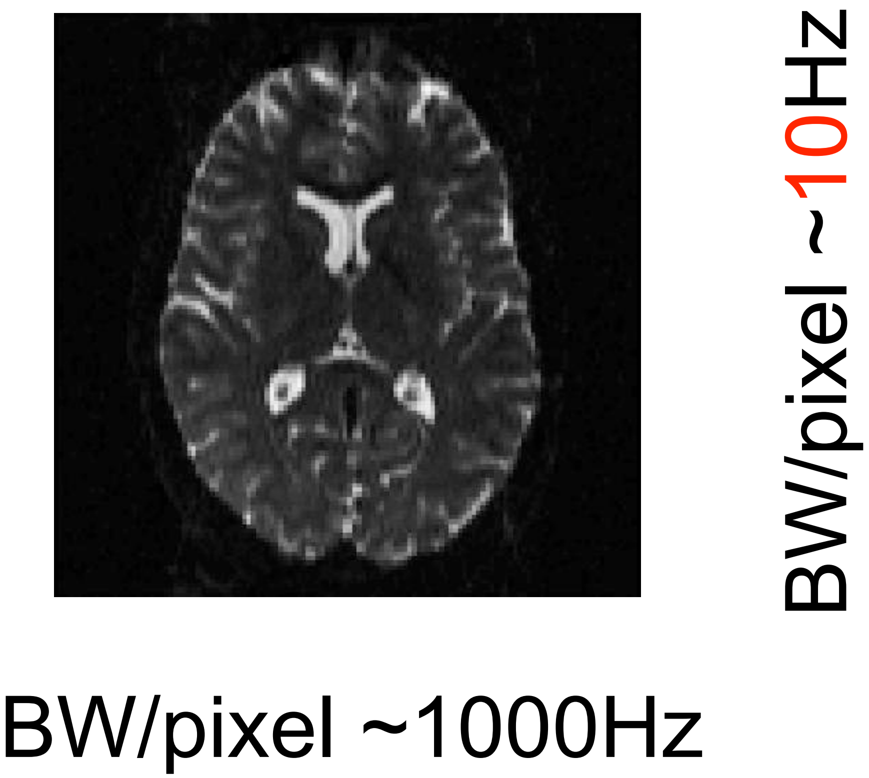


switch “this” on for each PE-step.

NOW, when we “see” a high frequency, what do we infer?

# Off-resonance field ⇒ Distortions

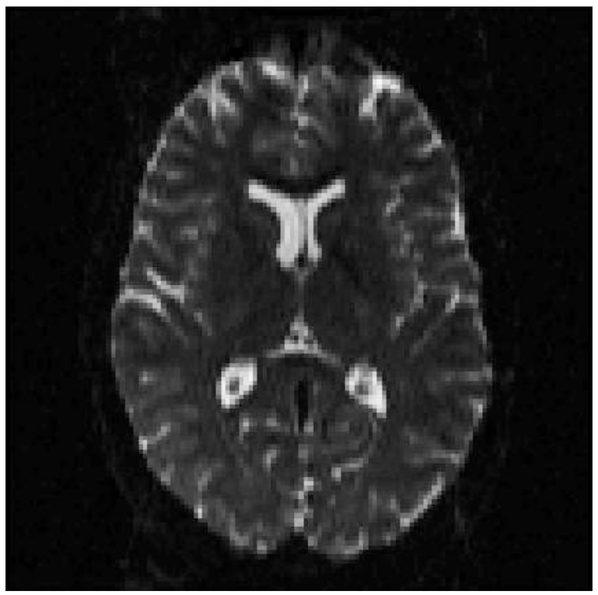


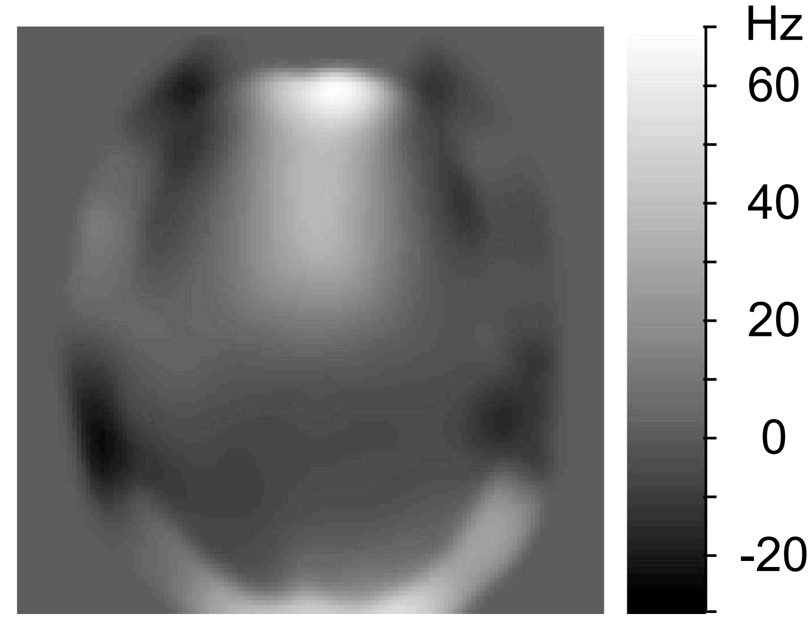

A Band Width/pixel of 1000Hz means that in order to get the location wrong by one pixel we must get the frequency wrong by 1000Hz.

And a BW/pixel of 10Hz means that in order to get the location wrong by one pixel we must get the frequency wrong by 10Hz.

# Off-resonance field ⇒ Distortions



BW/pixel ~10Hz

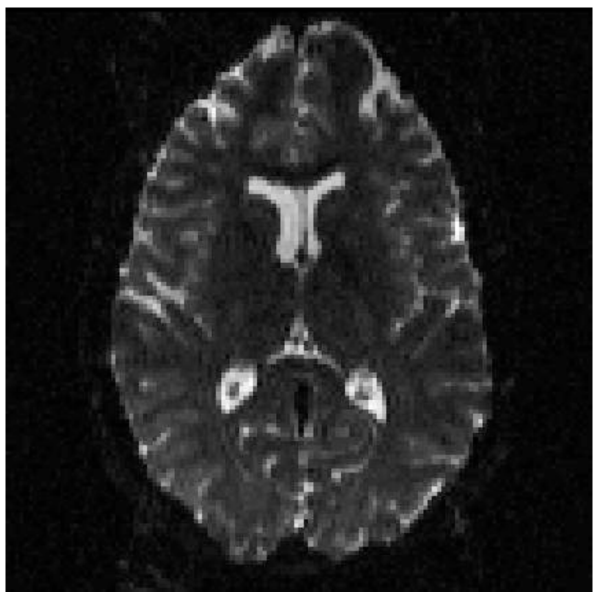



BW/pixel ~1000Hz

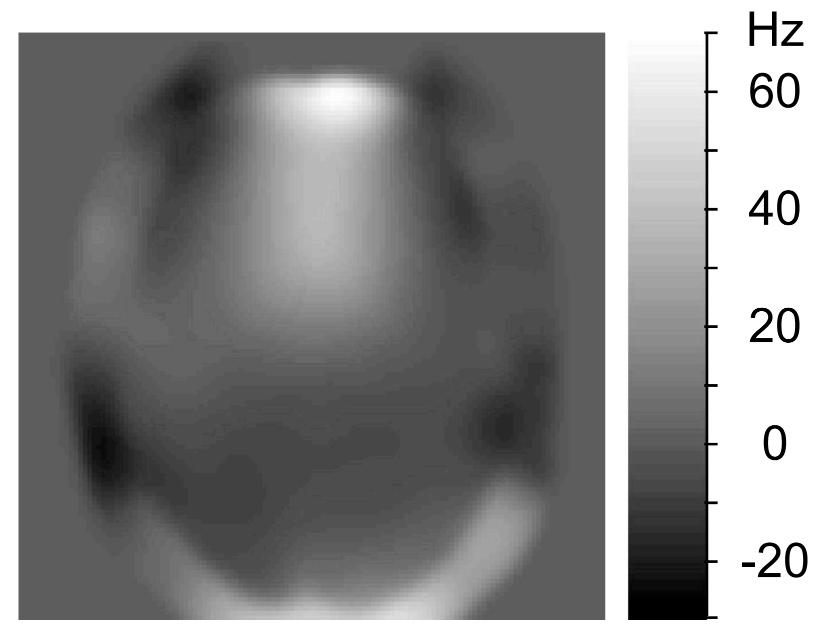
So, let's relate this to the numbers in the off-resonance map.

What do you think an image of this object will look like given this field and these imaging parameters?

# Off-resonance field ⇒ Distortions



BW/pixel ~10Hz

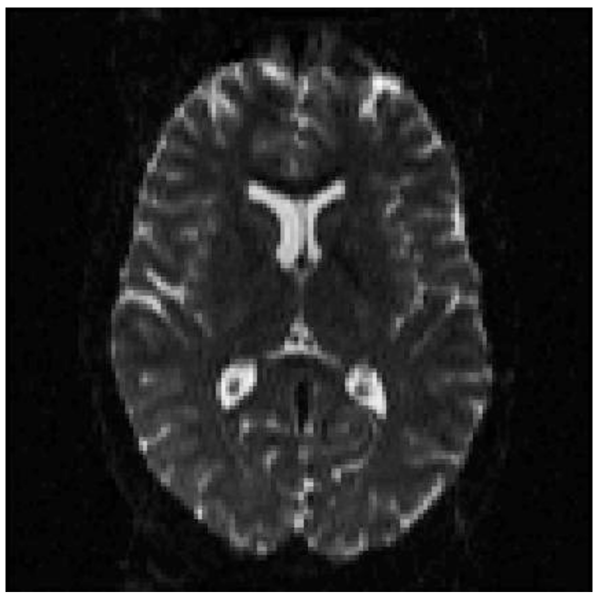



BW/pixel ~1000Hz

So, let's relate this to the numbers in the off-resonance map.

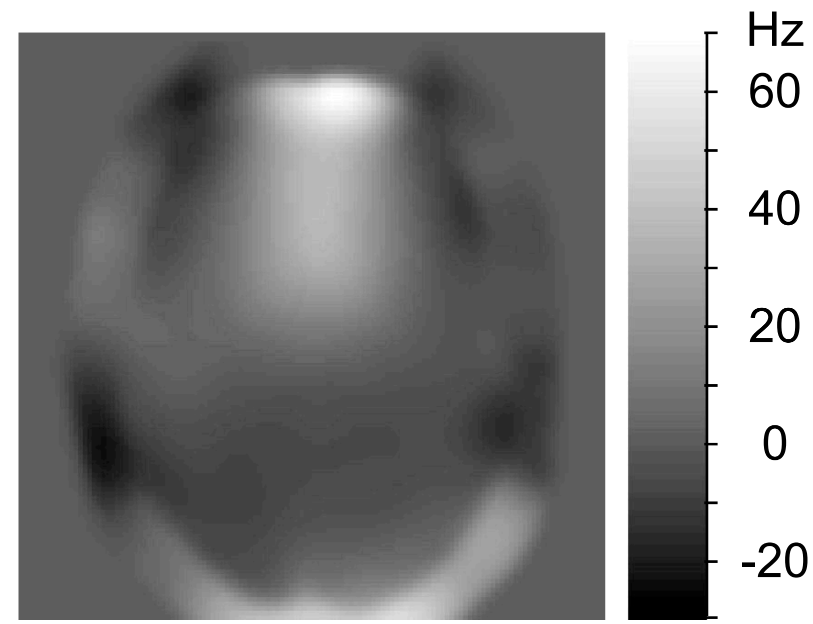
What do you think an image of this object will look like given this field and these imaging parameters?

# Off-resonance field ⇒ Distortions



BW/pixel ~10Hz

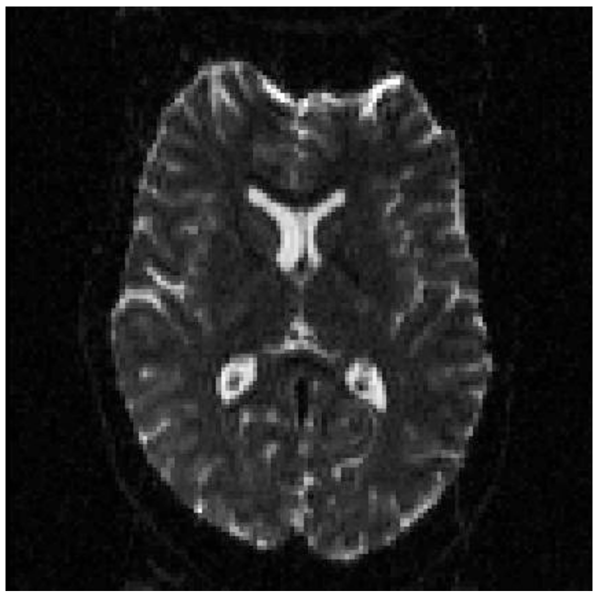
BW/pixel ~1000Hz




So, let’s relate this to the numbers in the off-resonance map.

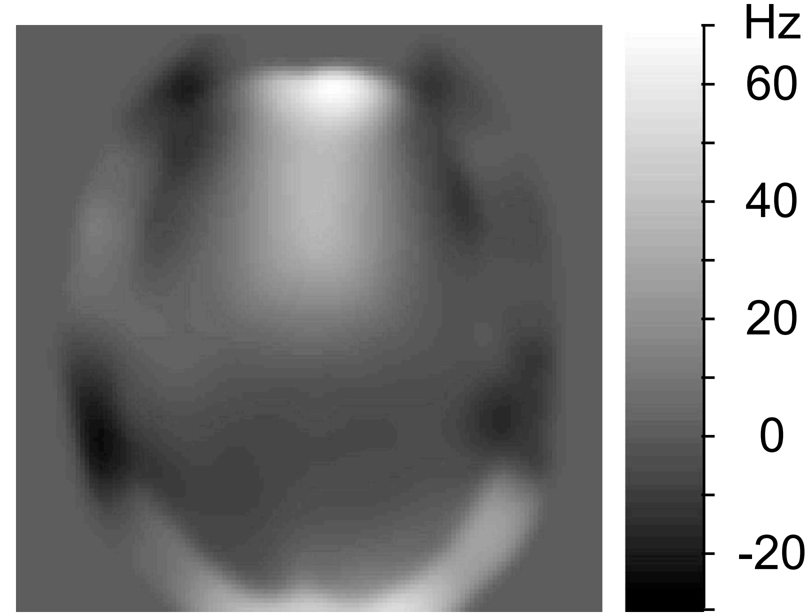
And what about this case?

# Off-resonance field ⇒ Distortions



BW/pixel ~10Hz

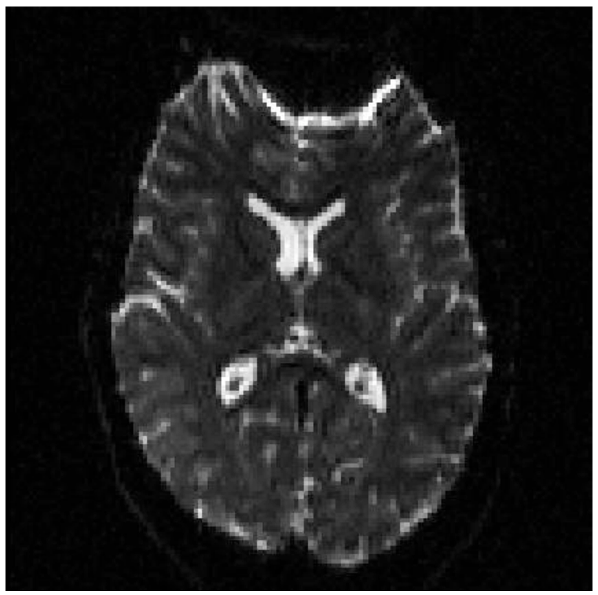
BW/pixel ~1000Hz



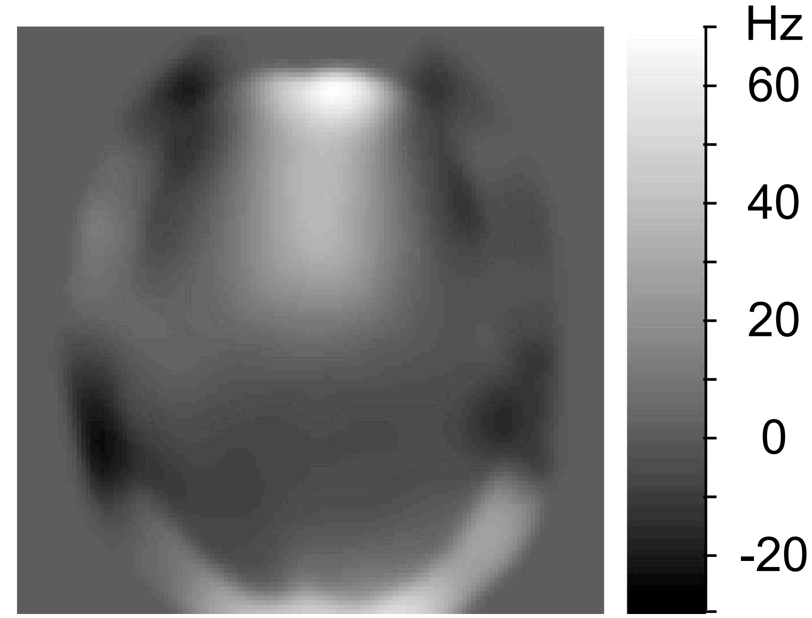

So, let’s relate this to the numbers in the off-resonance map.

And what about this case?

# Off-resonance field ⇒ Distortions



BW/pixel ~7Hz

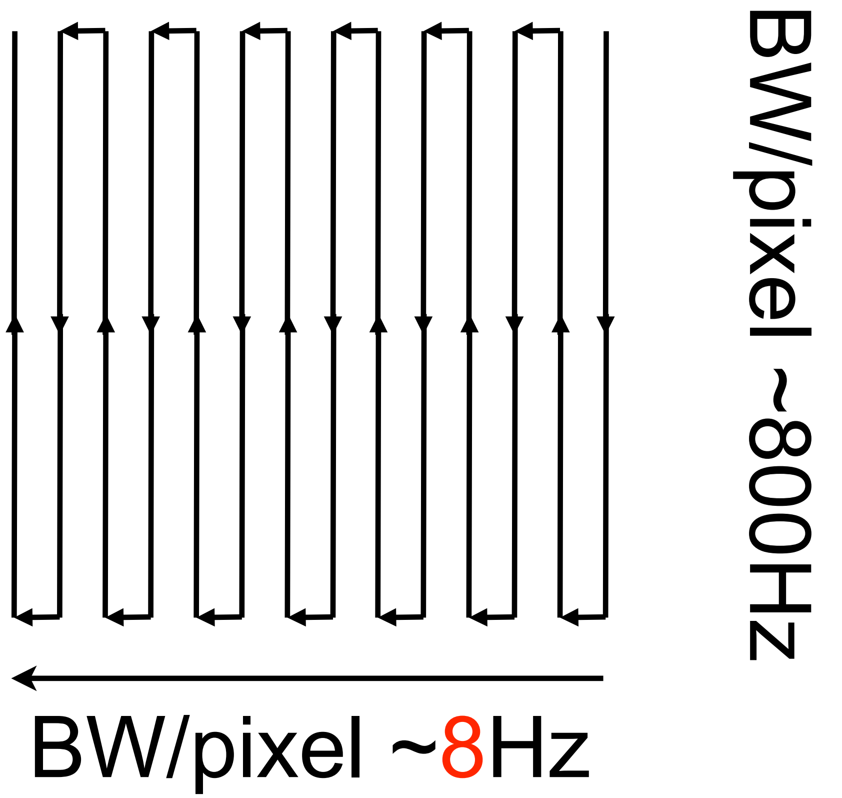


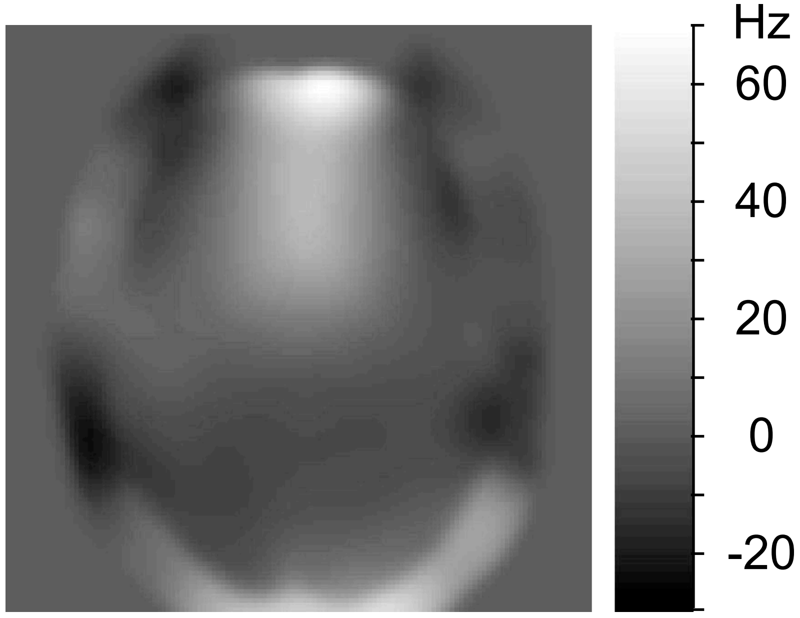

BW/pixel ~700Hz

So, let’s relate this to the numbers in the off-resonance map.

And if we change the BandWidth?

# Off-resonance field ⇒ Distortions

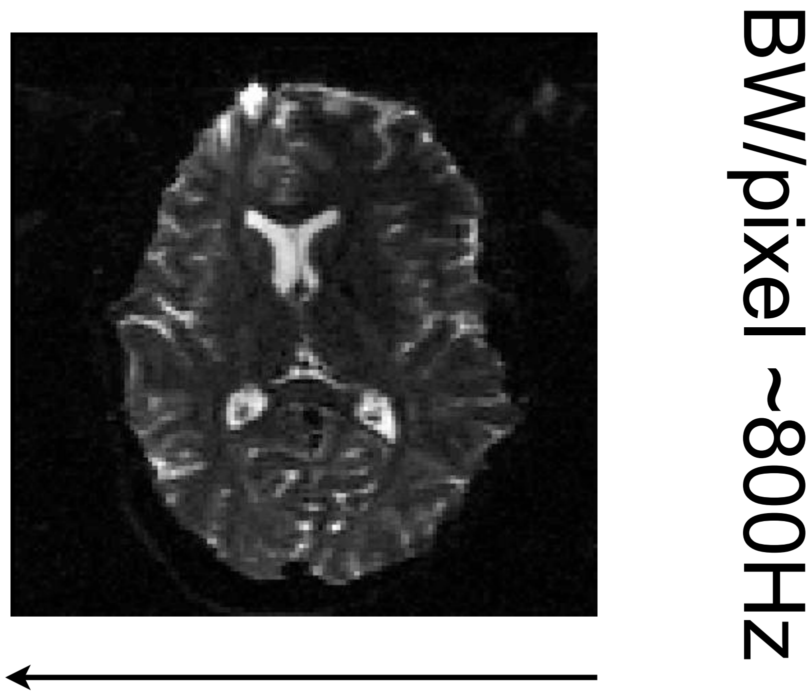


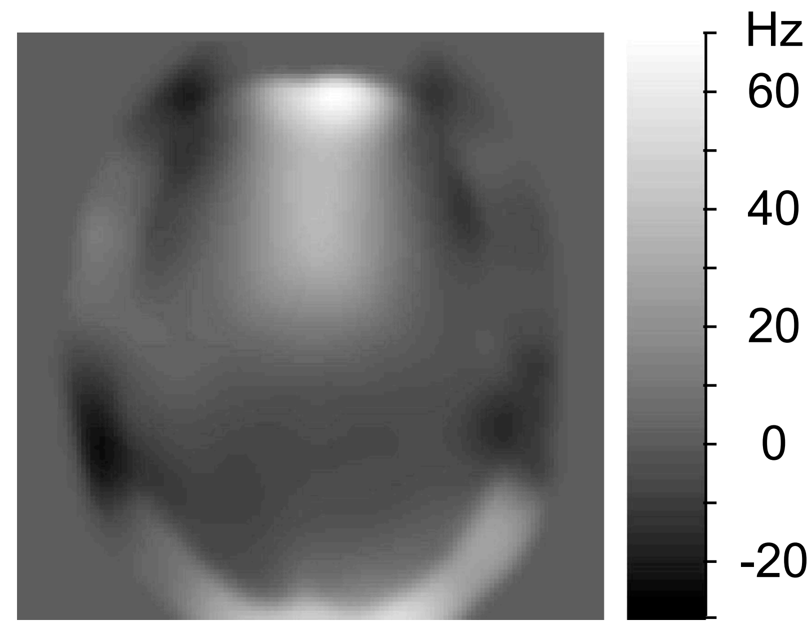




So, let's relate this to the numbers in the off-resonance map.

And if we change the direction of the phase-encoding?

# Off-resonance field ⇒ Distortions

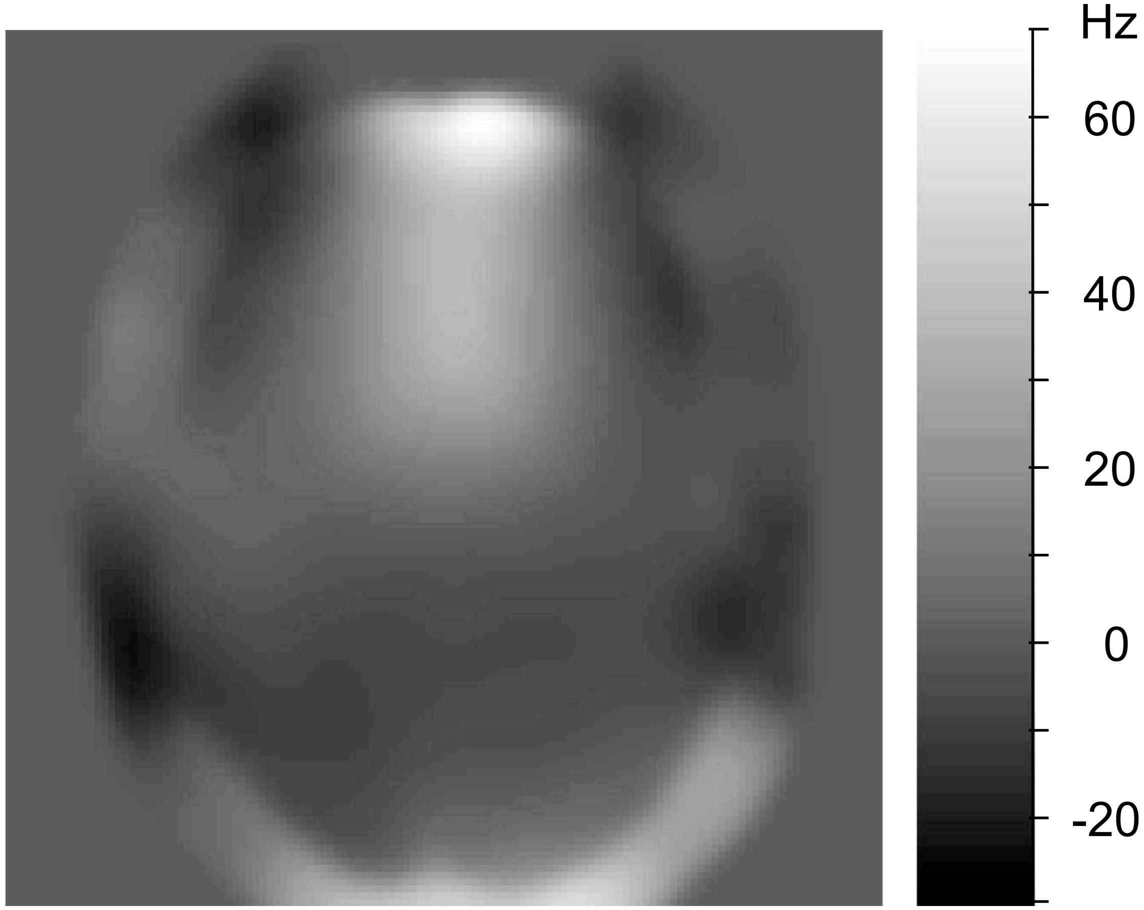



So, let's relate this to the numbers in the off-resonance map.

And if we change the direction of the phase-encoding?

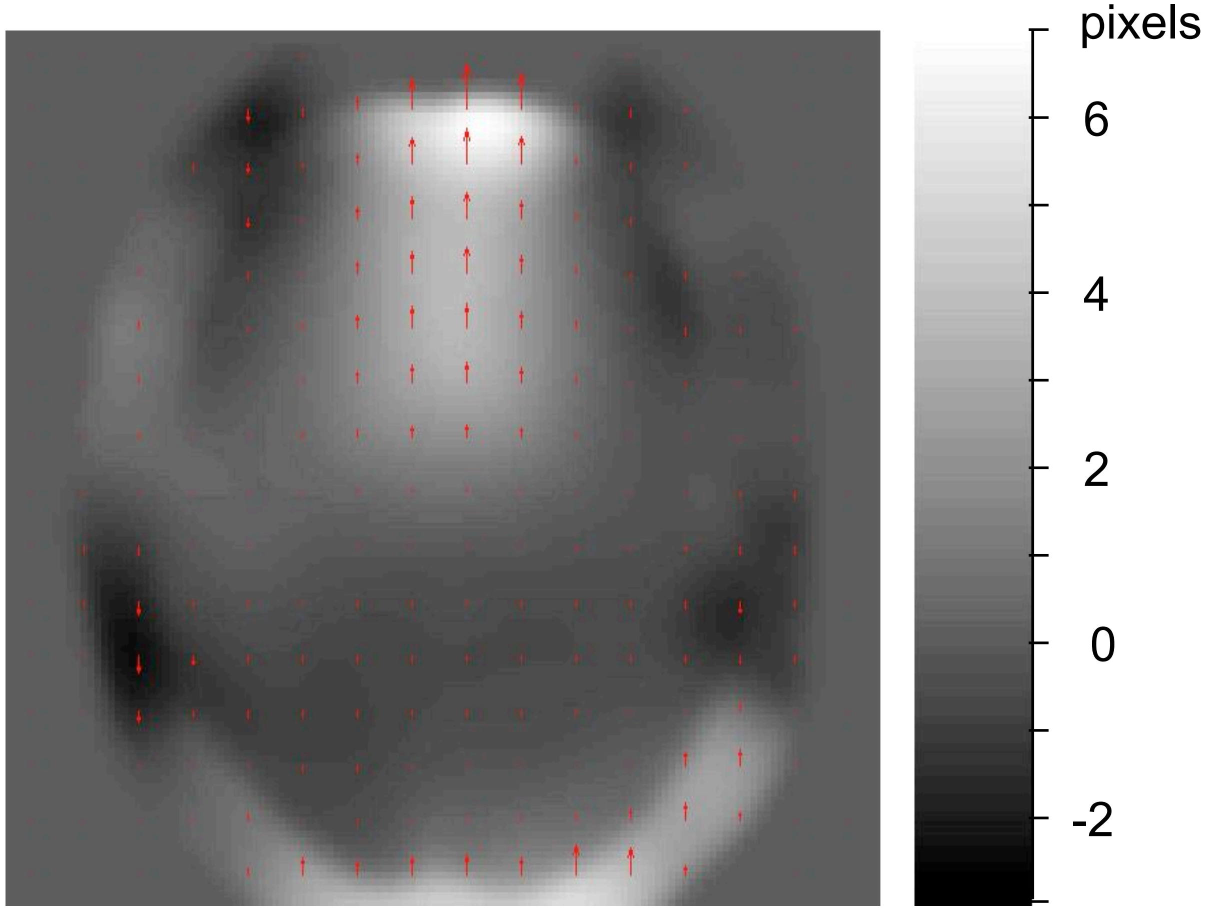
# Off-resonance field ⇒ Distortions




So, an off-resonance field is effectively a scaled voxel-displacement map.

And if we know the imaging parameters we can do the translation.

# Off-resonance field ⇒ Distortions

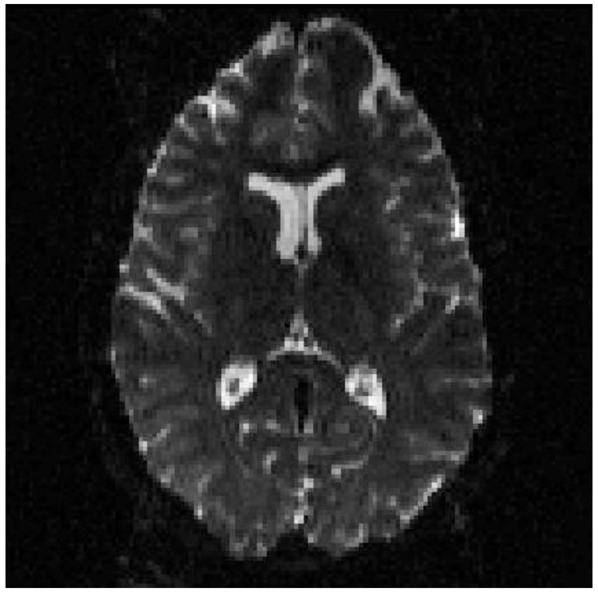



So, an off-resonance field is effectively a scaled voxel-displacement map.

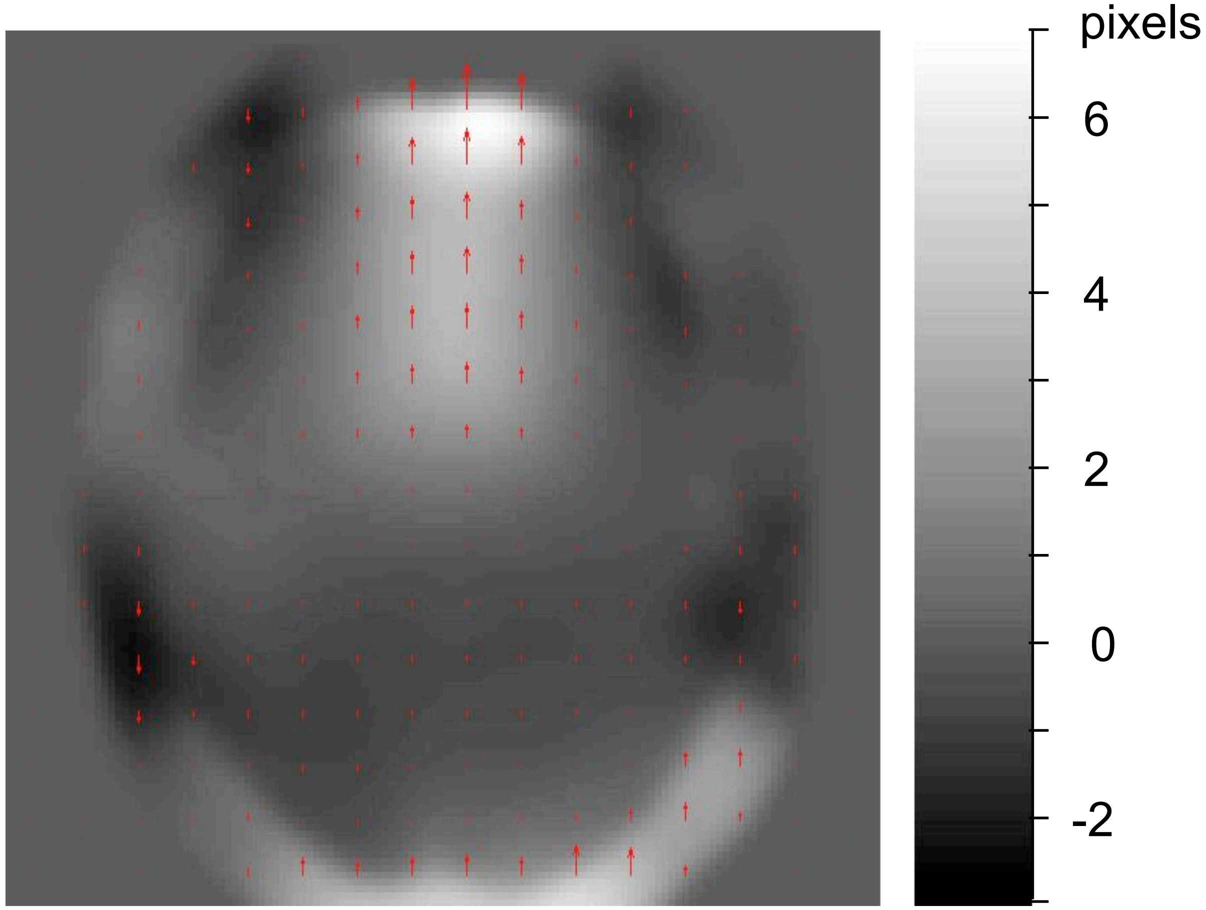
And if we know the imaging parameters we can do the translation.

BW/pixel = 10Hz, **p** = [0 1 0]

# Off-resonance field ⇒ Distortions



And know what to expect

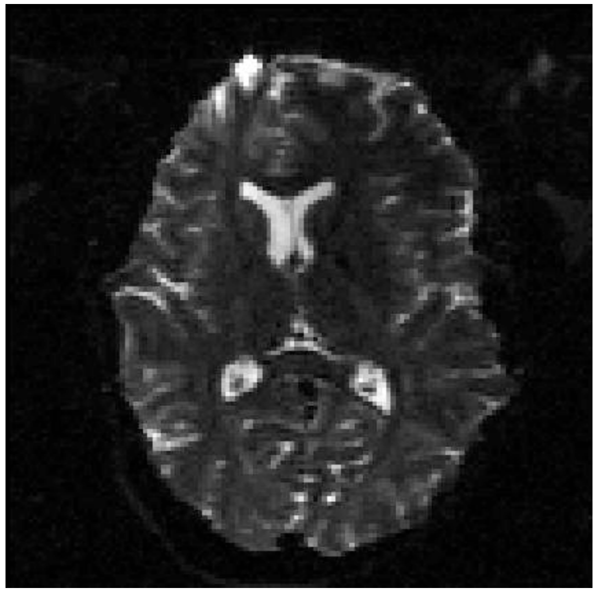


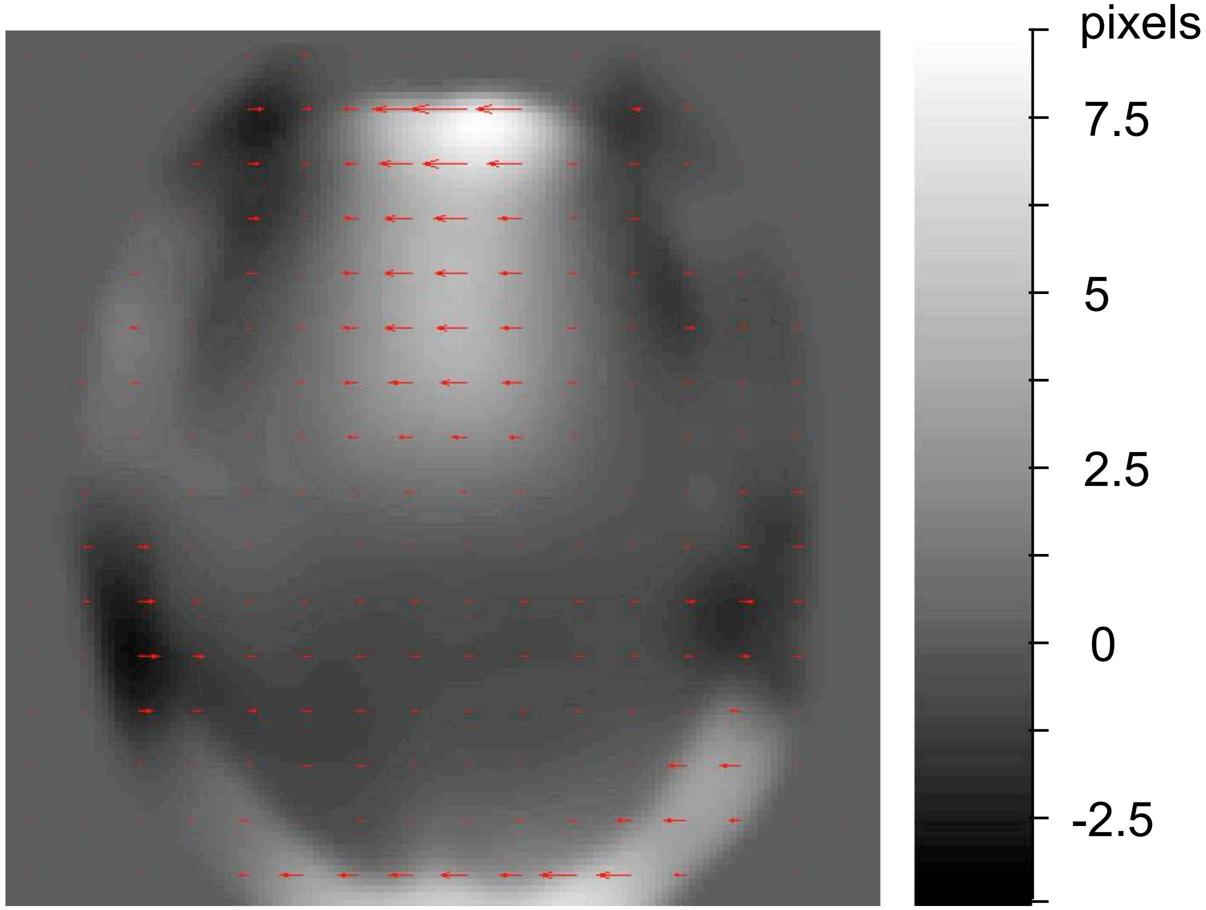

So, an off-resonance field is effectively a scaled voxel-displacement map.

And if we know the imaging parameters we can do the translation.

BW/pixel = 10Hz, **p** = [0 1 0]

# Off-resonance field ⇒ Distortions






And know what to expect

So, an off-resonance field is effectively a scaled voxel-displacement map.

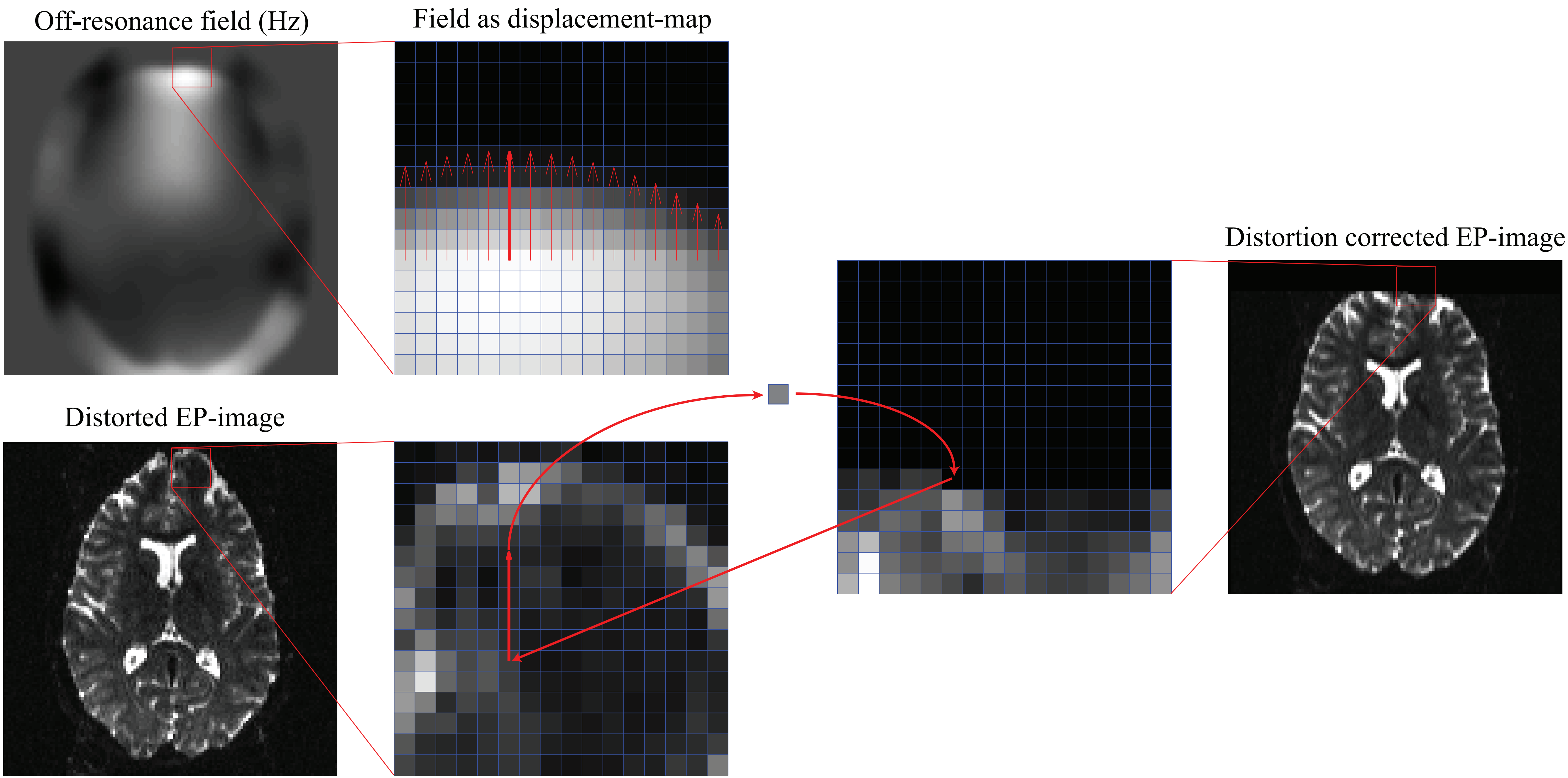
And if we know the imaging parameters we can do the translation.

BW/pixel = 8Hz, **p** = [-1 0 0]

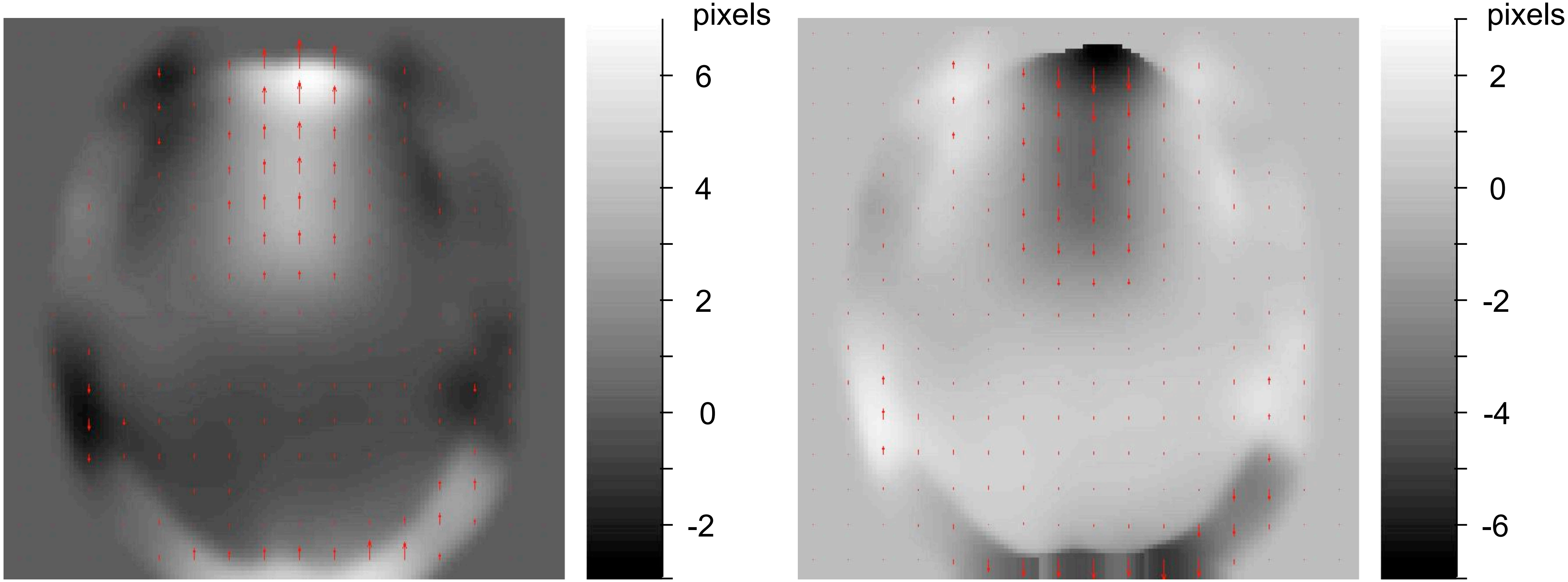
# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# How to “fix” a distorted image



# How to “mess up” a good image

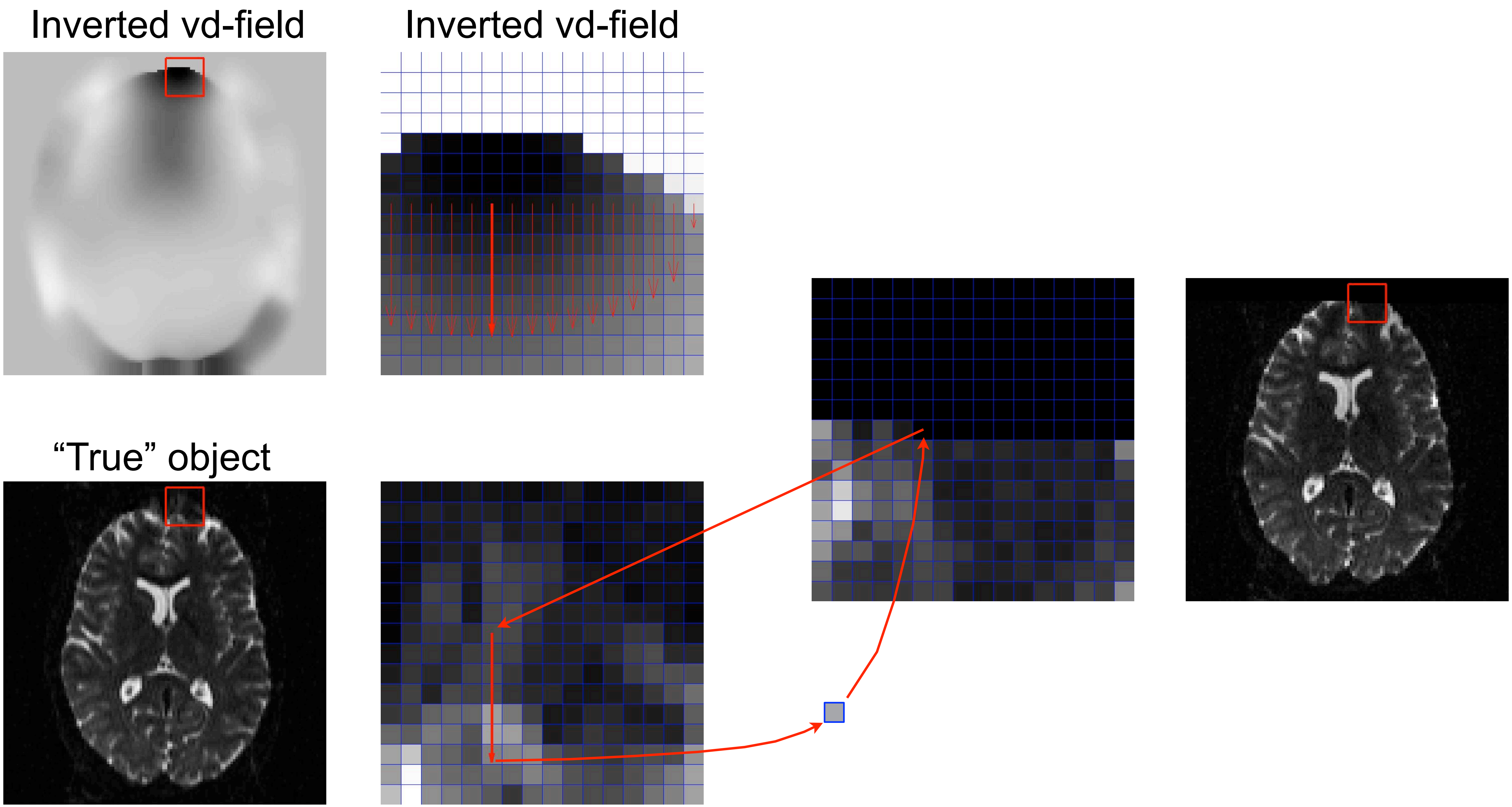



For that we need the inverse displacement map.

The inverse map is that which “undoes what the forward map does”.

N.B. the inverse map is NOT the negation of the forward map.

# How to "mess up" a good image




Then we do just like before.

# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# Where does the off-resonance field come from?

- There are two sources
- The first is the object (head) itself.

(CT of) Human head

$B_0 \odot$

Resulting field

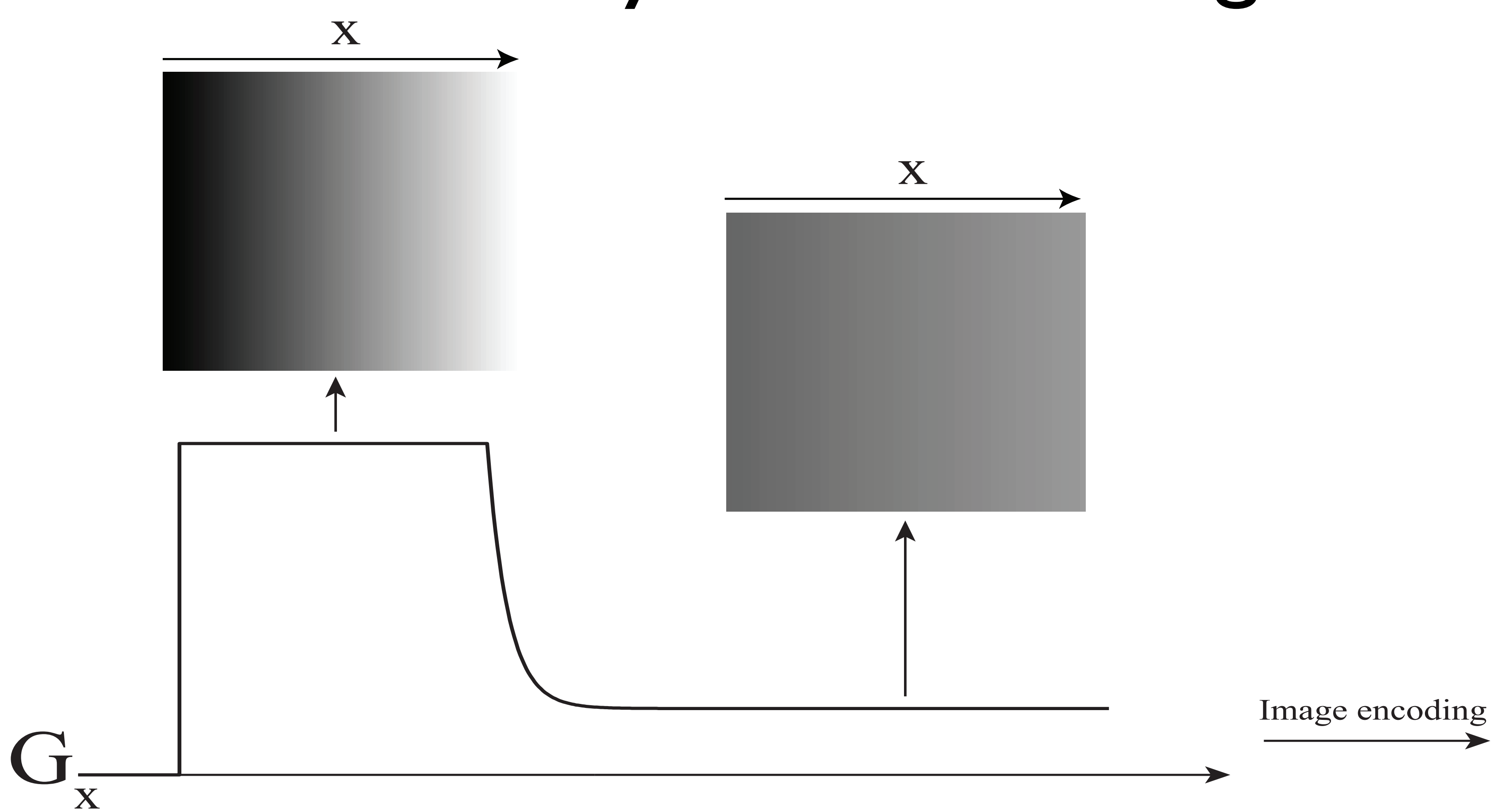
PPMs

$$\text{Must fulfil} \begin{cases} \nabla \times \mathbf{H} = 0 \\ \nabla \bullet \mathbf{B} = 0 \end{cases} \quad \text{(still)}$$

# Where does the off-resonance field come from?

- There are two sources
- The first is the object (head) itself.

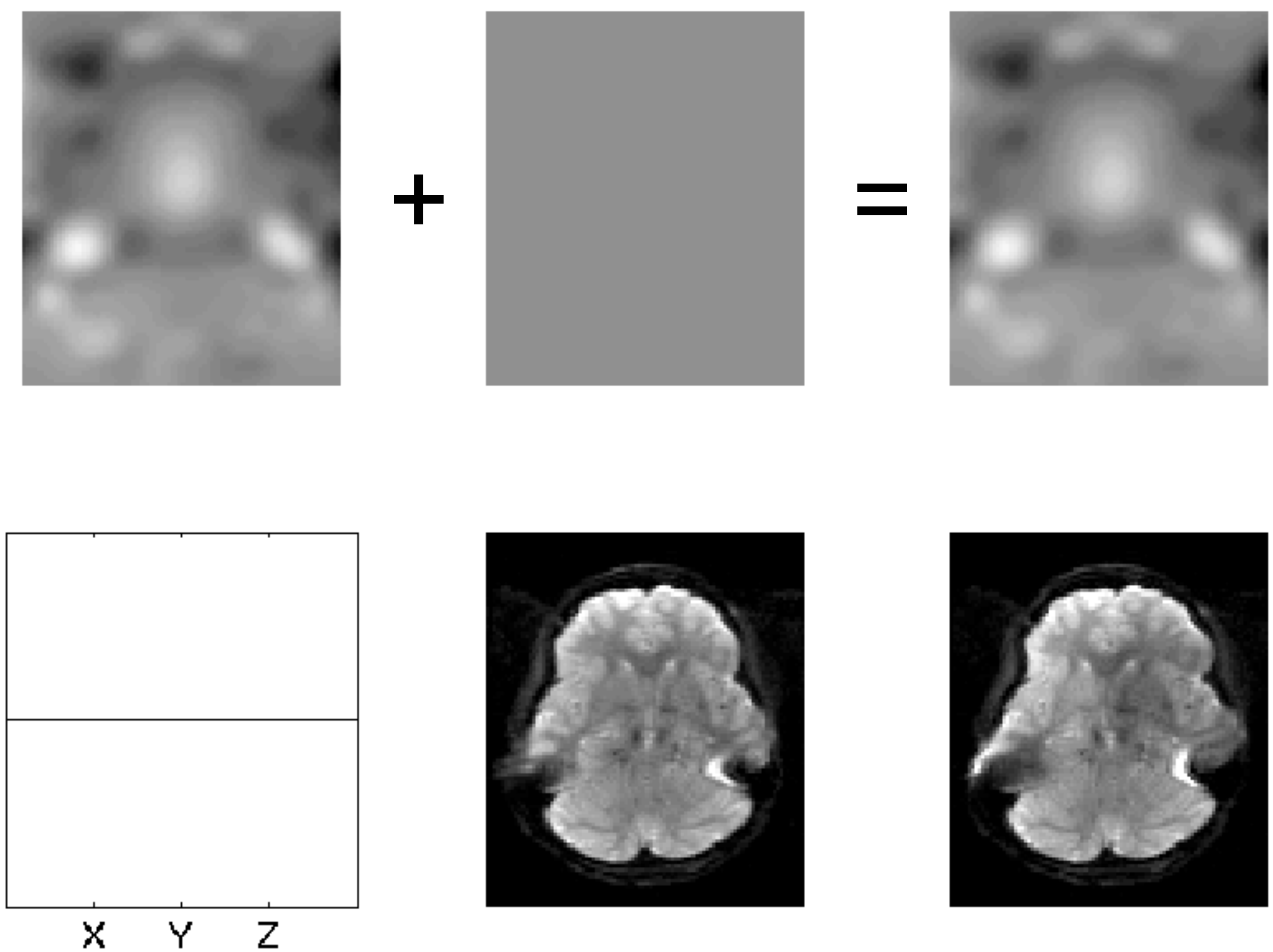
- The second is caused by the diffusion gradient



# Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions

Susceptibility | Eddy currents | Total

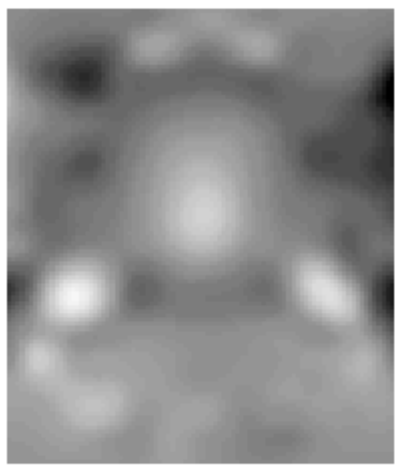



Diffusion gradient | “True” object | Observed image

# Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions

Susceptibility

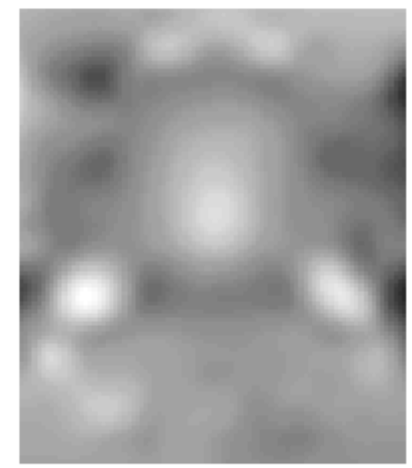


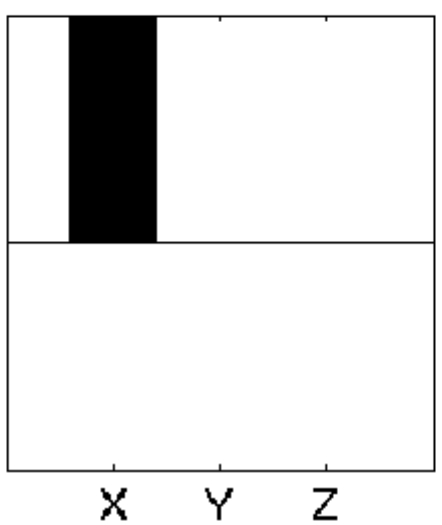
+

Eddy currents

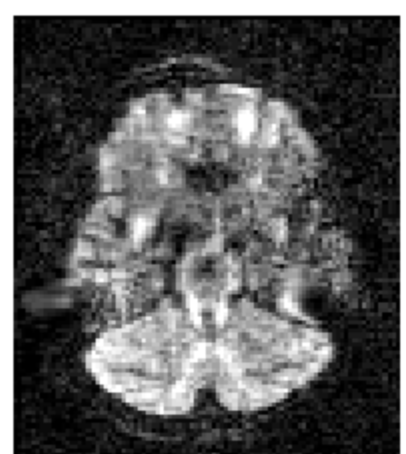
=

Total






Diffusion gradient

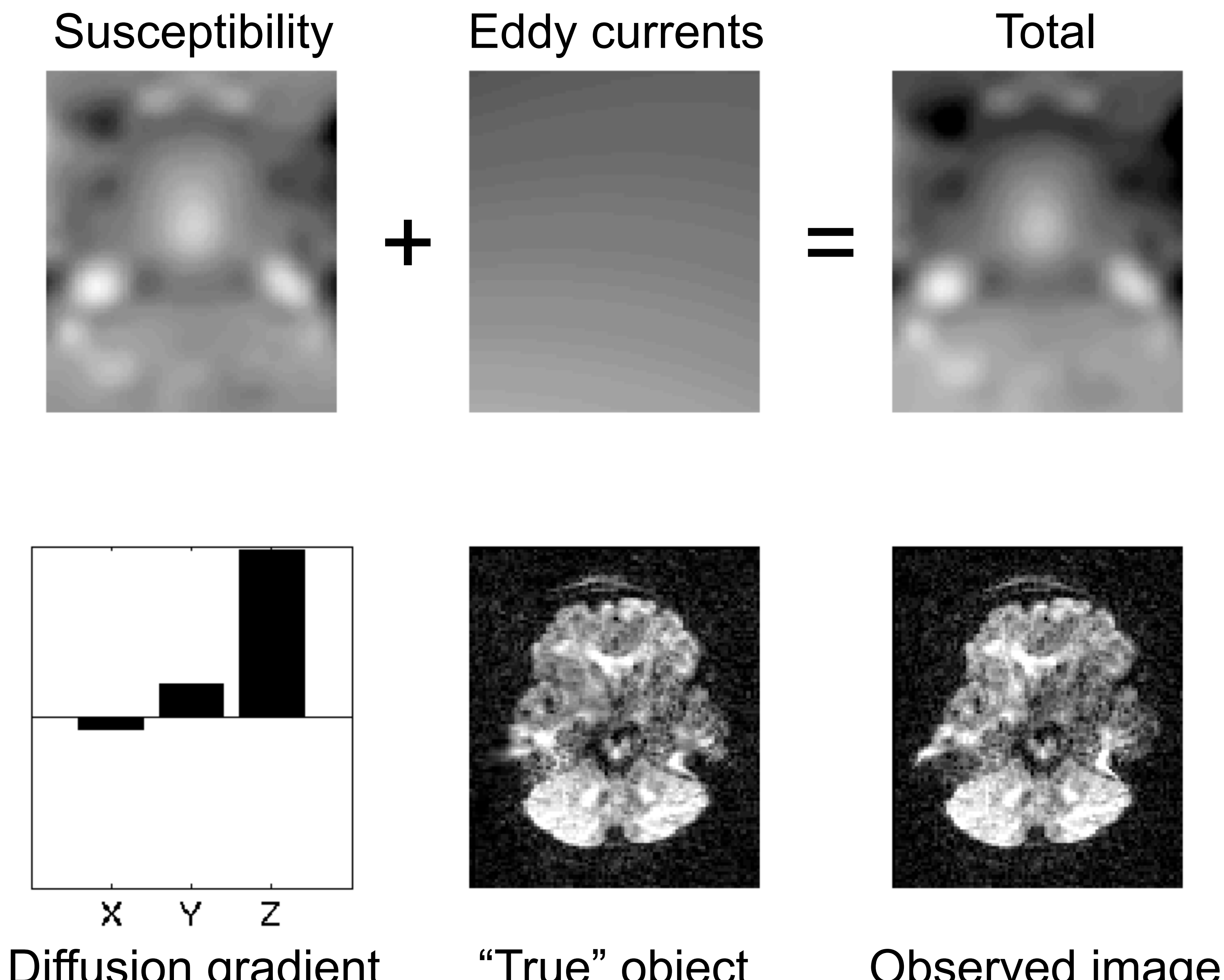


“True” object

Observed image

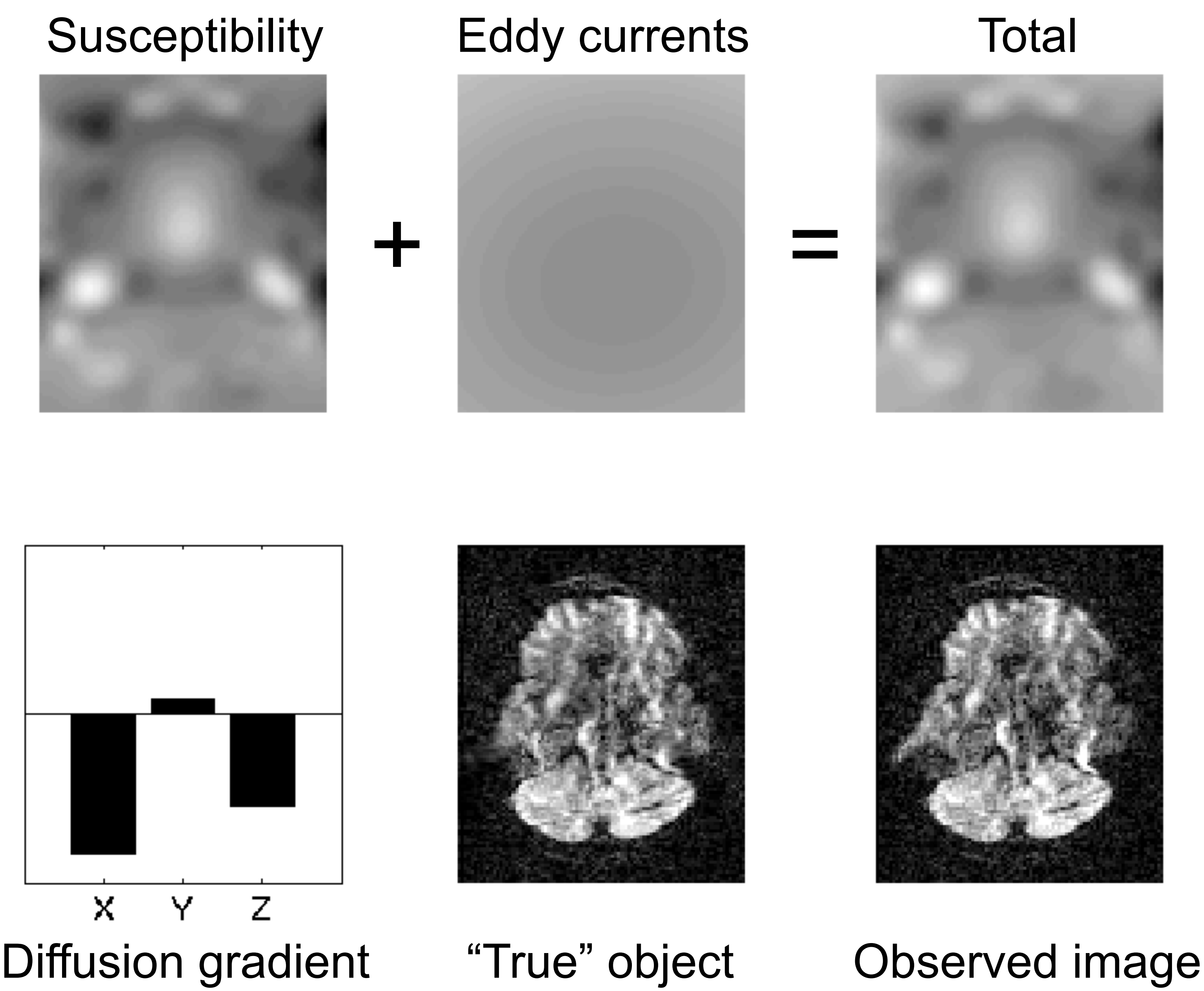
# Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



# Where does the off-resonance field come from?

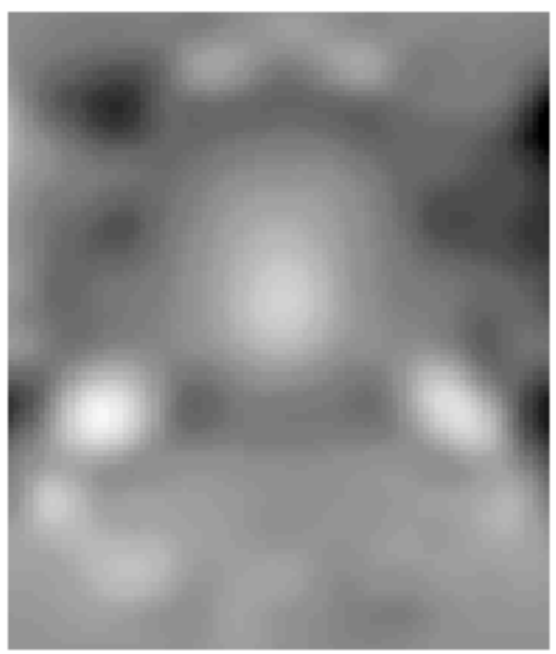
So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



# Separate estimation of susceptibility- and eddy current-fields

So, what we need to estimate is

One of these per subject



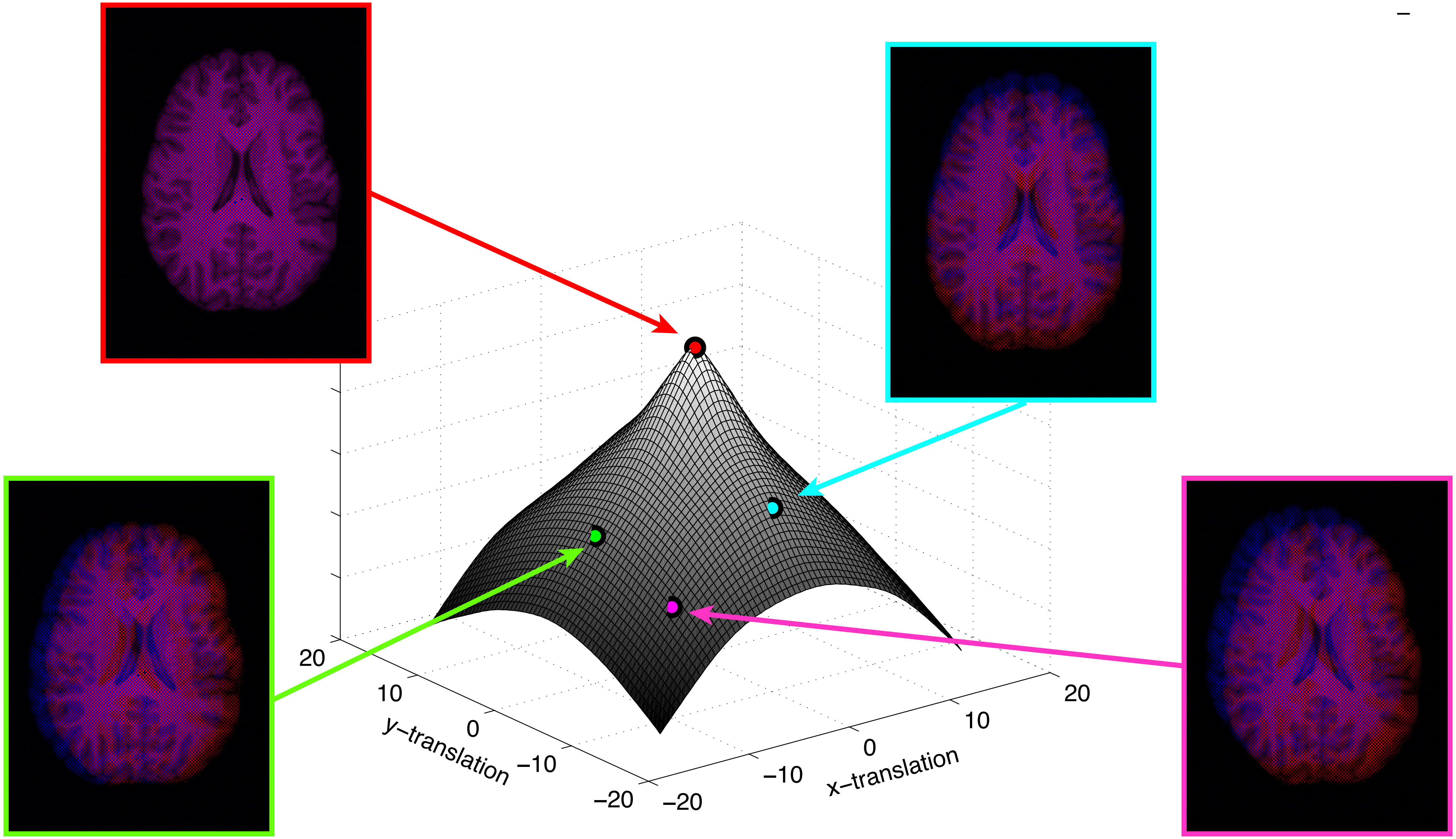
One of these per volume

FSL-tools: topup eddy

# Outline of the talk

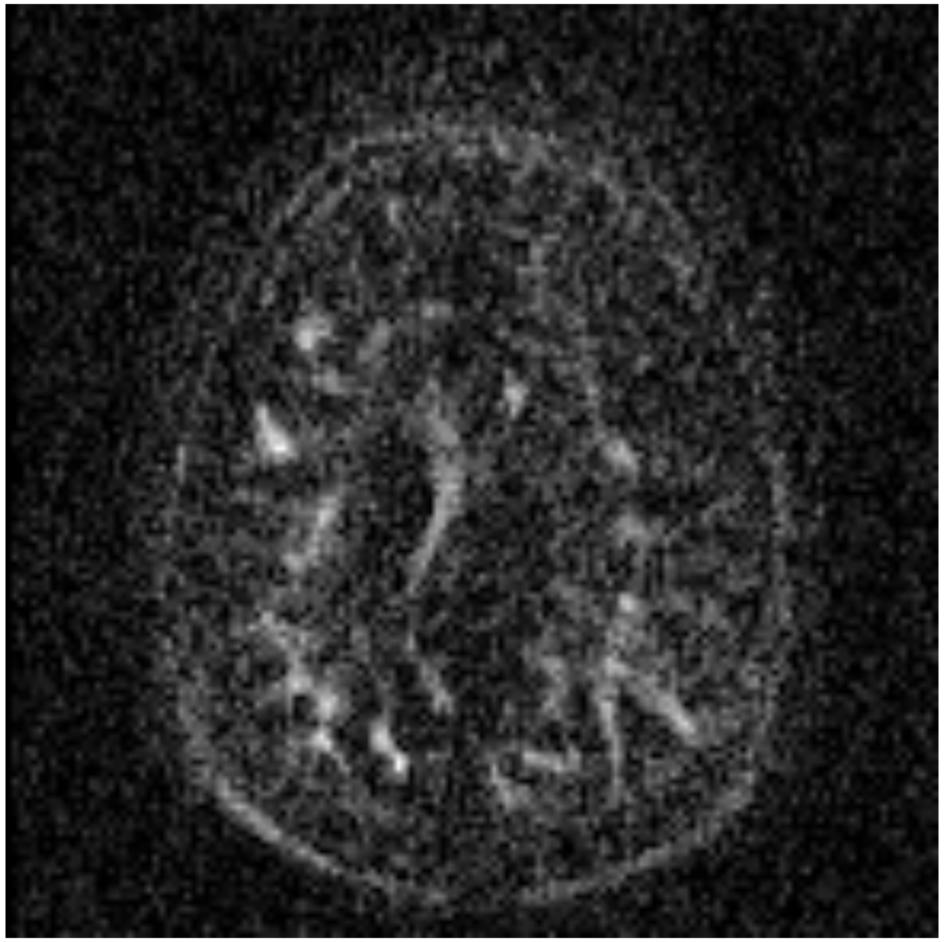
- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# Worlds shortest course on image registration

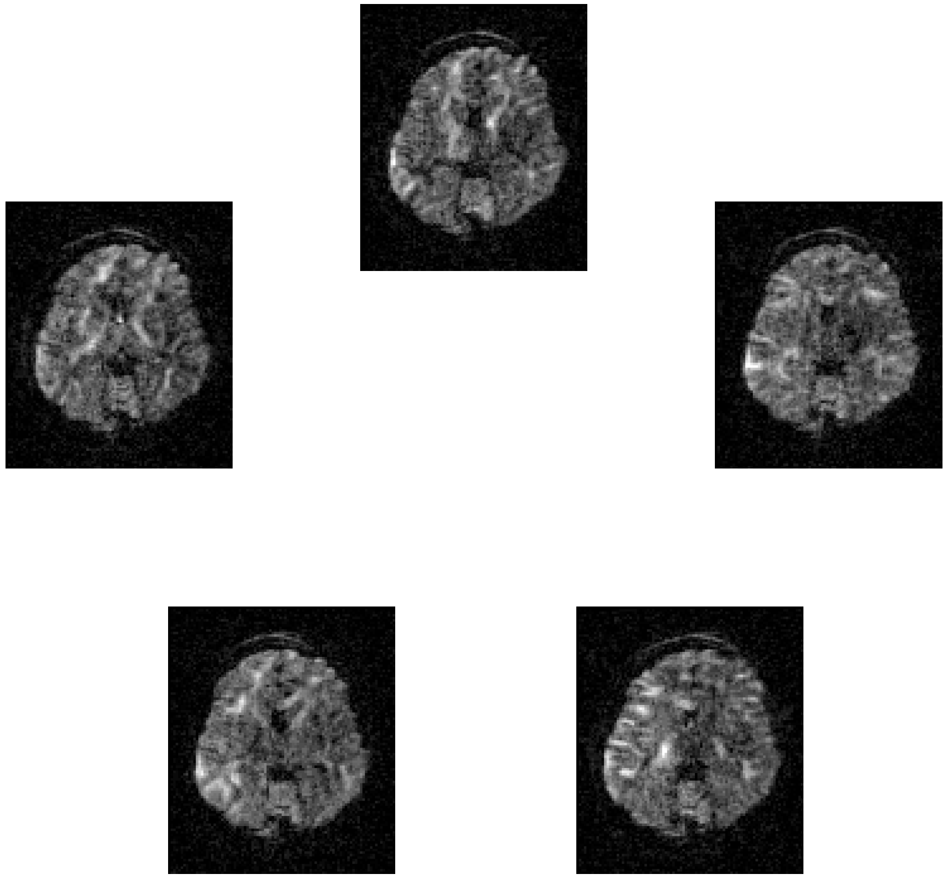



Maximising/minimising an objective/cost-function

# But it is not easy to register diffusion weighted images



The different diffusion weighted images have different contrast.

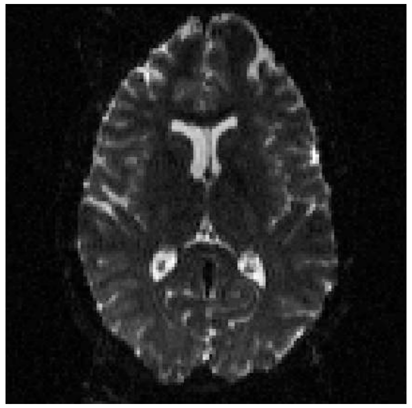


All the images are distorted, only differently. How do we know the truth?

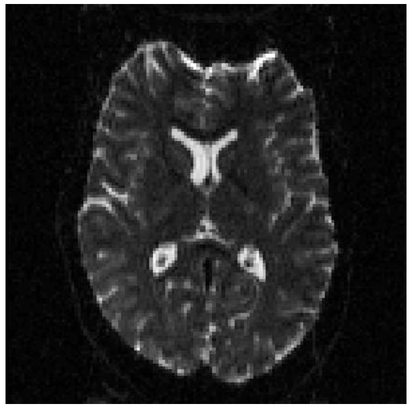
# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# How topup works (very briefly)



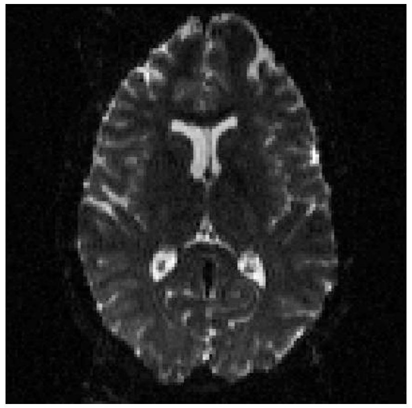
**p**=[0 1 0]



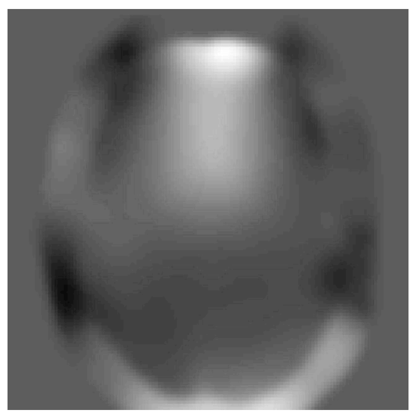
**p**=[0 -1 0]

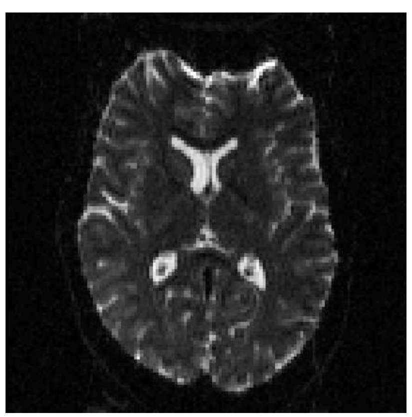
If we have two images acquired with different phase-encoding

# How topup works (very briefly)



**p**=[0 1 0]

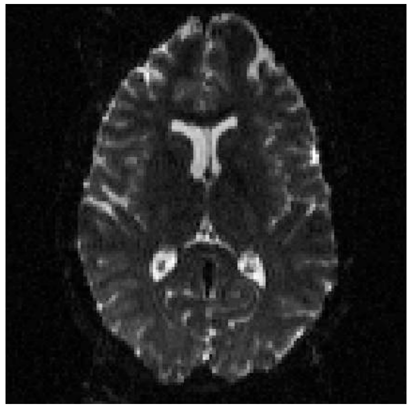




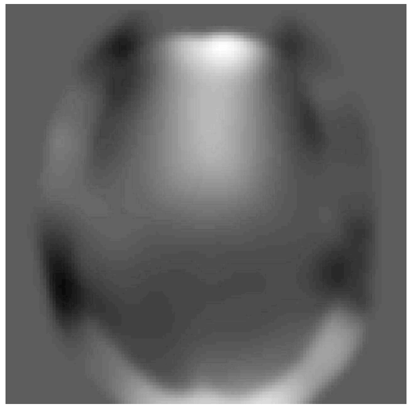
**p**=[0 -1 0]

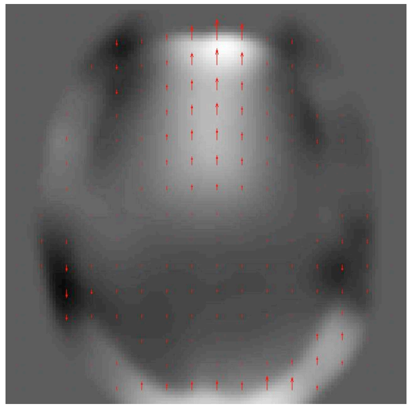
And we know what the off-resonance field is

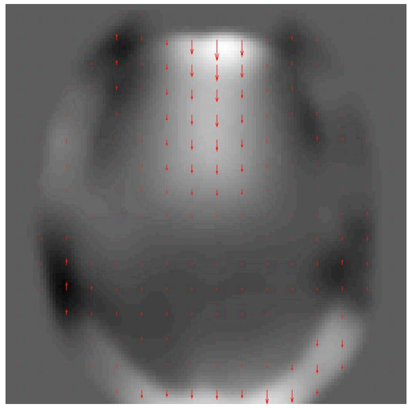
# How topup works (very briefly)

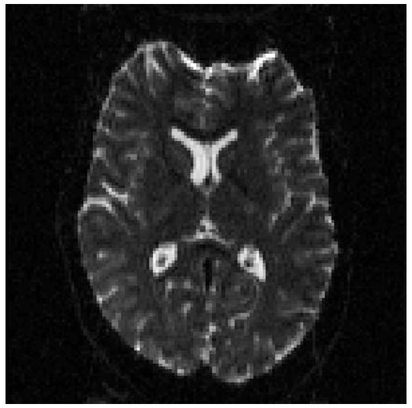


p=[0 1 0]





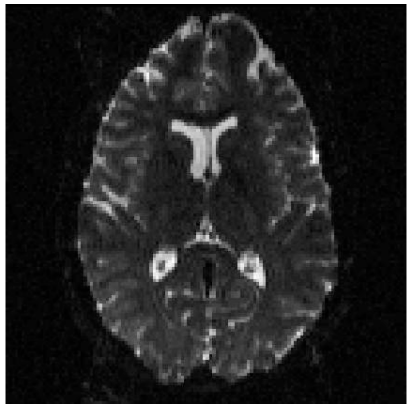




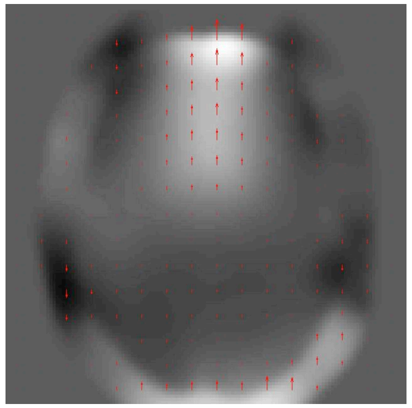
p=[0 -1 0]

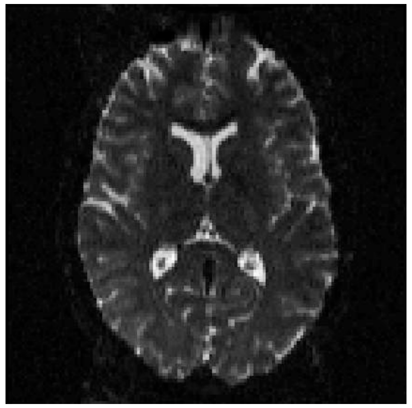
We can combine this with the PE information to get displacement maps

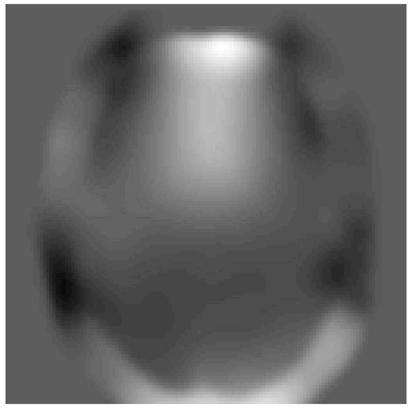
# How topup works (very briefly)

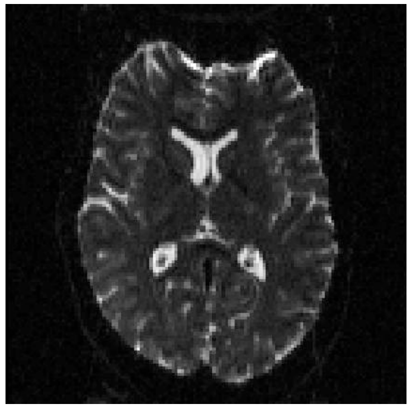


**p**=[0 1 0]

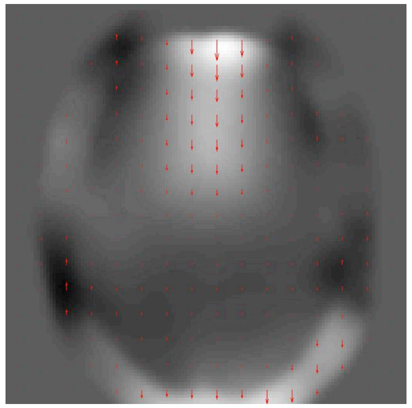


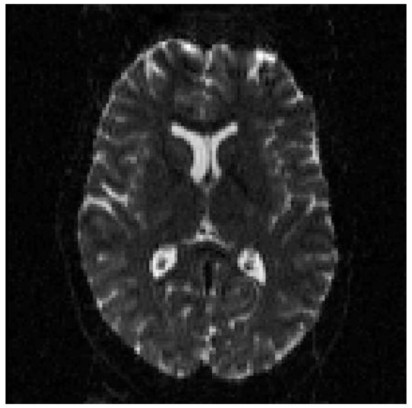






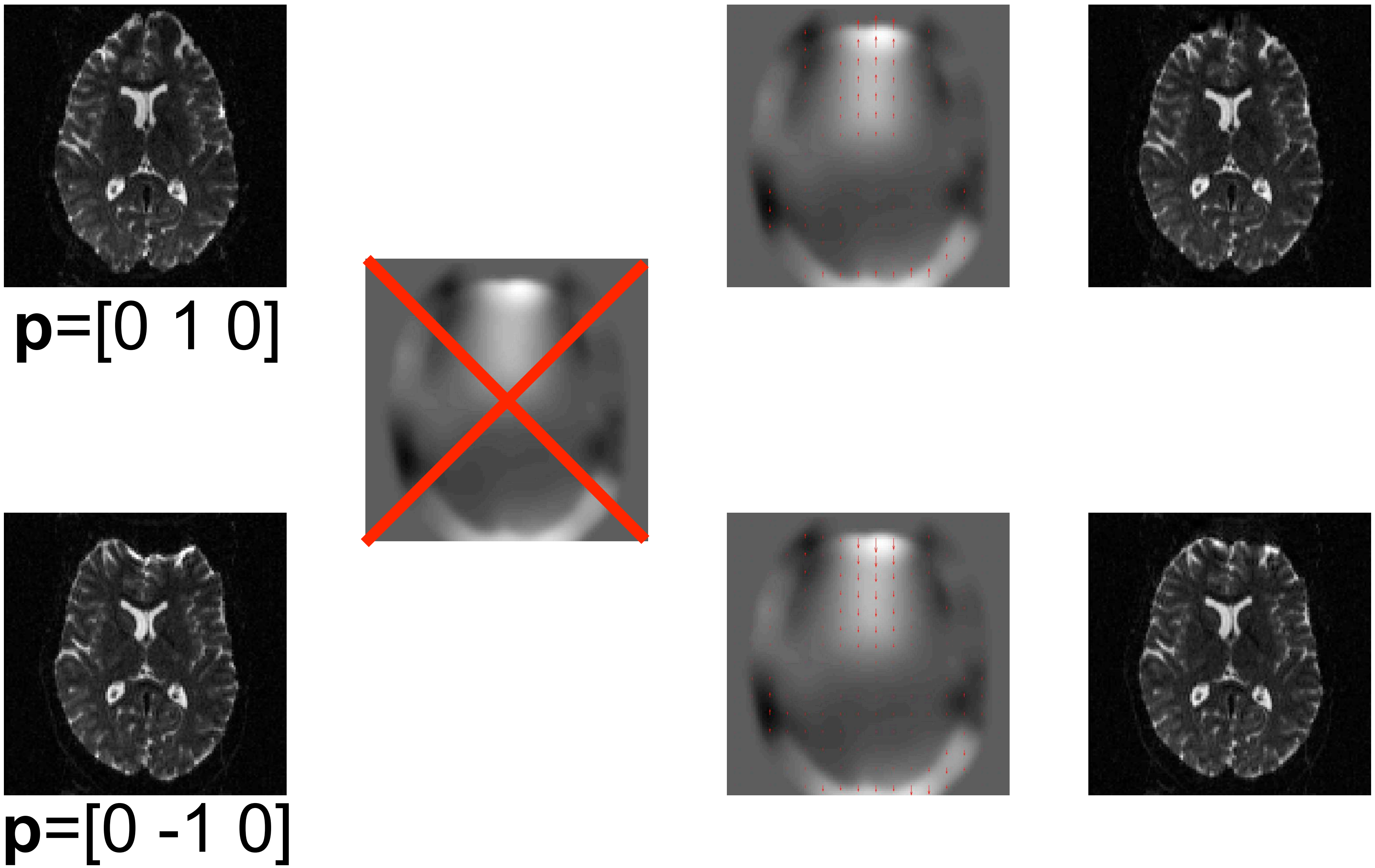
**p**=[0 -1 0]





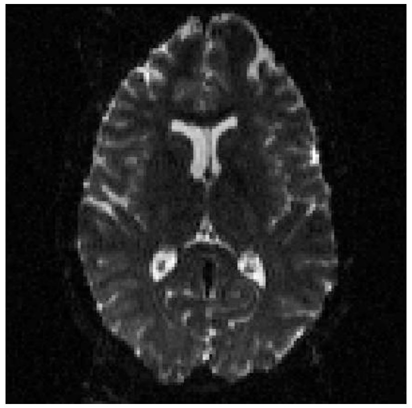
And use that to correct the distortions

# How topup works (very briefly)

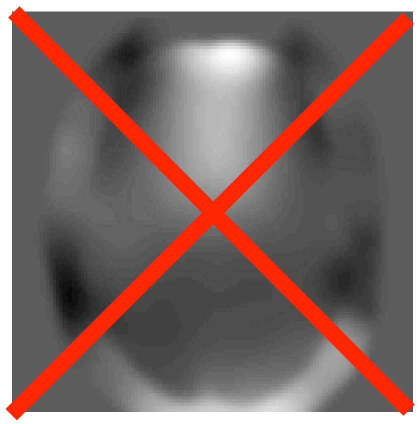


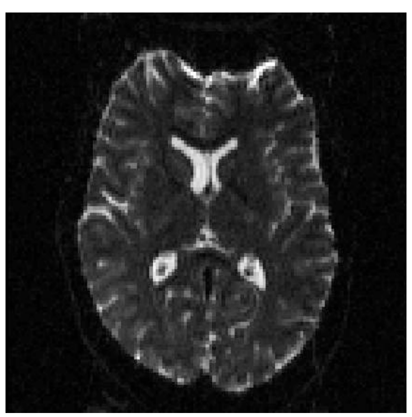

BUT we don't know the field. That is what we want topup to calculate.

# How topup works (very briefly)



**p**=[0 1 0]

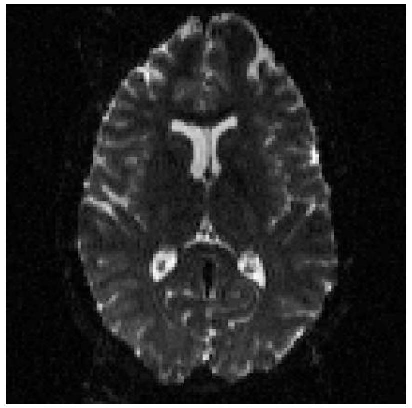




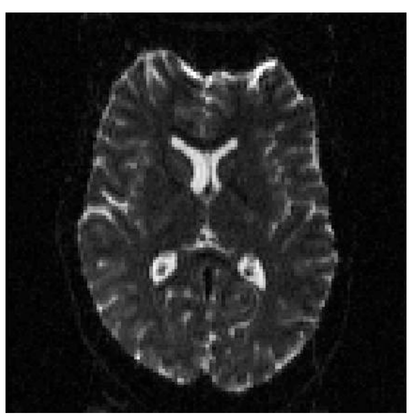
**p**=[0 -1 0]

BUT we don’t know the field. That is what we want topup to calculate.

# How topup works (very briefly)



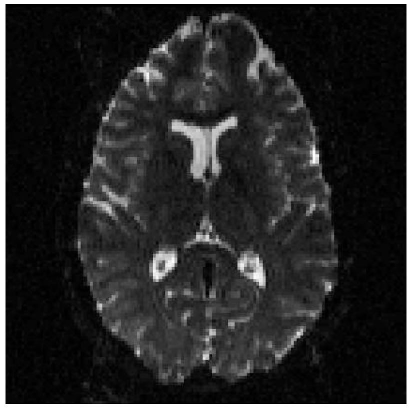
**p**=[0 1 0]



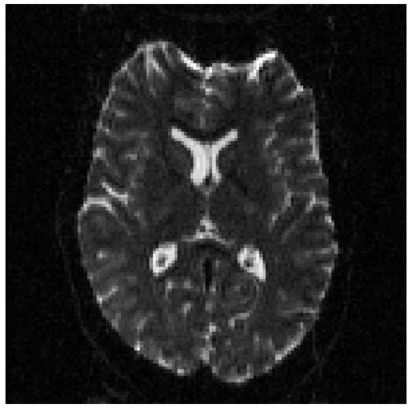
**p**=[0 -1 0]

So, topup “guesses” a field...

# How topup works (very briefly)



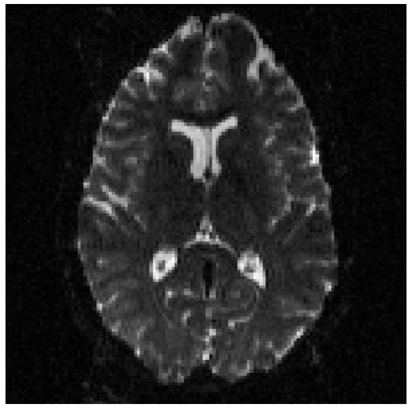
**p**=[0 1 0]



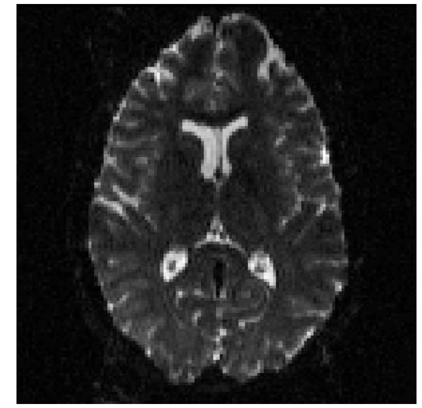
**p**=[0 -1 0]

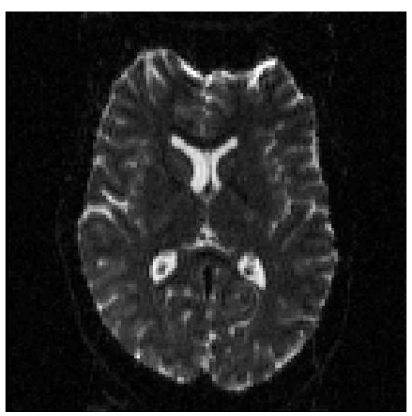
...calculates the displacement maps...

# How topup works (very briefly)

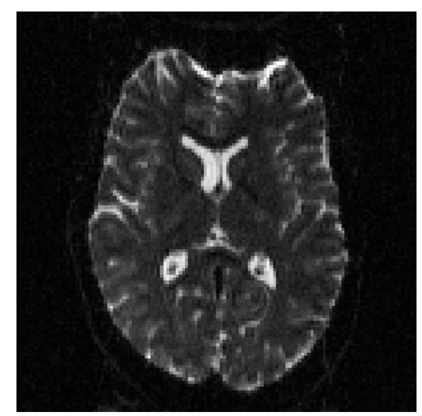


**p**=[0 1 0]



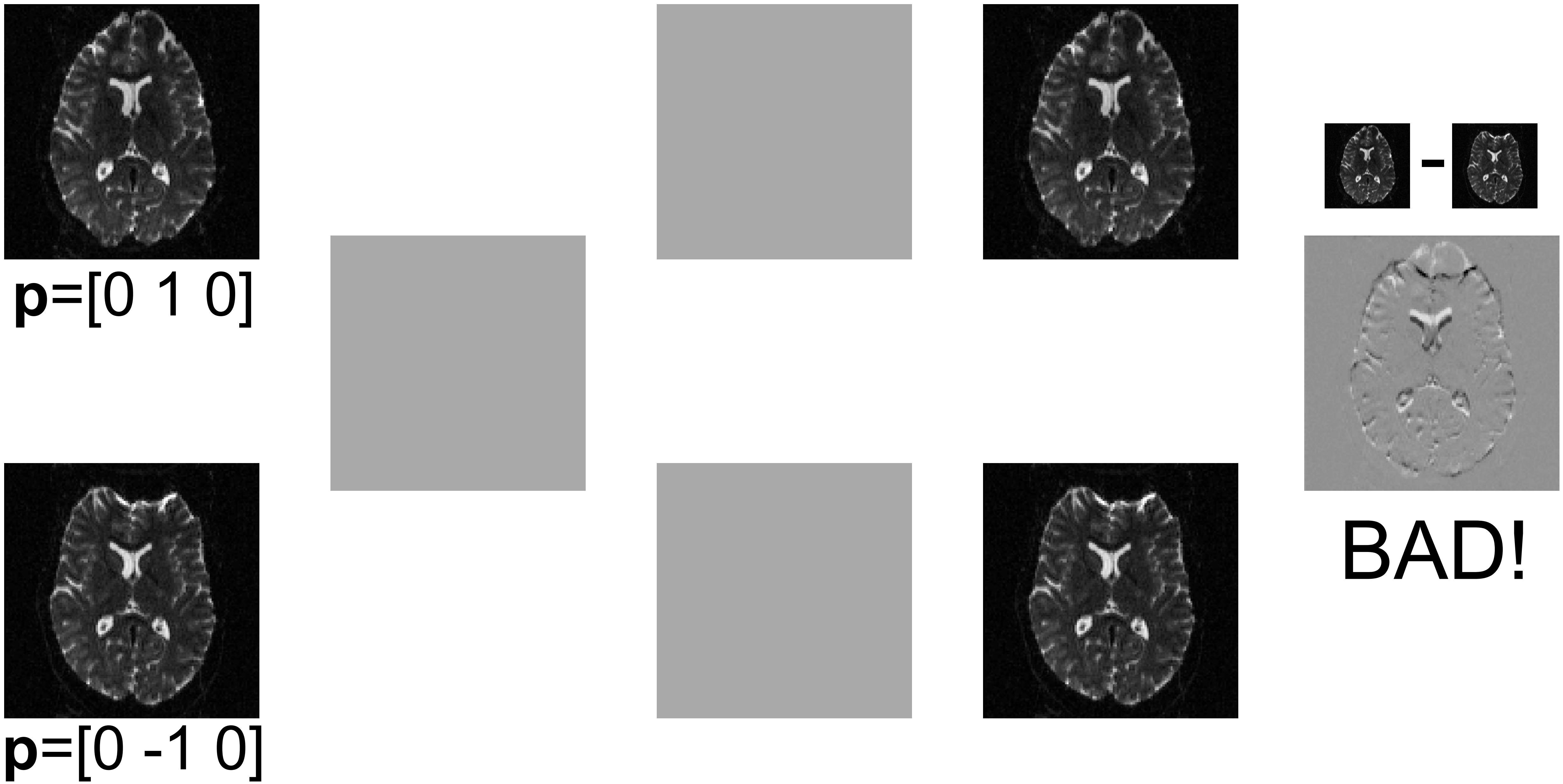


**p**=[0 -1 0]



..."corrects" the images...

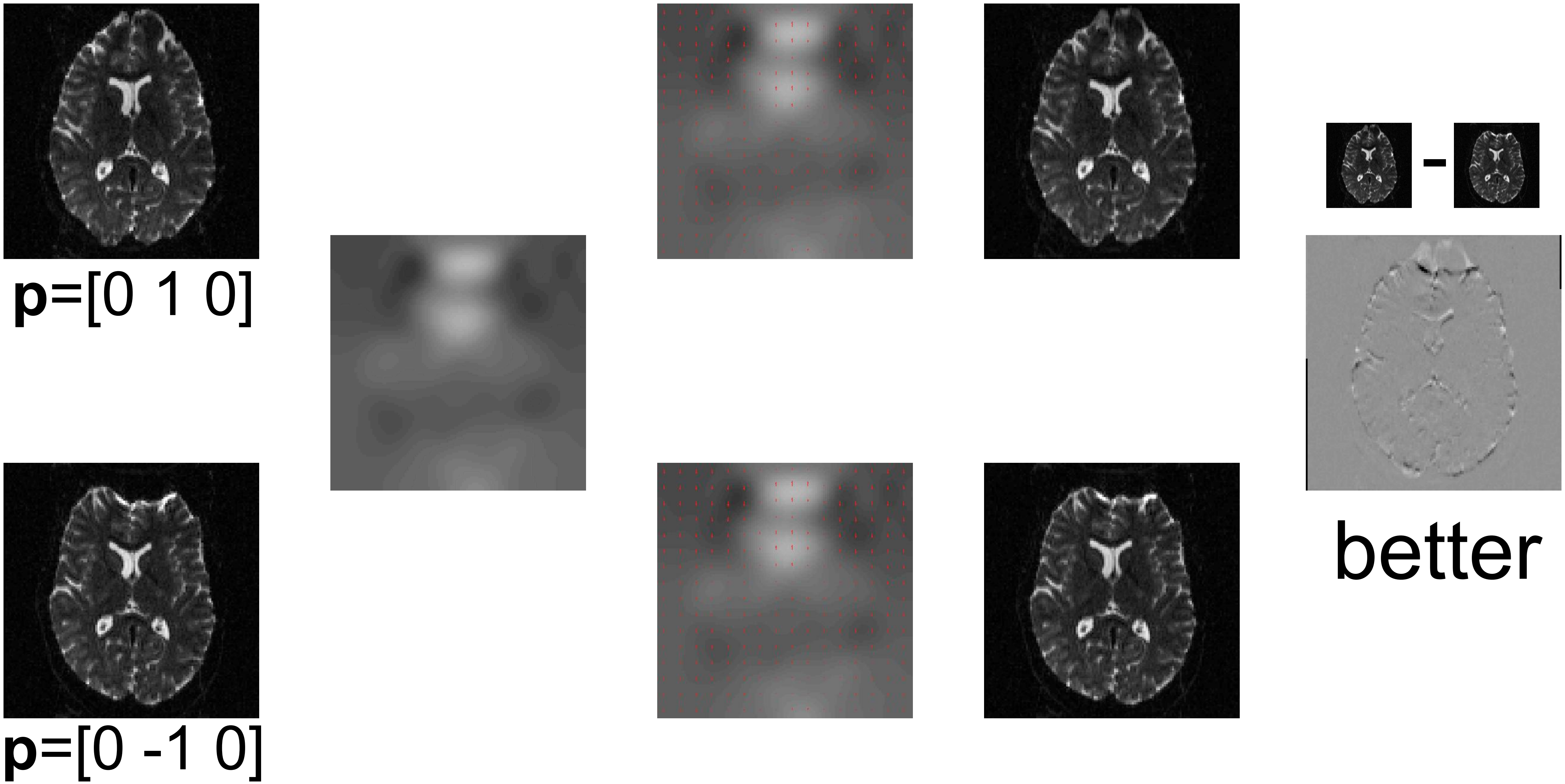
# How topup works (very briefly)




...and evaluates the results...

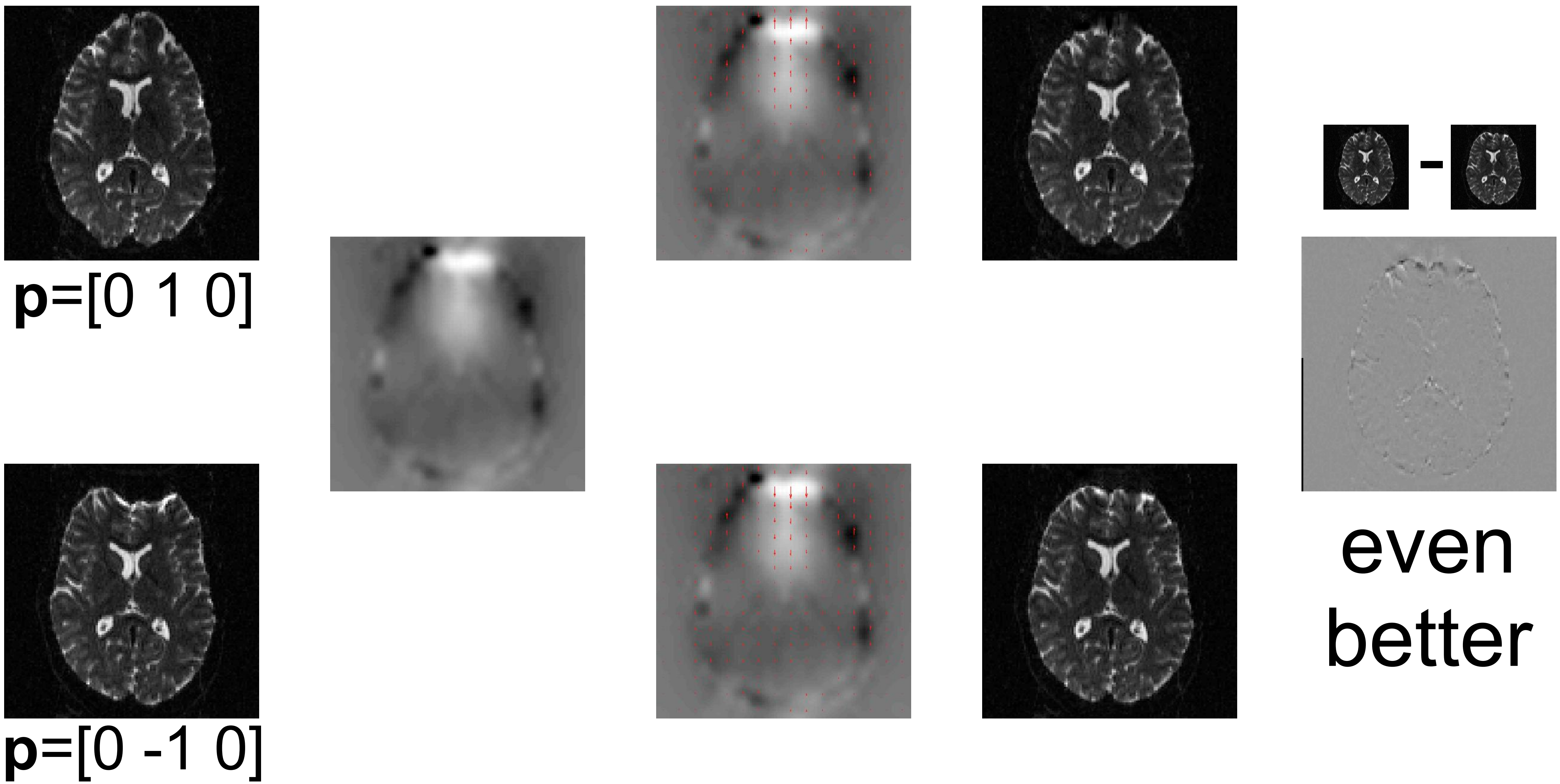
And **this** is the crucial bit.

# How topup works (very briefly)




Because topup can then "guess" another field

# How topup works (very briefly)

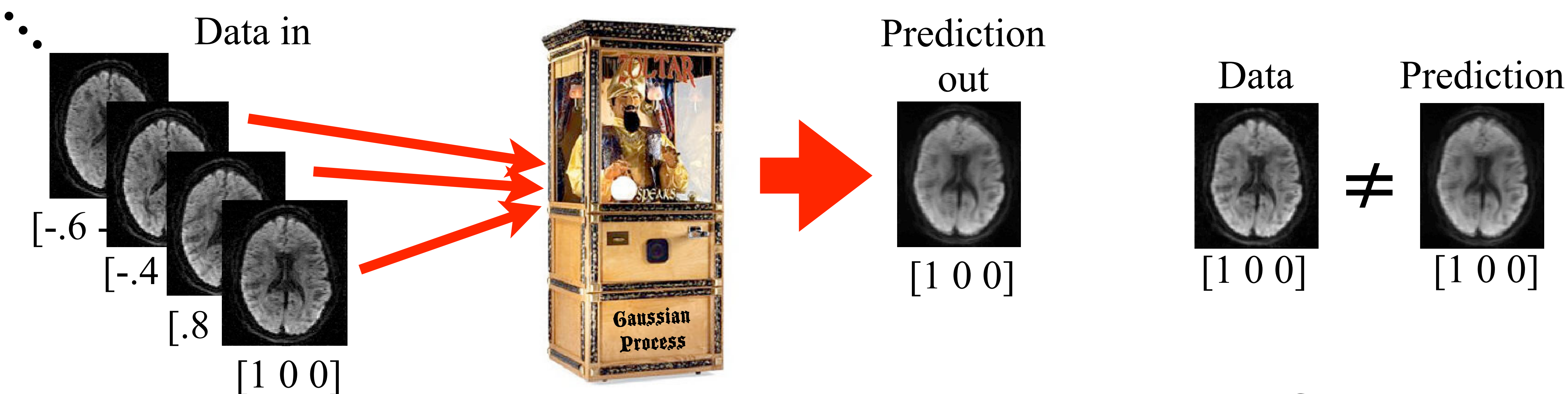



...and another...until it is happy, and then it “knows” the field

# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# Zoltar -- The prediction maker




Given some data in, Zoltar will make a prediction what the data “should” be.

The prediction for a given dwi will not be identical to the “input” for that dwi

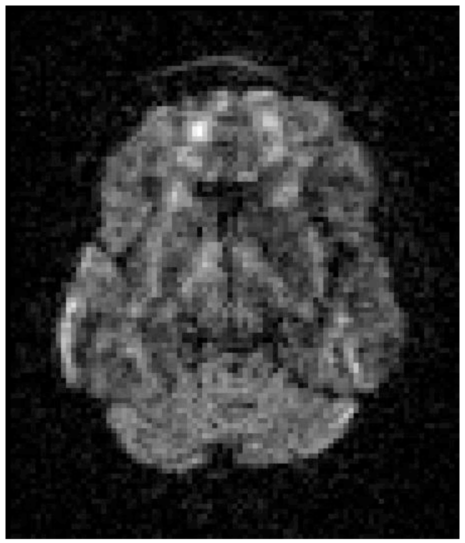
I know this sounds crazy, but suspend disbelief for now and we’ll get back to Zoltar later (he’s actually a Gaussian Process)

# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

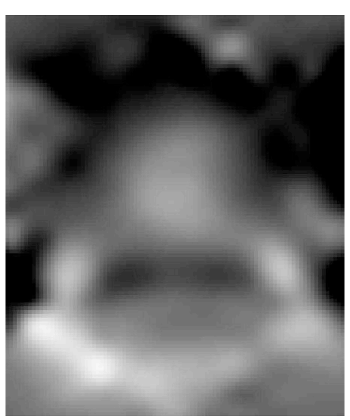
# How eddy works: Loading step

Pick the first dwi



Use current estimates of

Susc

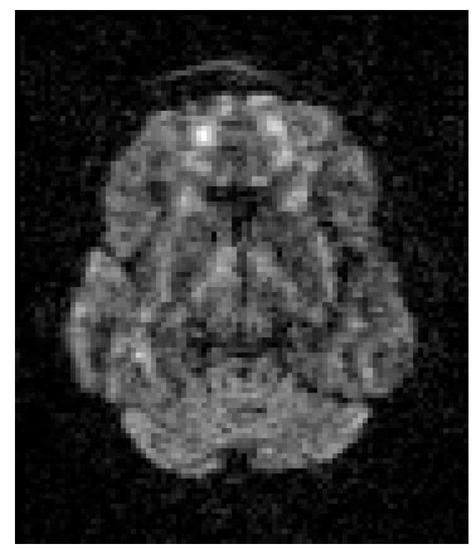


EC

MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

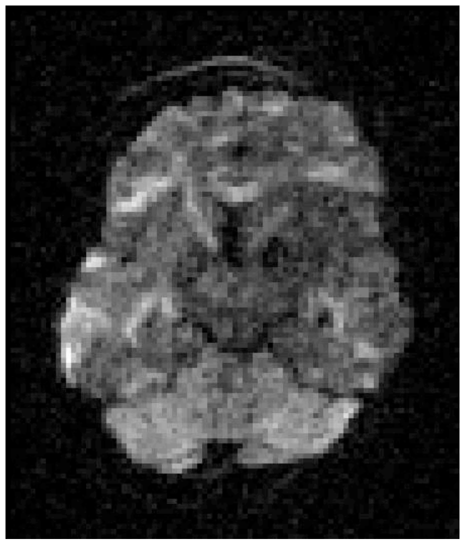
To correct image



And load into prediction maker

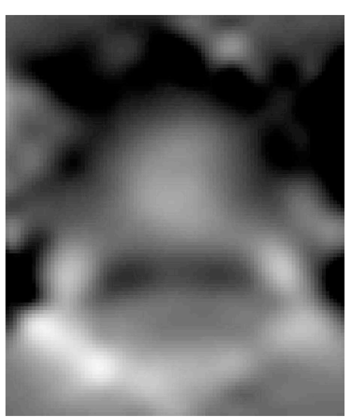
# How eddy works: Loading step

then the 2nd dwi



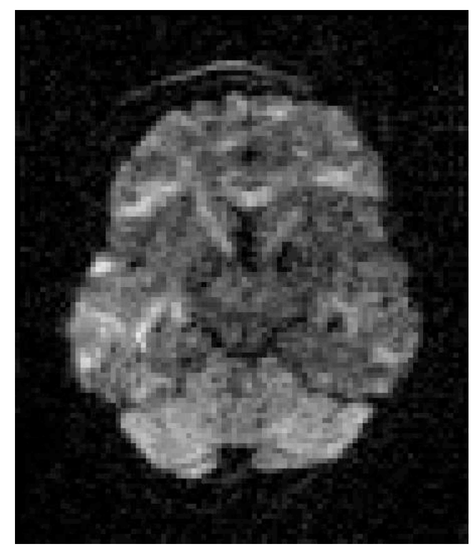
Use current estimates of

Susc EC MP



$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

To correct 2nd image

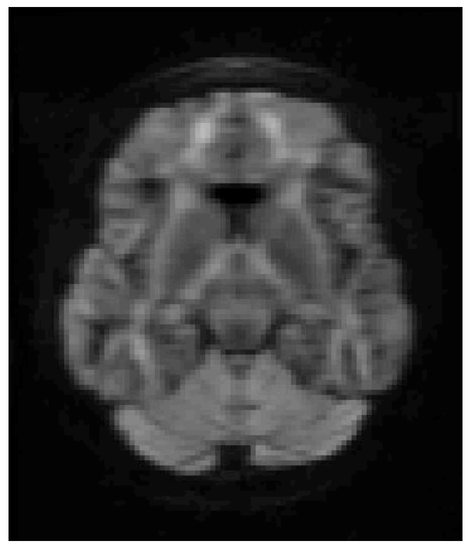


And load into prediction maker


Until we have loaded all dwis

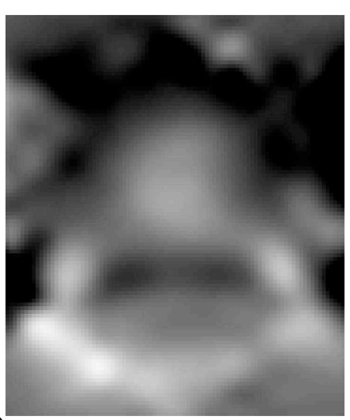
# How eddy works: Estimation step

Draw a prediction
for first dwi

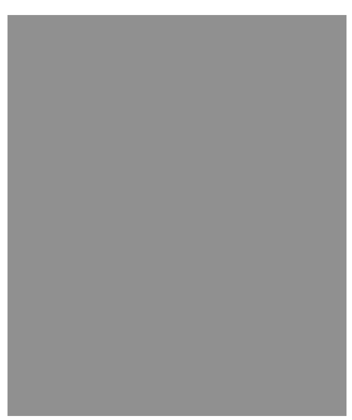



Use current estimates of

Susc



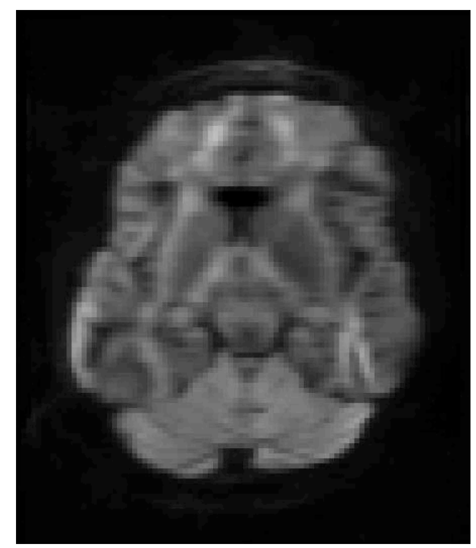
EC



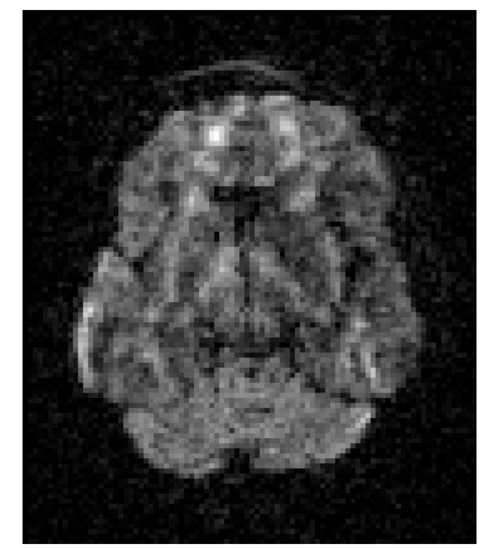
MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Invert

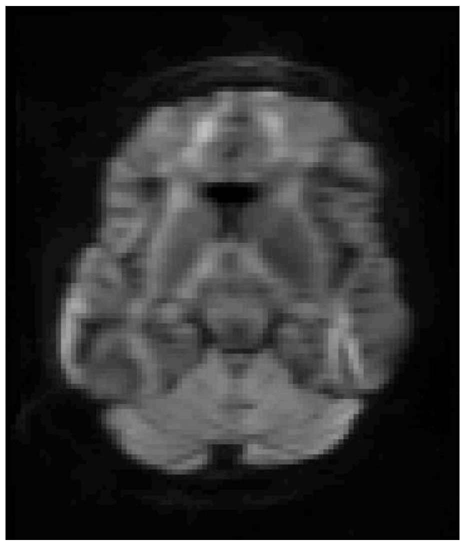


To get prediction in "observation space"



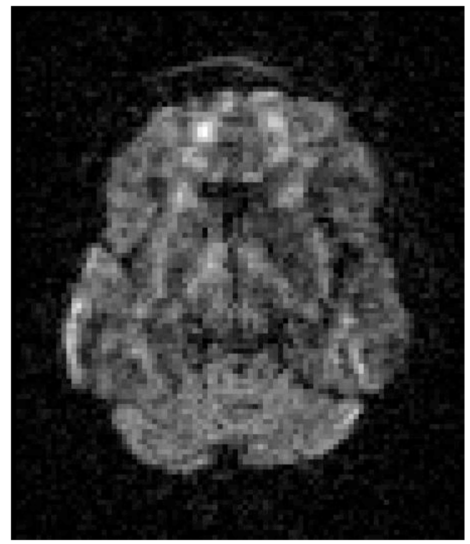
And compare to actual observation

# How eddy works: Estimation step



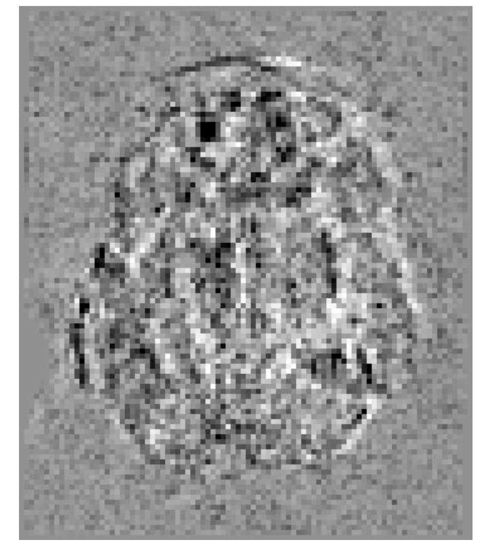
Prediction in "observation space"

-



Observation

=

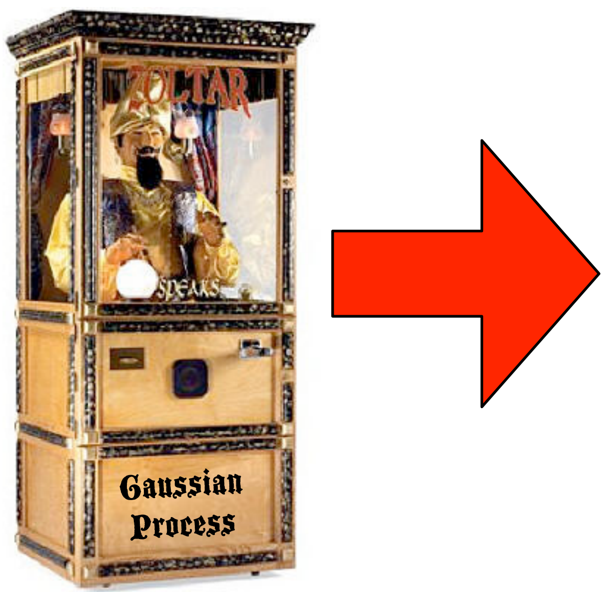


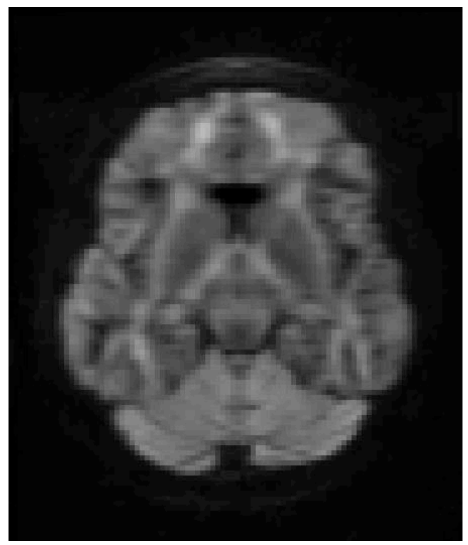
This observed difference is what drives the estimation

But how do we go from this observed difference to estimates of the eddy currents (and movements)?

# How eddy works: Estimation step

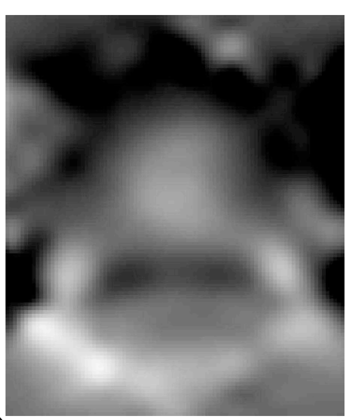
Draw a prediction
for first dwi



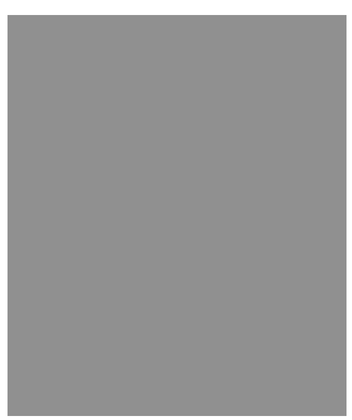



Use current estimates of

Susc



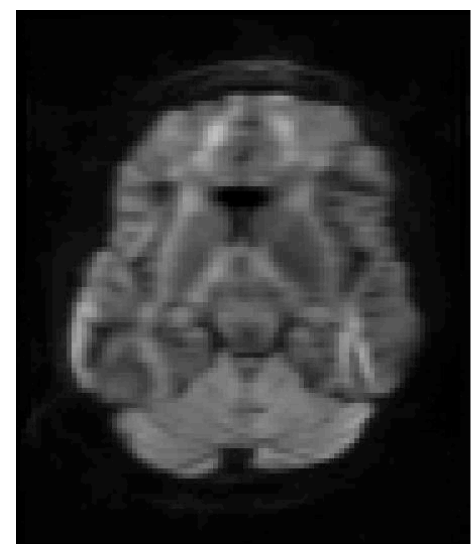
EC



MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Invert

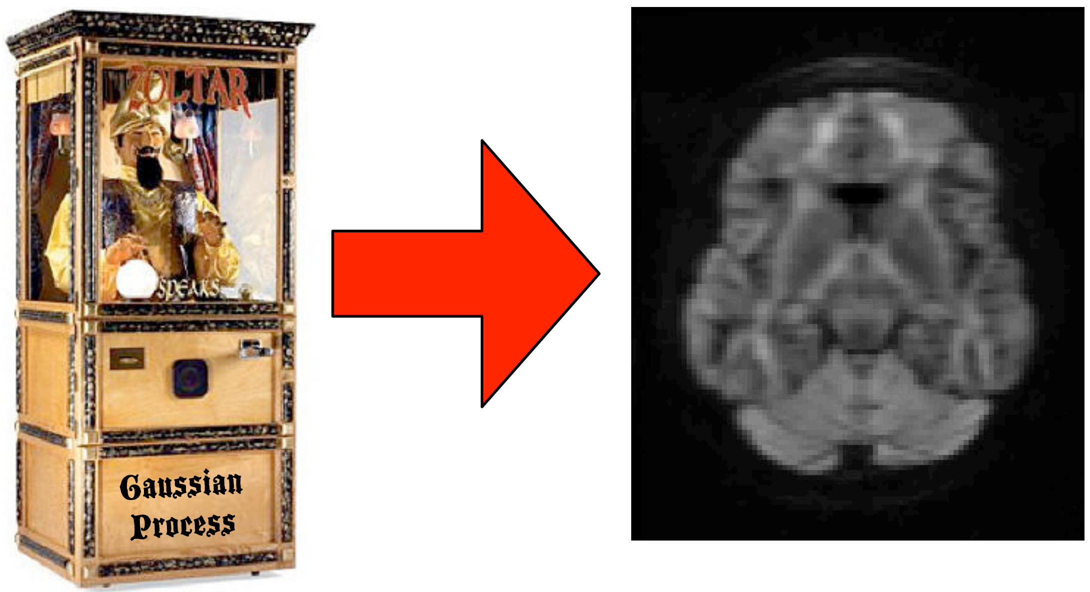


Prediction with
current
estimates.

But what if the
estimates
were a little
different?

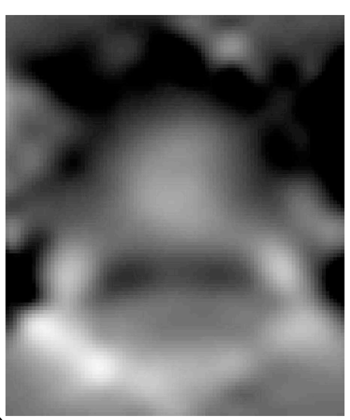
# How eddy works: Estimation step

Draw a prediction for first dwi

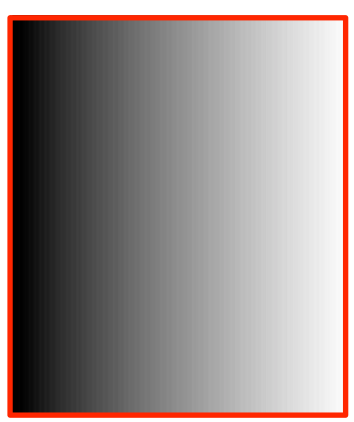



Use current estimates of

Susc



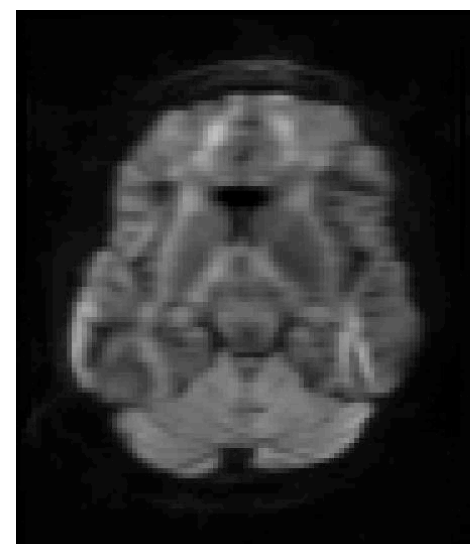
EC



MP

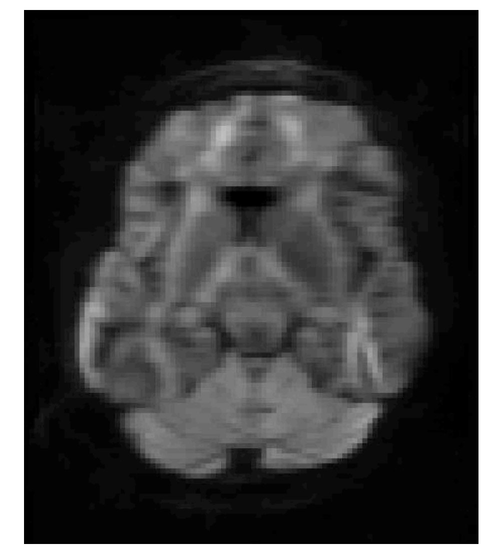
$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Invert



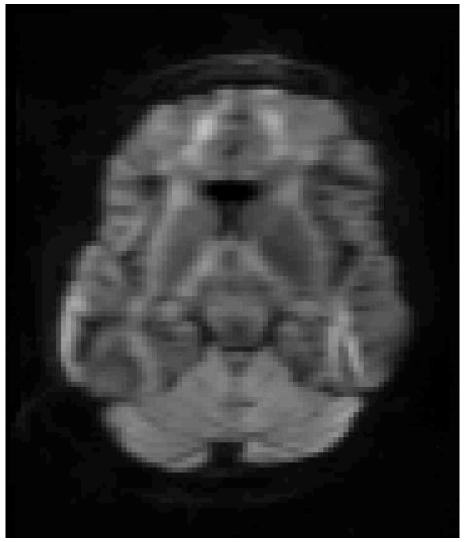
Prediction with "different" estimates.

(linear x-gradient)

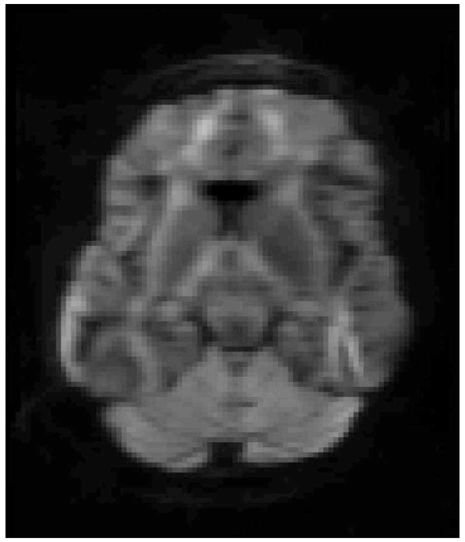


Prediction with current estimates.

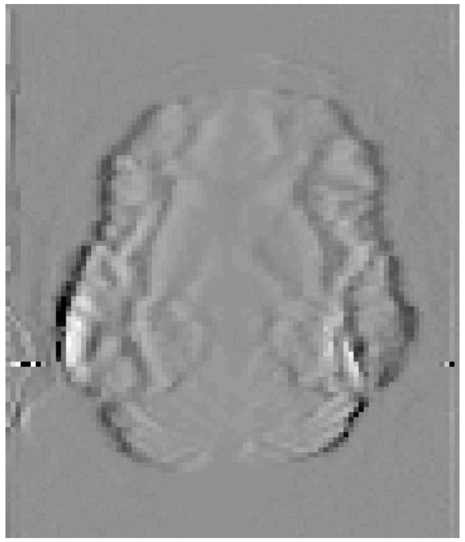
# How eddy works: Estimation step



-



=

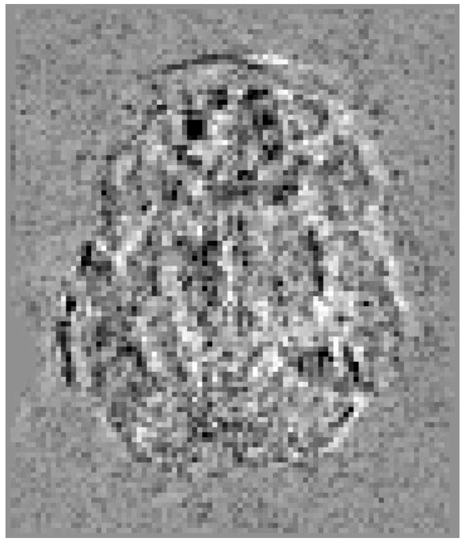


Prediction with "different" estimates. (linear x-gradient)

Prediction with current estimates.

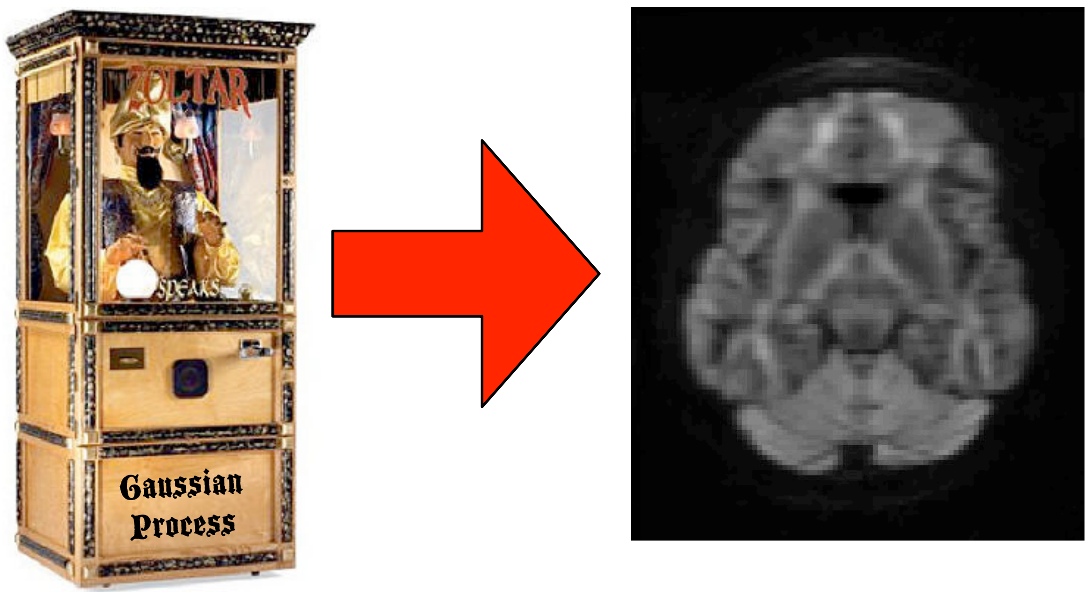
Derivative w.r.t. linear x-gradient

And then we compare this to the observed difference



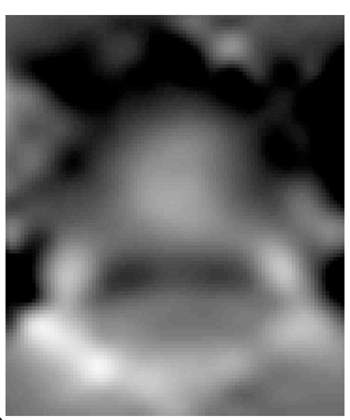
# How eddy works: Estimation step

Draw a prediction
for first dwi

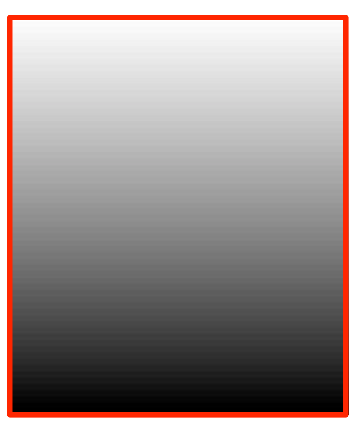



Use current estimates of

Susc



EC

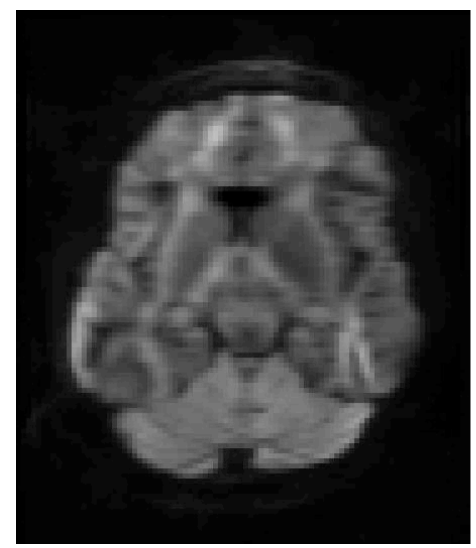


MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

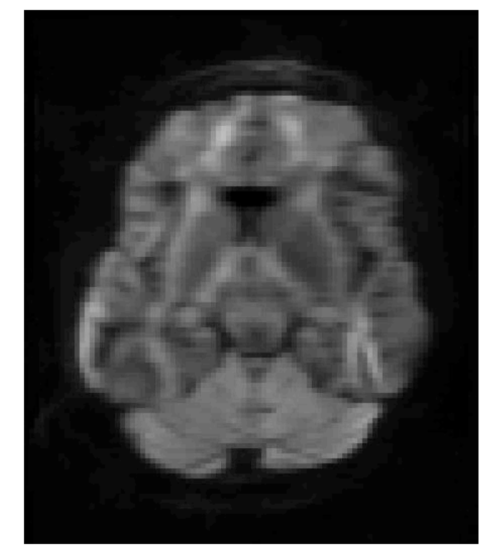
Invert

And we can repeat
this for all EC and
MP parameters



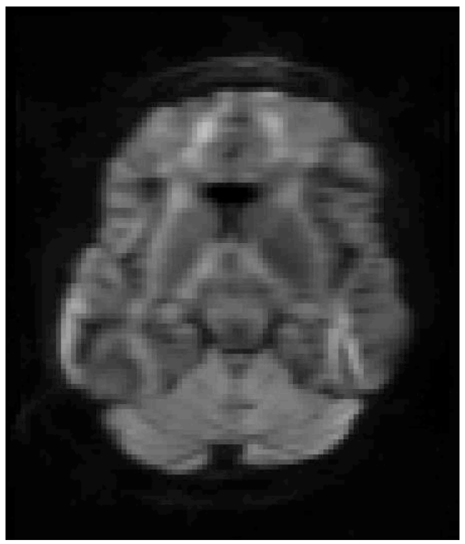
Prediction with
"different"
estimates.

(linear y-gradient)



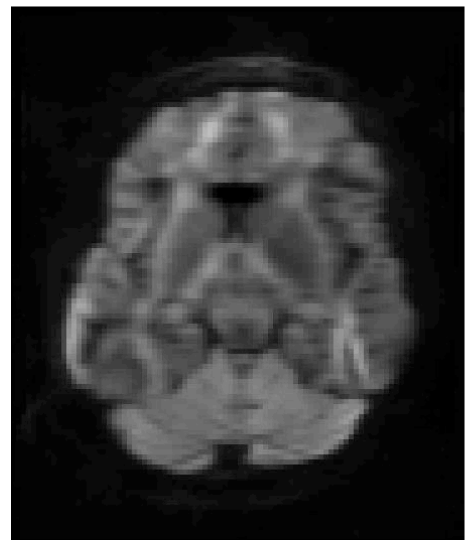
Prediction with
current
estimates.

# How eddy works: Estimation step



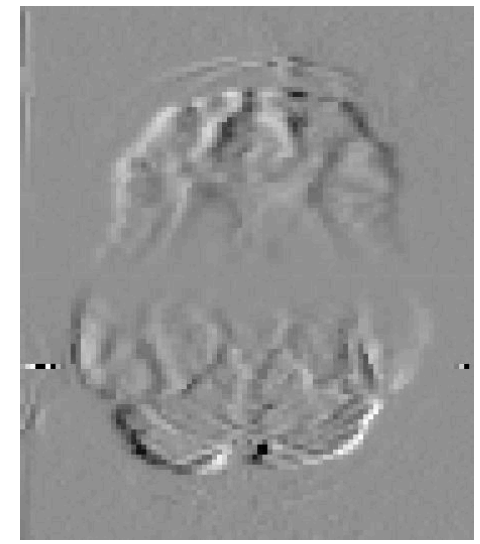
Prediction with "different" estimates. (linear y-gradient)

\-



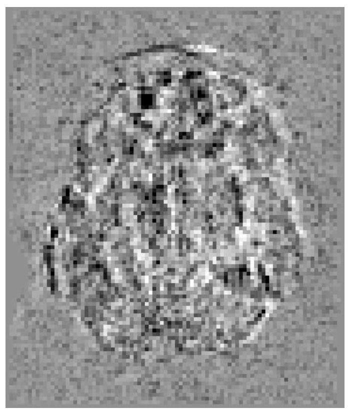
Prediction with current estimates.

=

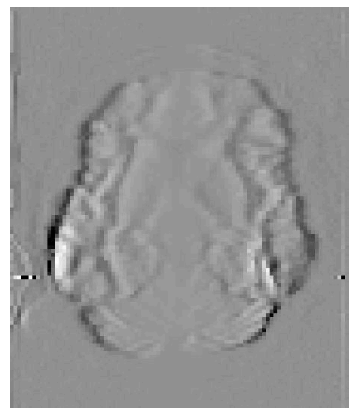


Derivative w.r.t. linear y-gradient

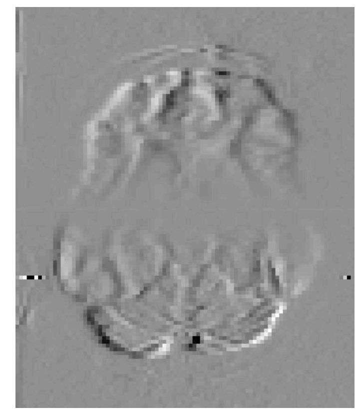
# How eddy works: Estimation step



$\approx \beta_1$



$+\beta_2$



$+\beta_3$

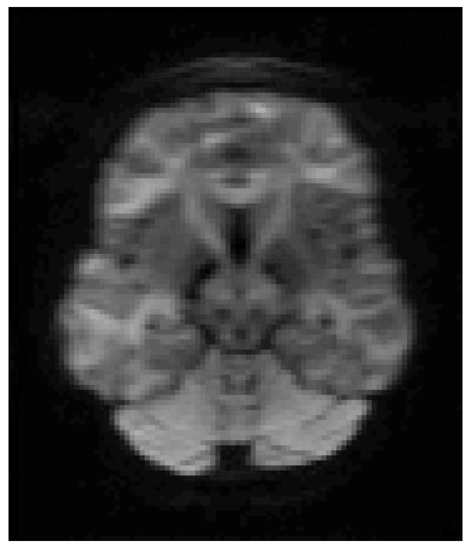
$+\dots$

observed difference | x-grad | y-grad | z-grad

We model the observed difference as a linear combination of the “tentative” causes. It is trivial to solve for β and get a first estimate of these parameters.

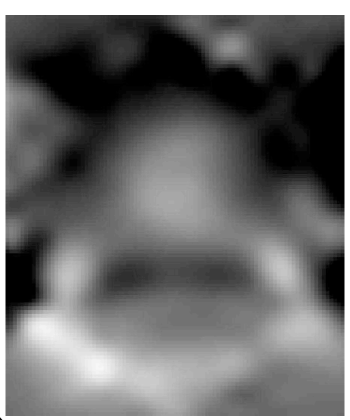
# How eddy works: Estimation step

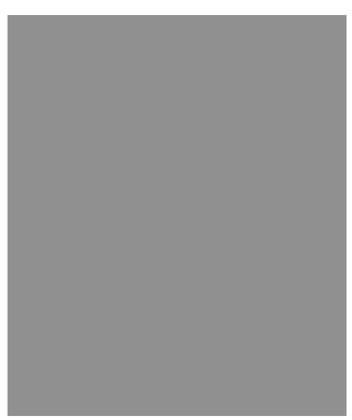
Draw a prediction
for 2nd dwi

Use current estimates of

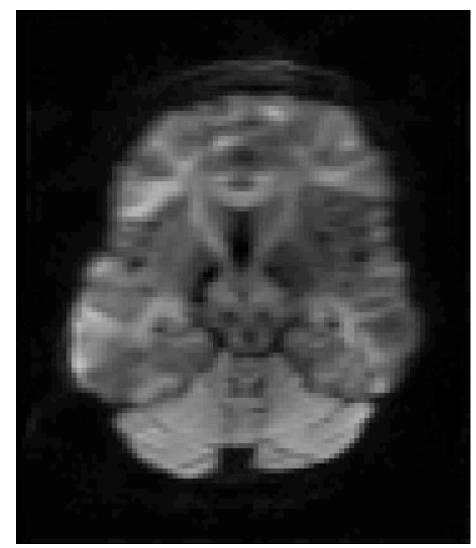
Susc EC MP

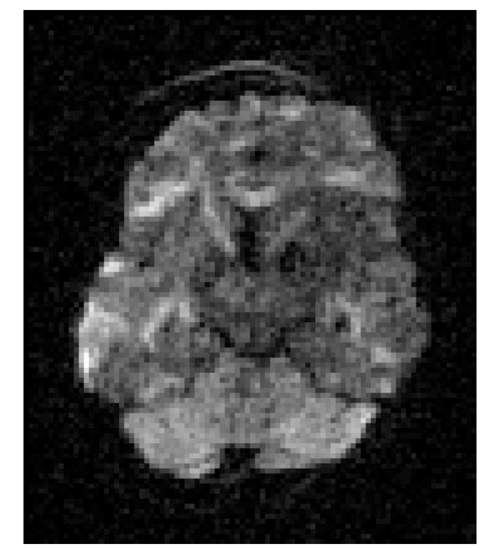




$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

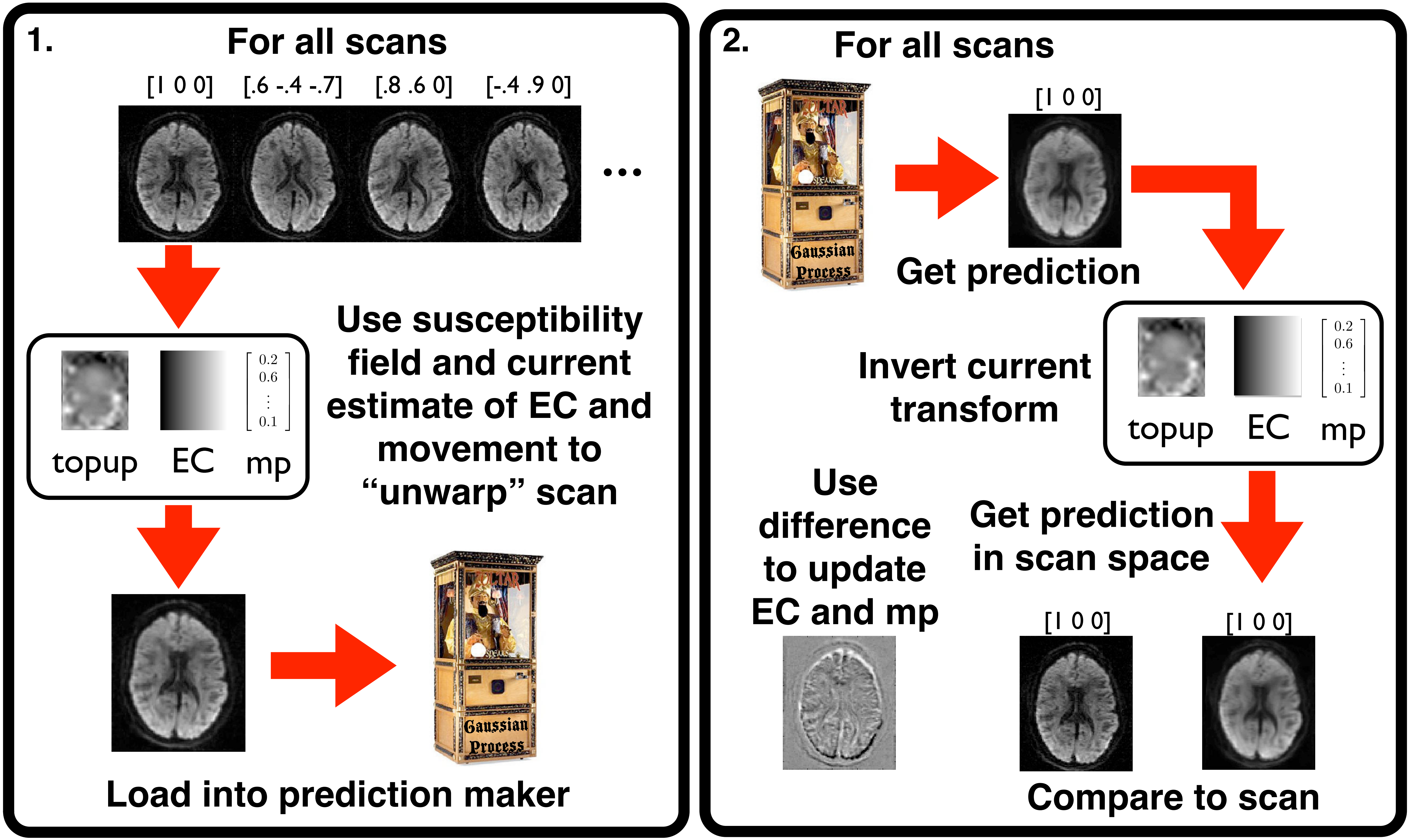
Invert





And then we repeat
the procedure for the
next dwi ...

# How eddy works



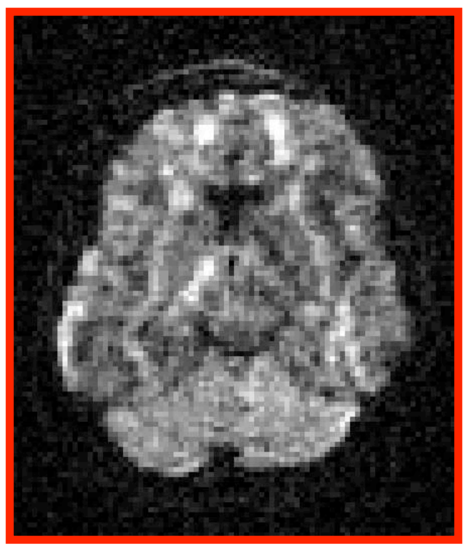
# Outline of the talk

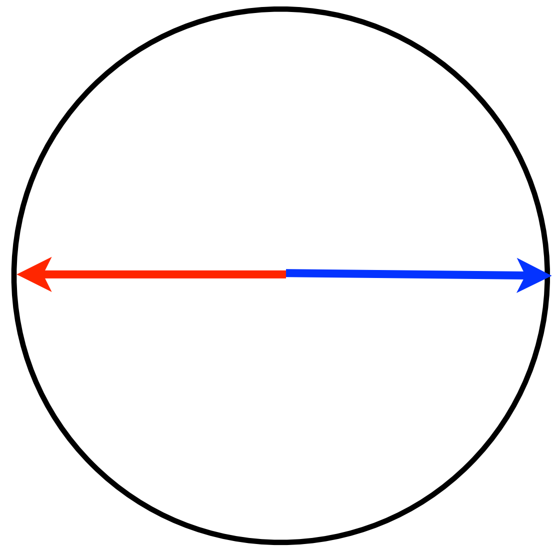
- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

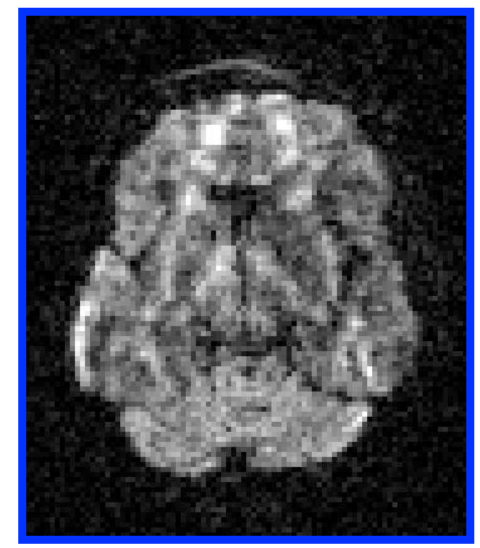
# Under the hood of Zoltar

Let us first look at what we
could have done ... but didn't

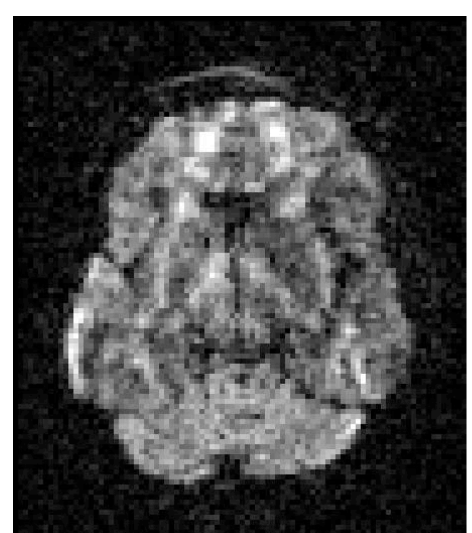
Diffusion gradient







We could have used
the average of dwis
with opposing gradients
as our prediction

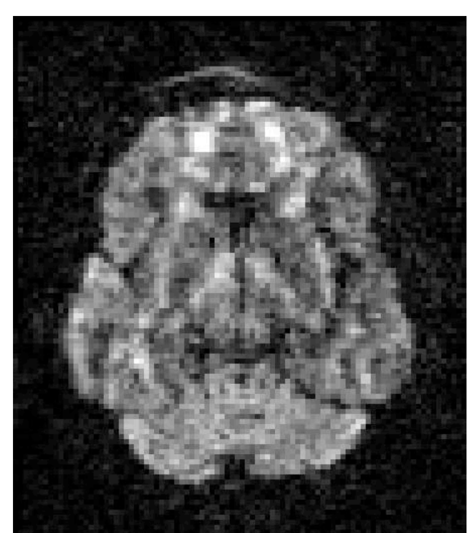


but didn't ...

# Under the hood of Zoltar

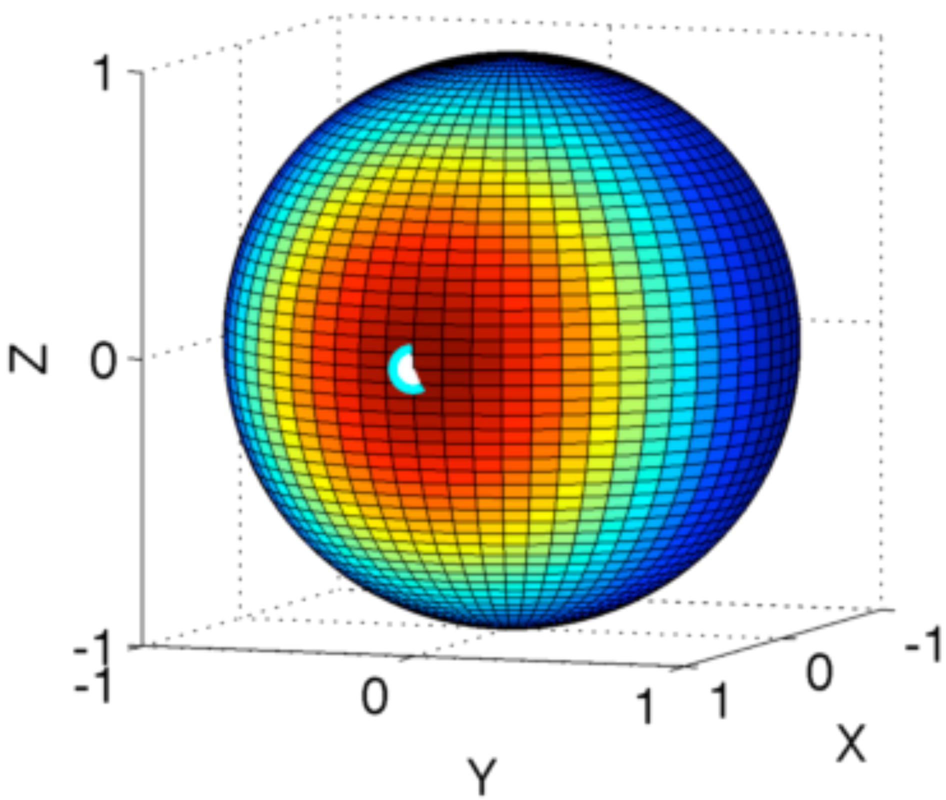
Let us first look at what we could have done ... but didn't

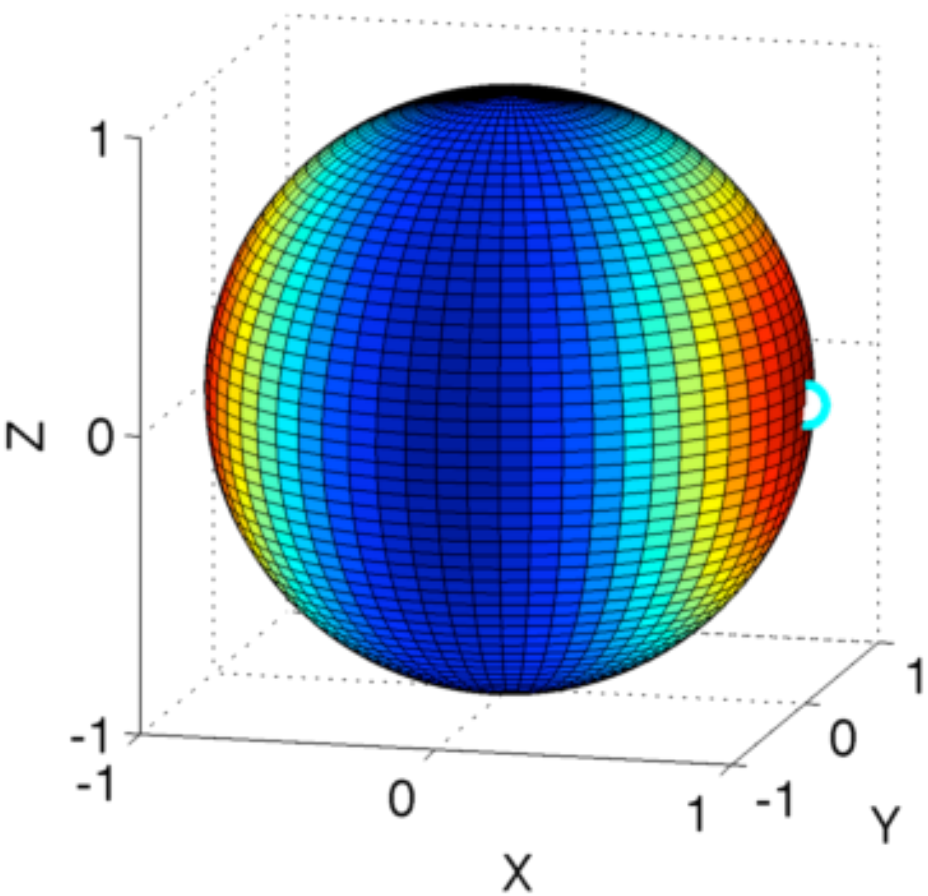

We could have used the average of dwis with opposing PE-blips as our prediction



but didn't ...

# Under the hood of Zoltar

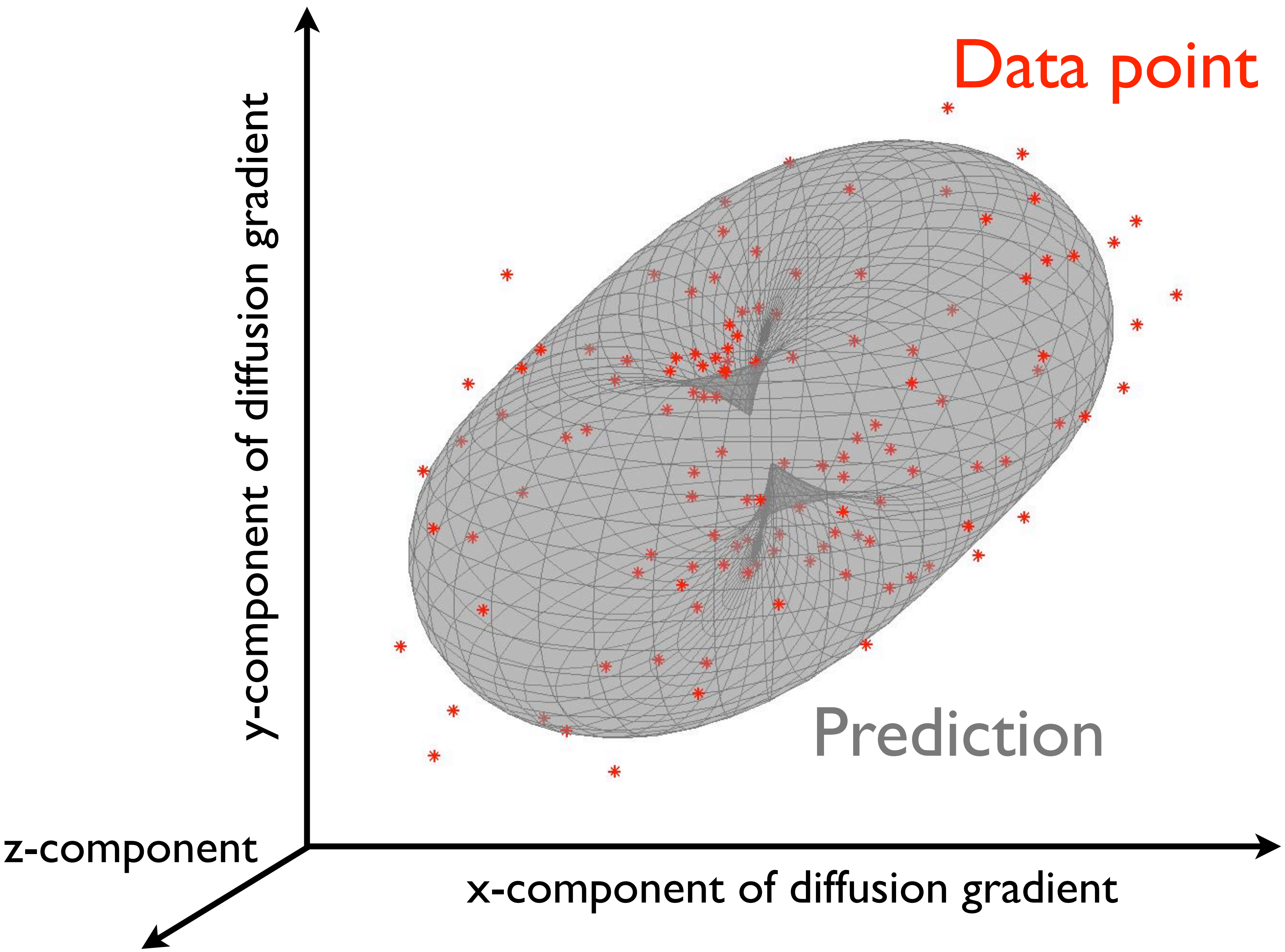


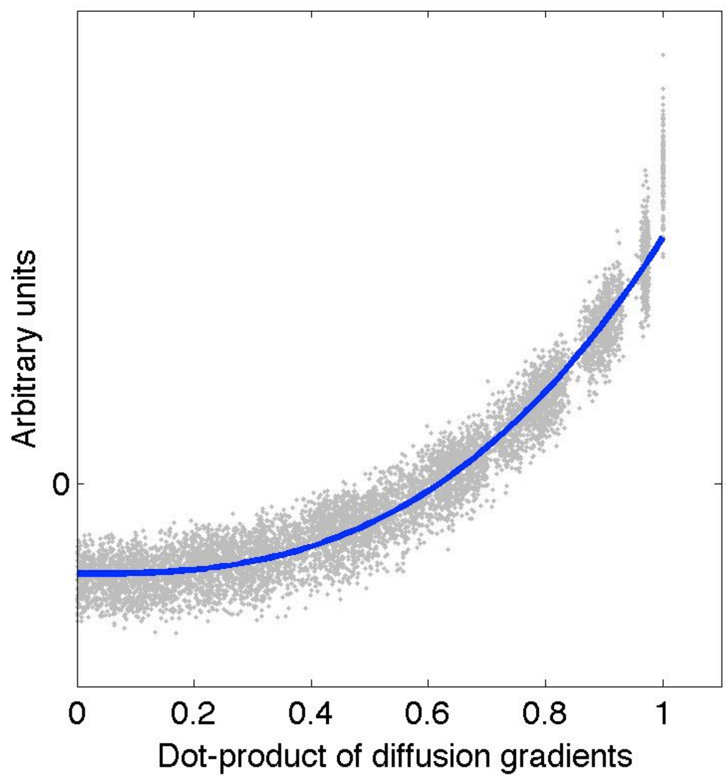




We used a weighted average of all dwis.

The weights were calculated using a Gaussian Process

# Under the hood of Zoltar

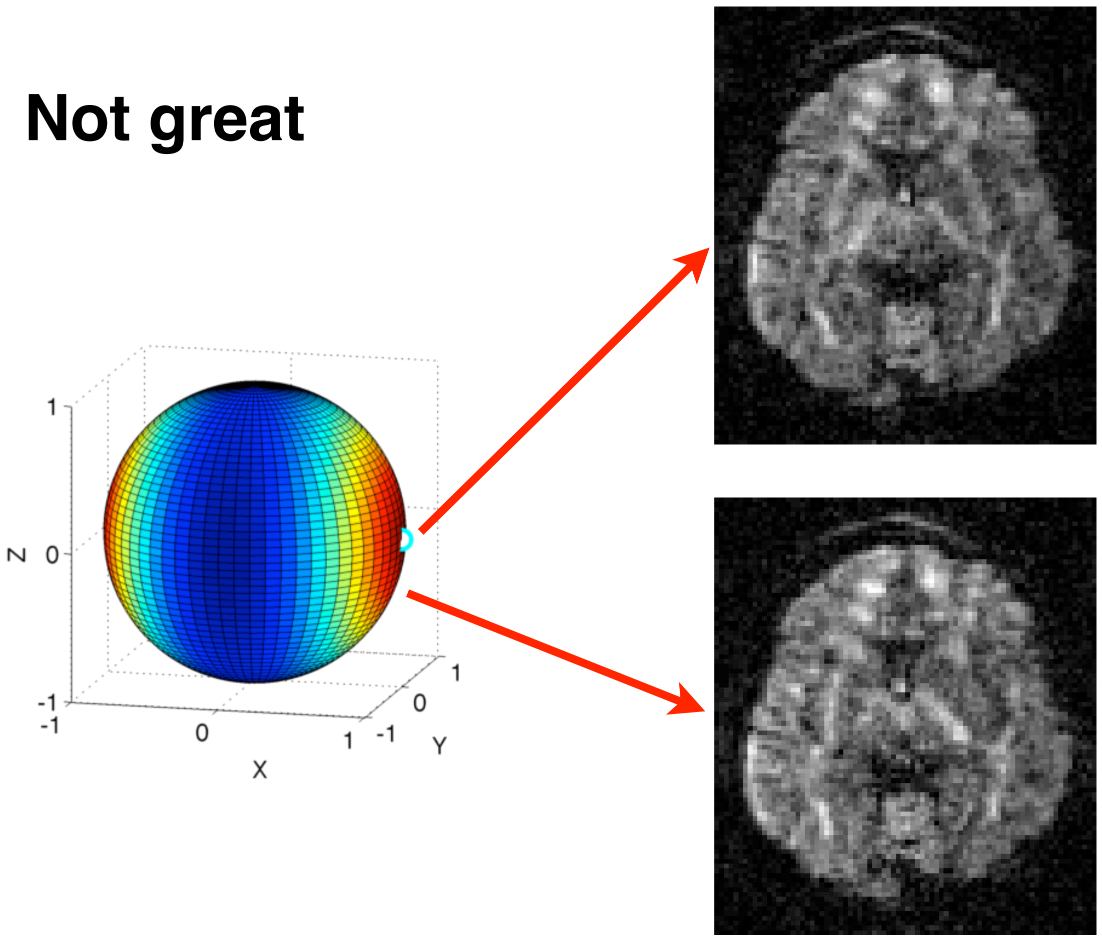



$$\hat{y}_{\mathbf{g}} = K(\mathbf{g}, \mathbf{G}) \left[ K(\mathbf{G}, \mathbf{G}) + \sigma^2 \mathbf{I} \right]^{-1} \mathbf{y}$$

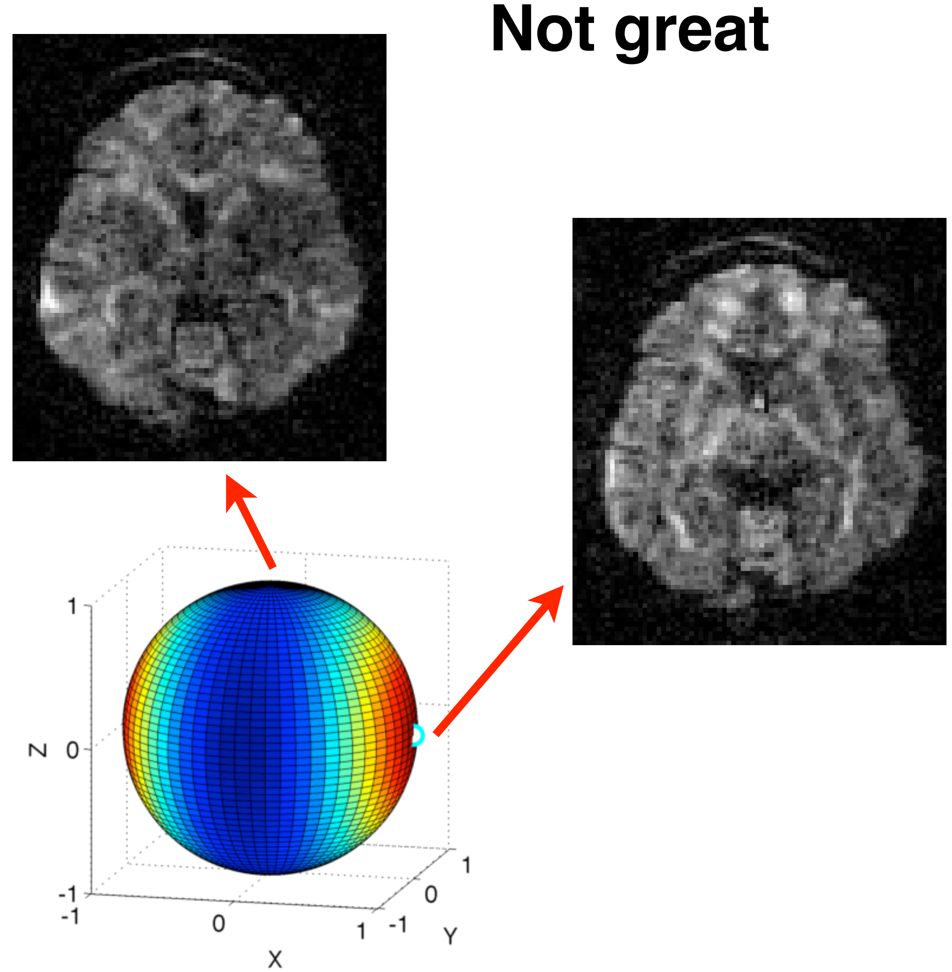
# Where is the “good”information?

**Not great**



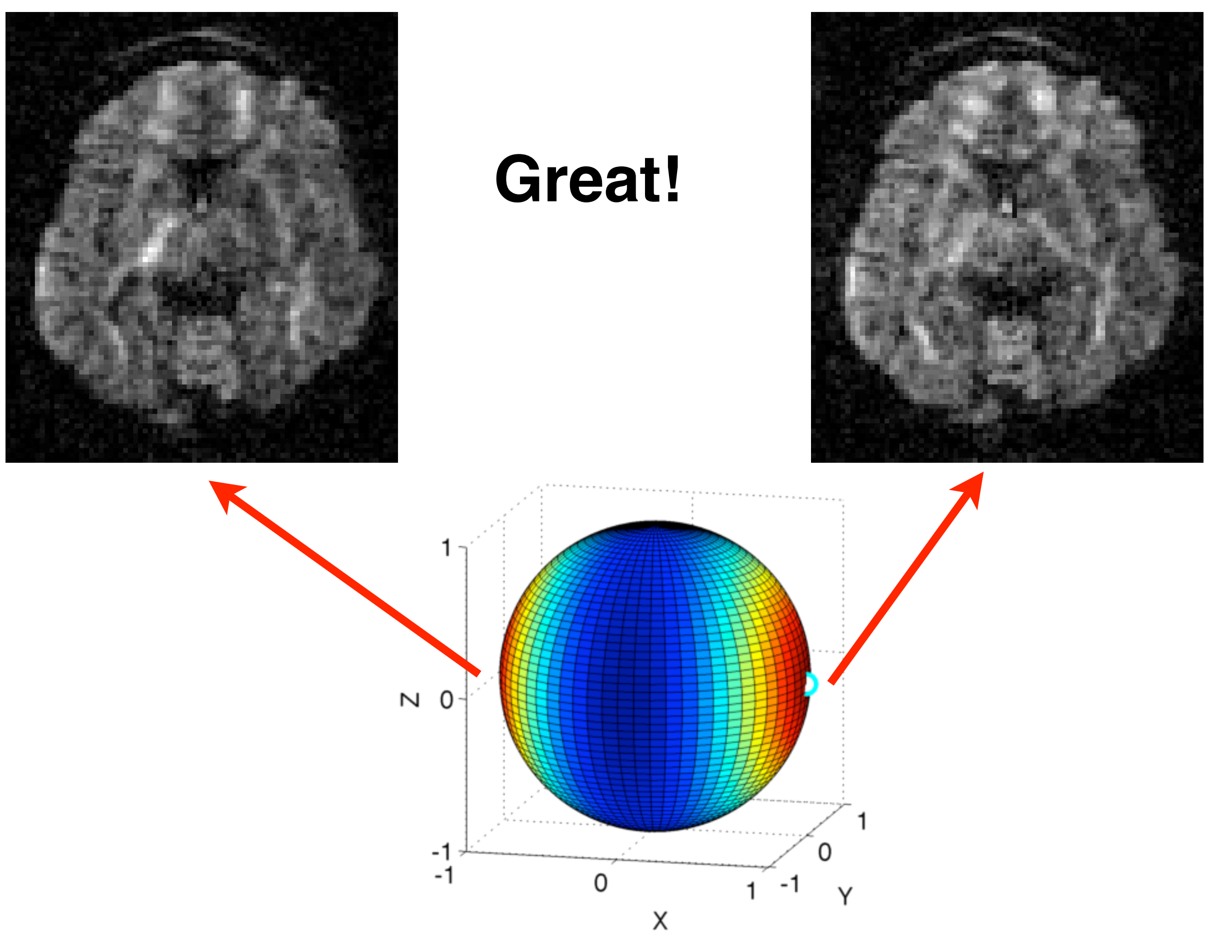

**Contrast similar, but so are distortions**

**Not great**

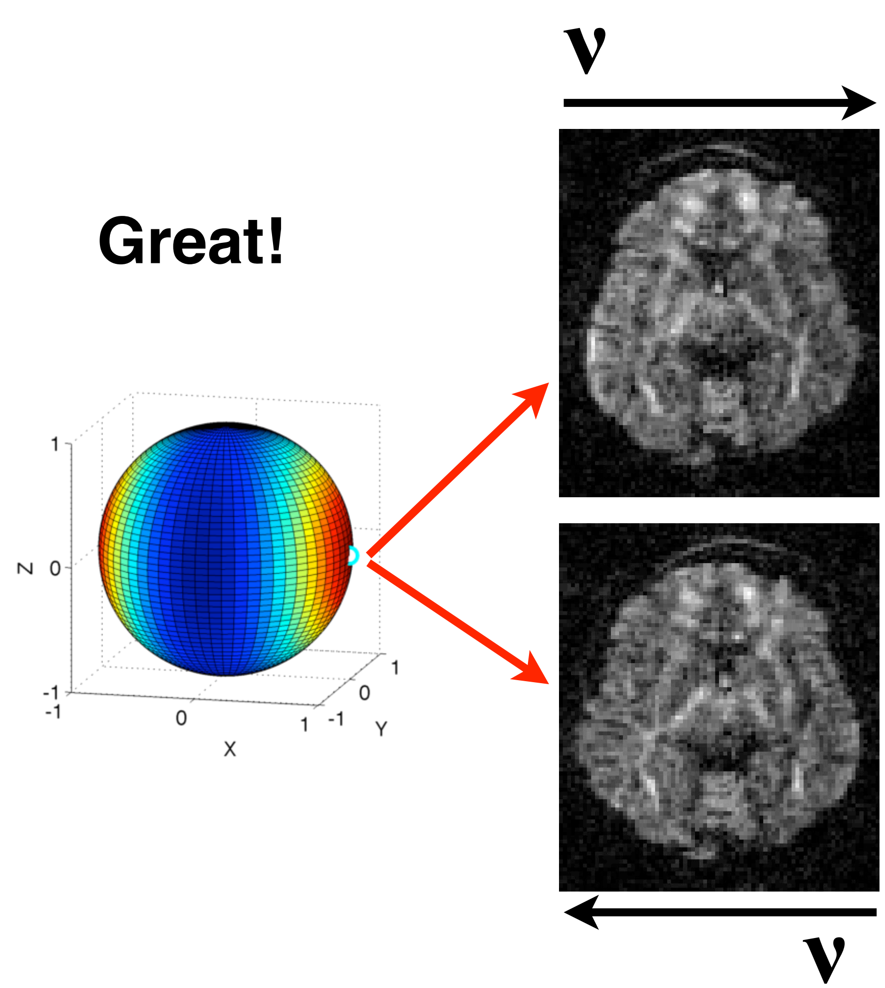



**Distortions different, but so is contrast**

# Where is the "good" information?




Distortions different, similar contrast

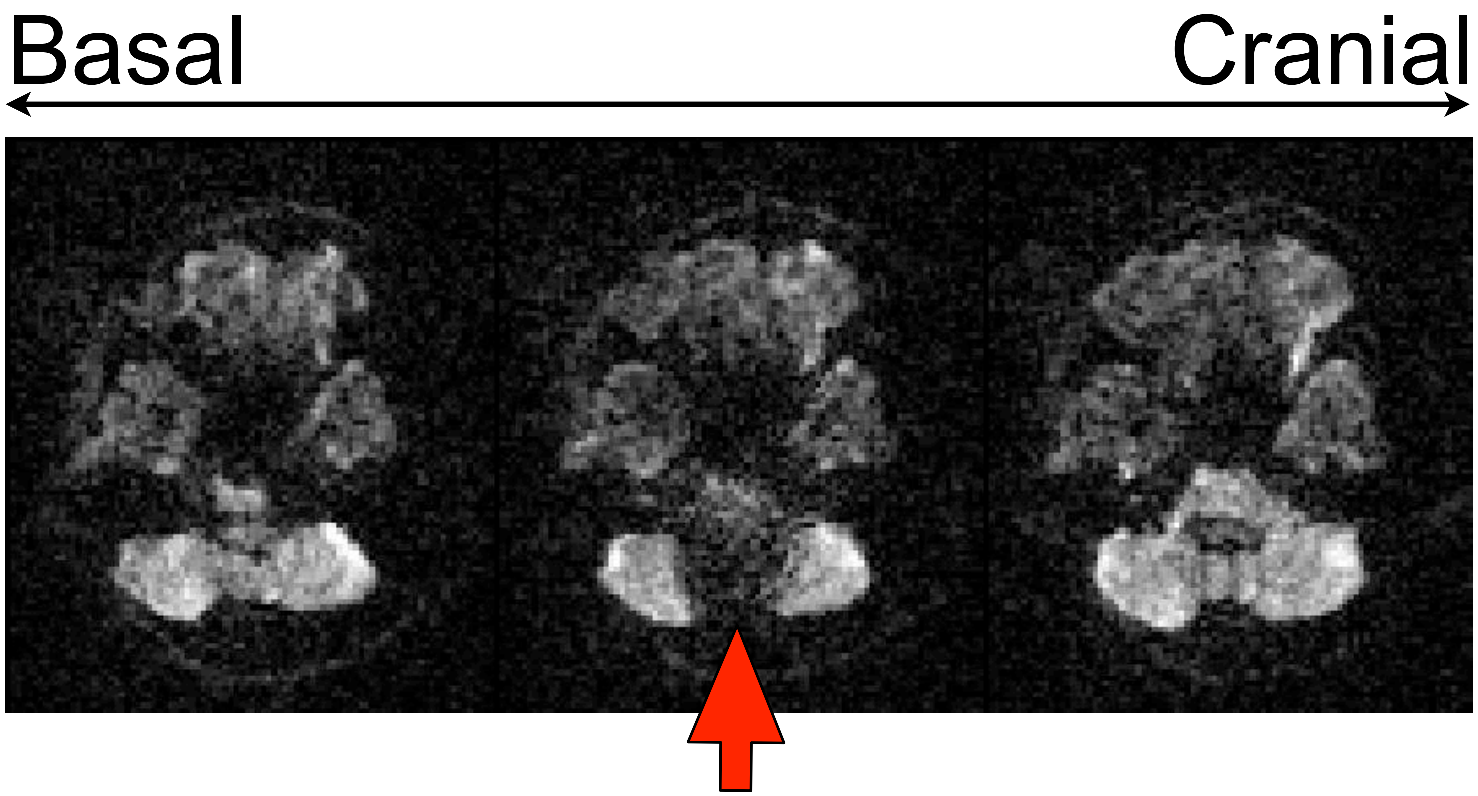


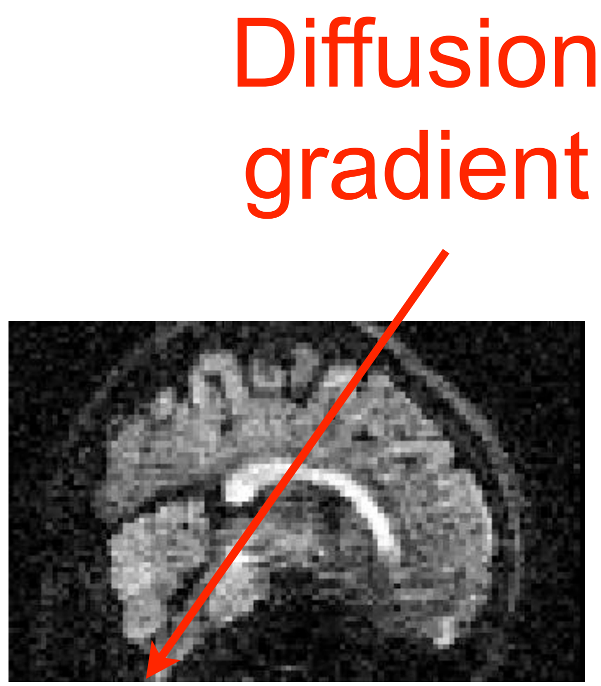

Distortions opposed, identical contrast.
N.B. needs topup field

# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

# Outlier detection

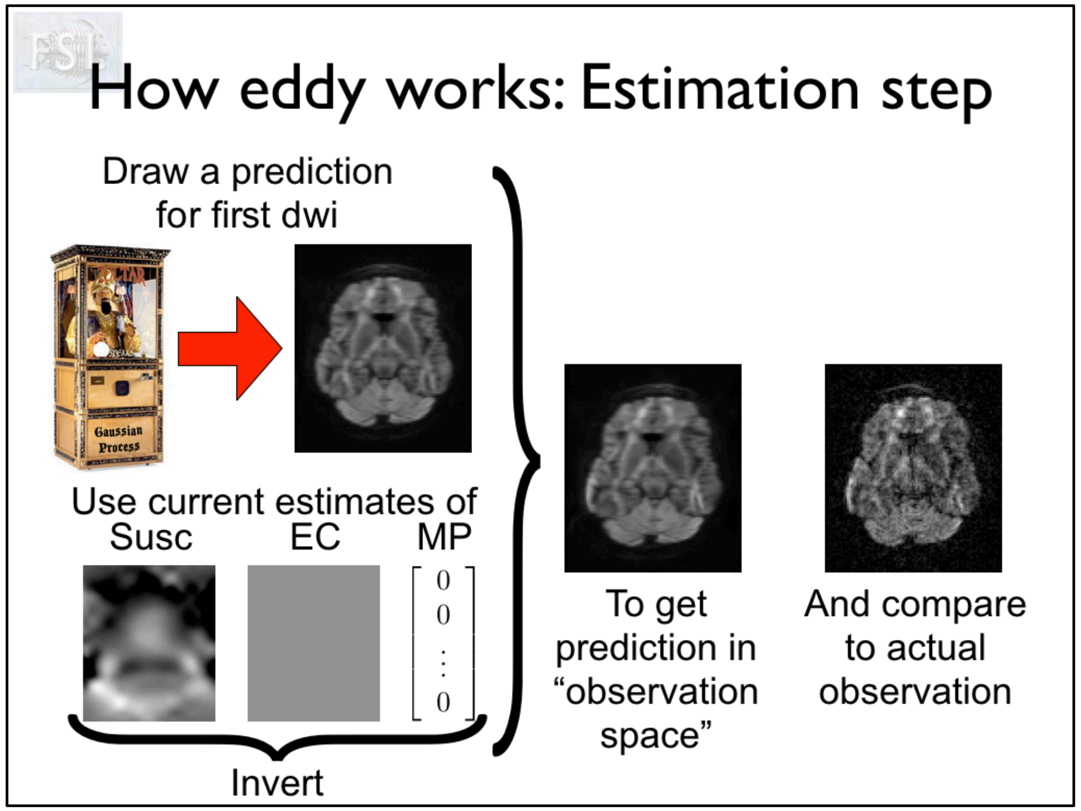


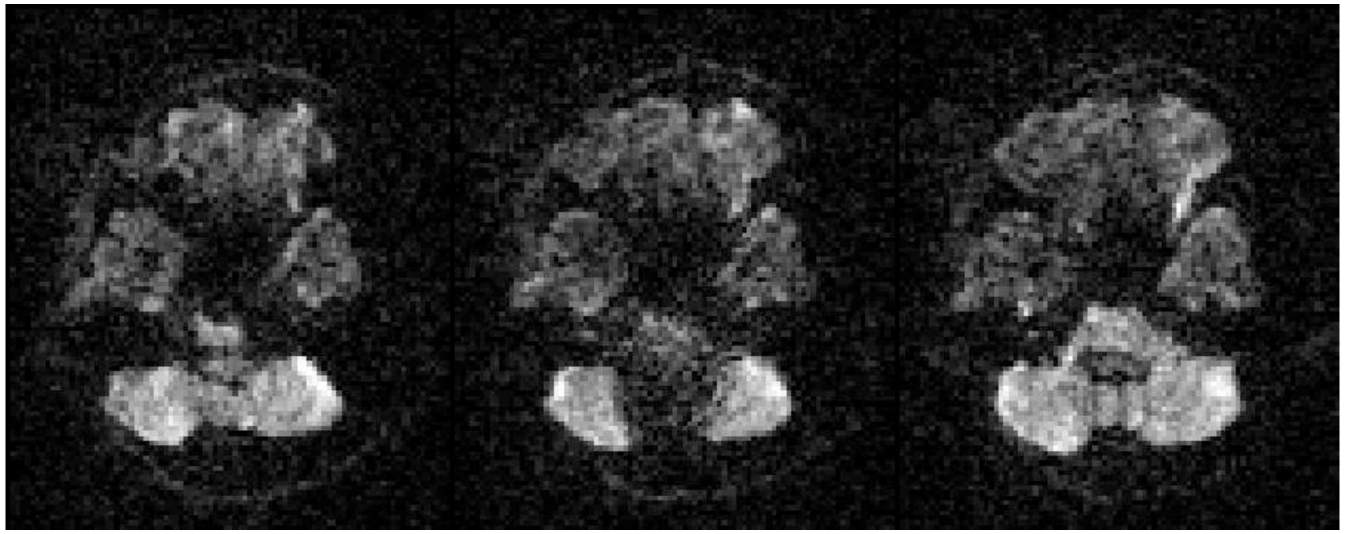

Sometimes things don’t go as we would like them to.

These drop-outs are related to coherent movement coinciding with the diffusion encoding which causes the signal to be translated outside the acquisition window in *k*-space.

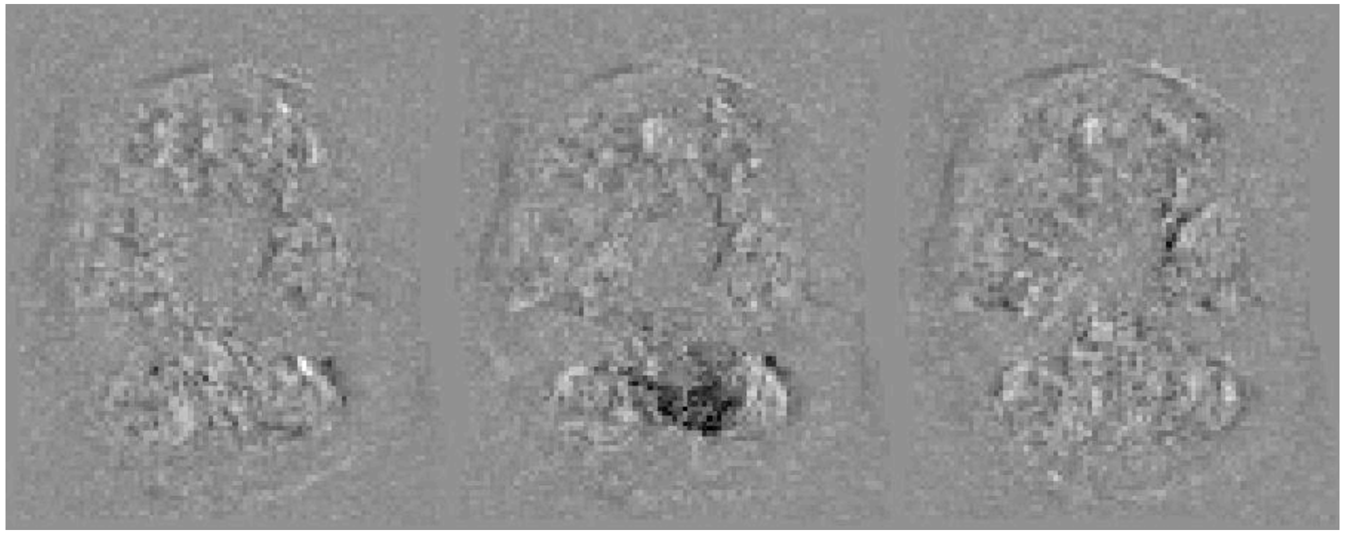
# Outlier detection




Observed data



Observed - predicted

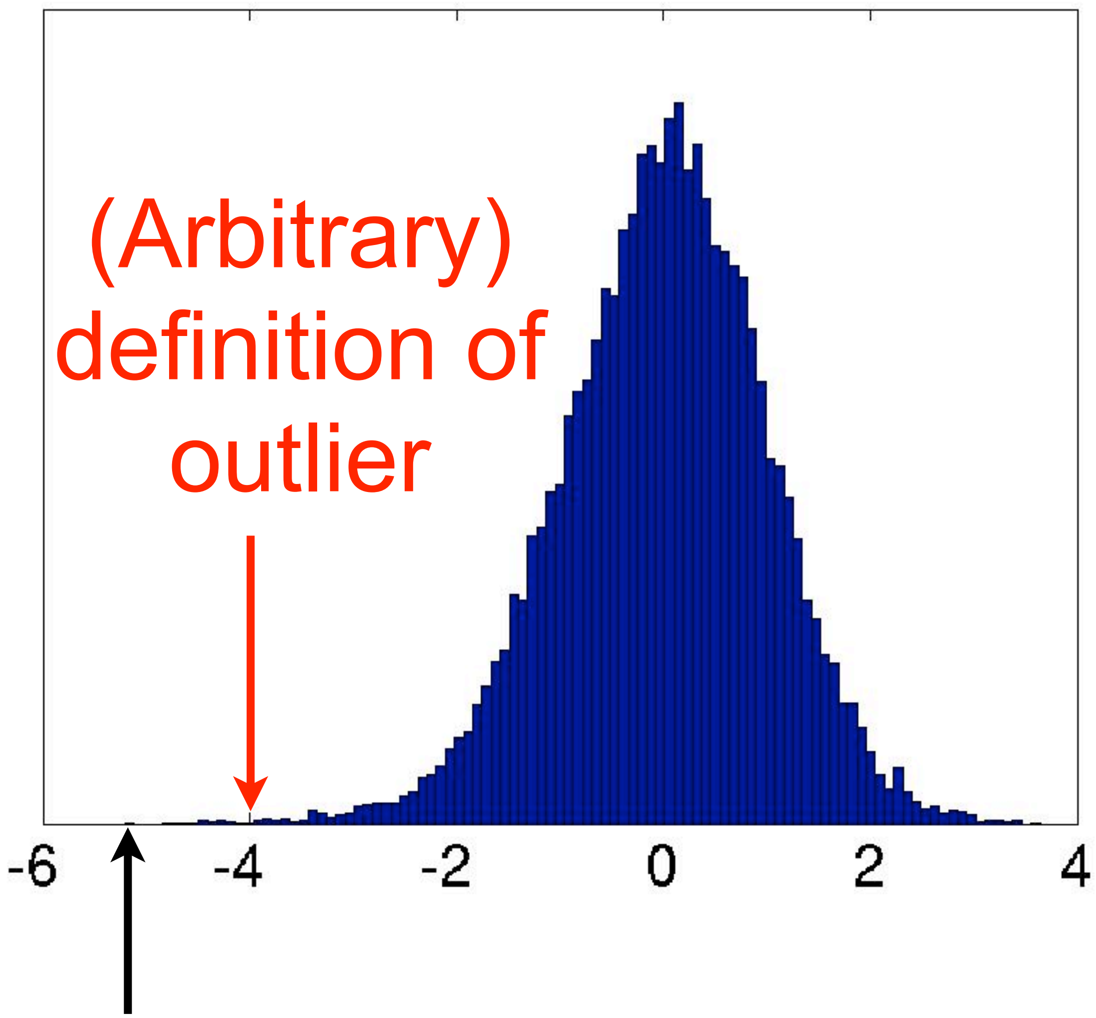


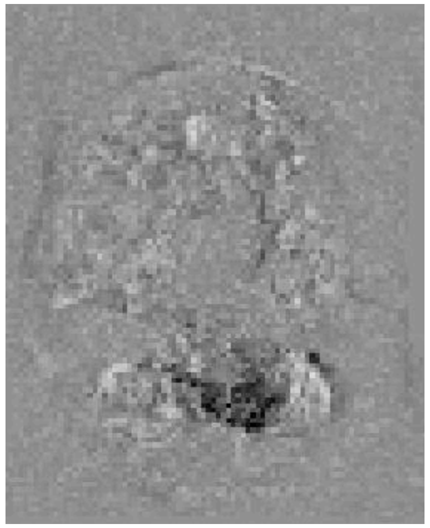
$\overline{x} = 0.084$ $\overline{x} = -0.791$ $\overline{x} = -0.125$

Remember that we do all comparisons in observation space.

This allows us to calculate the a per-slice mean difference between observation and prediction

# Outlier detection



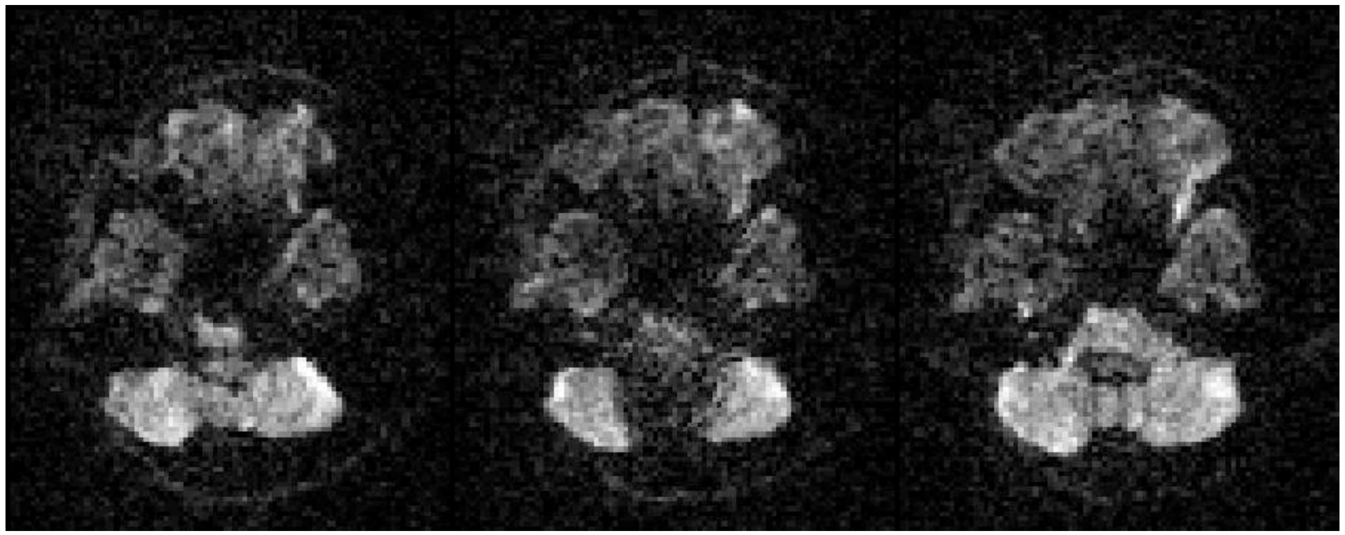
Worst slice

We can calculate the mean difference for every slice in every volume and get an empirical distribution that we can convert to z-scores

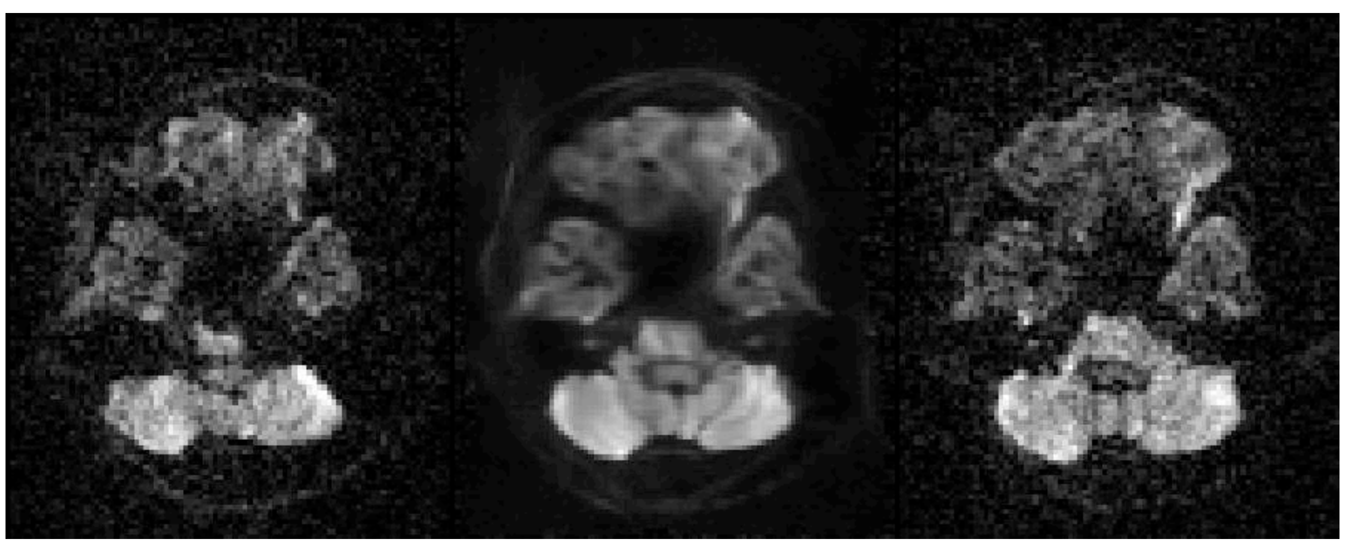
We can define an outlier slice as one with a z-score above an (arbitrary) threshold. We then have a choice of reporting outliers and/or replacing them with their predictions.

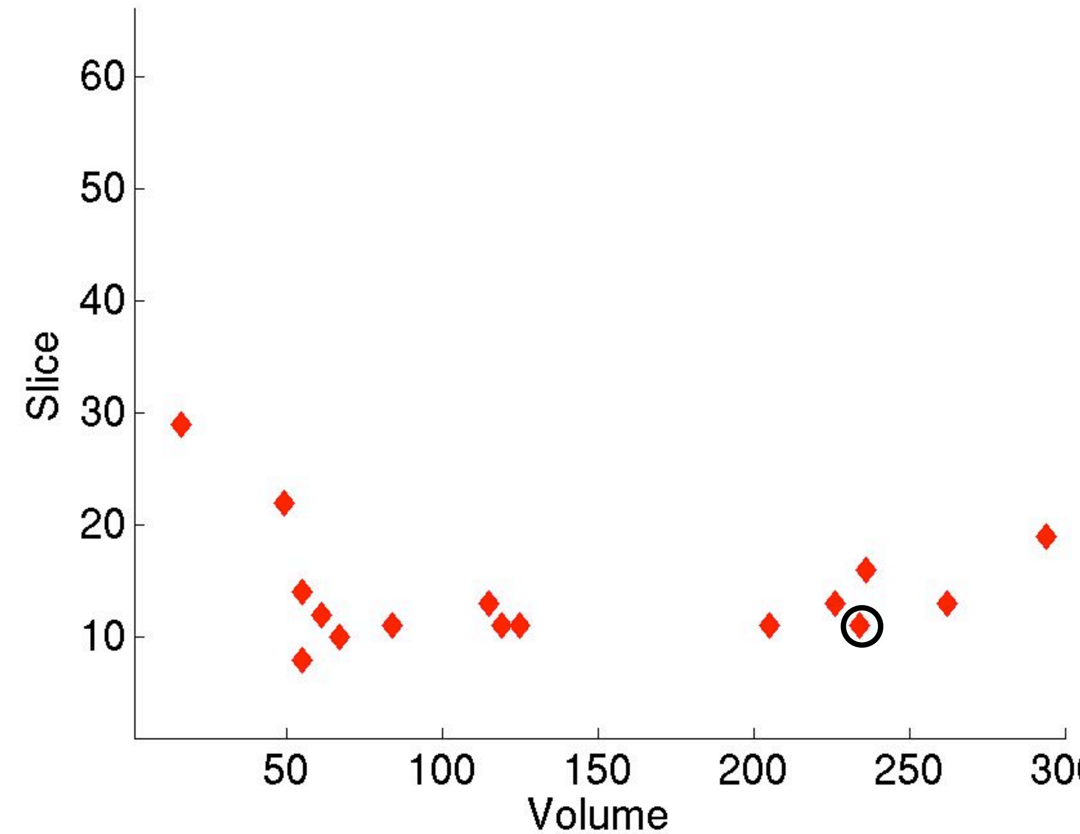
# Outlier detection

Original data



Data after replacement

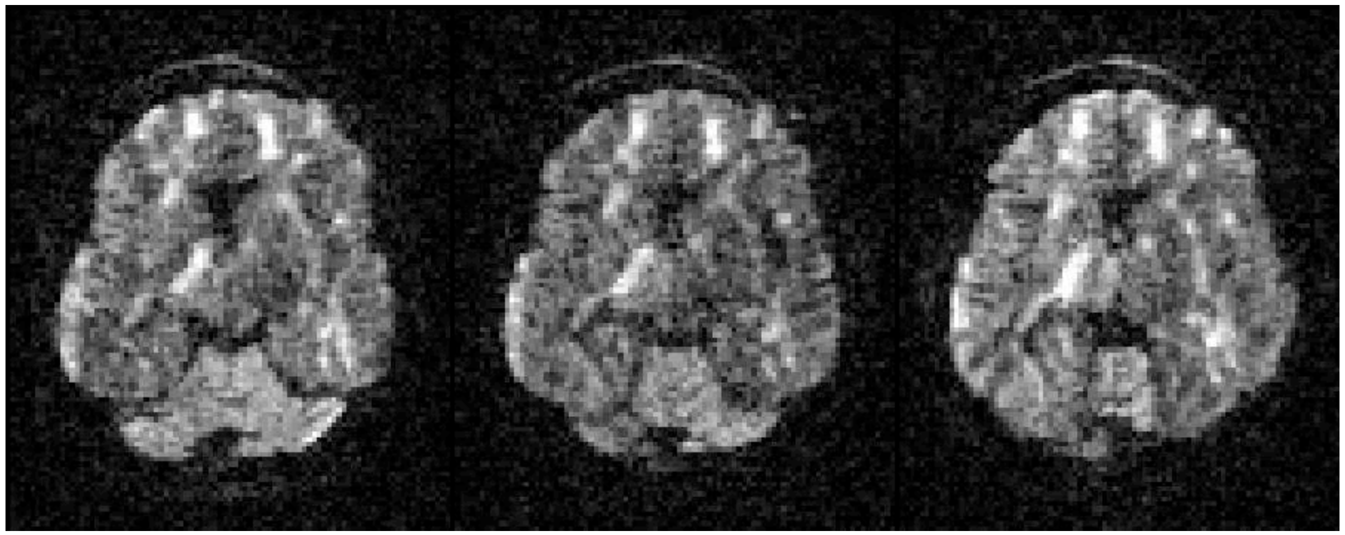




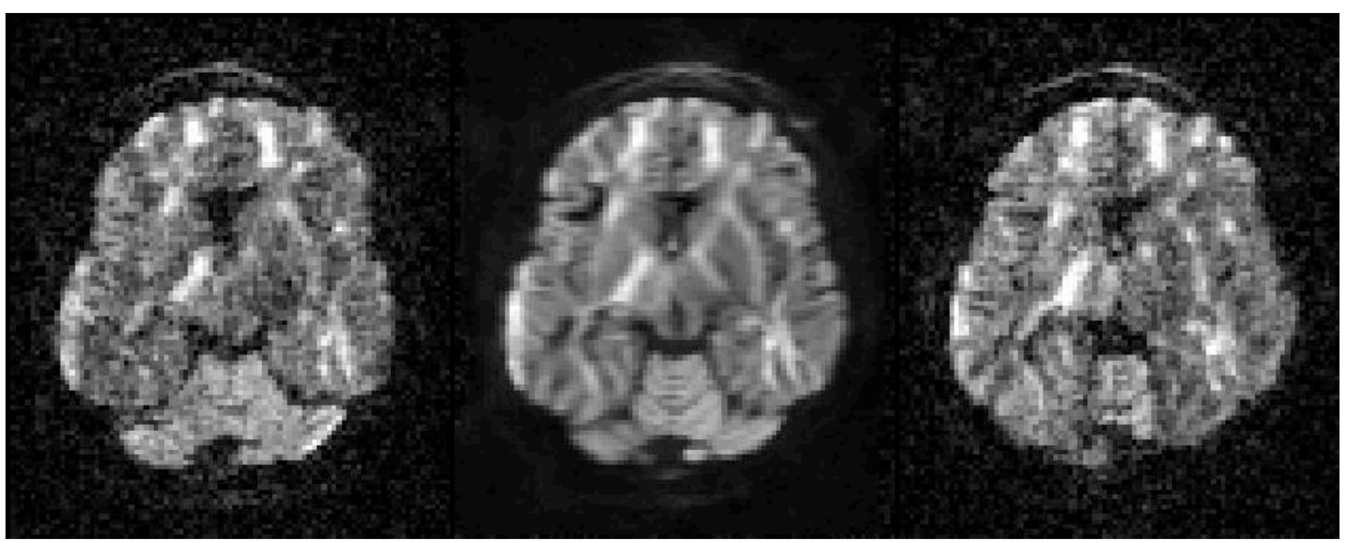

Outliers for a very still volunteer. Outliers mainly in basal slices.

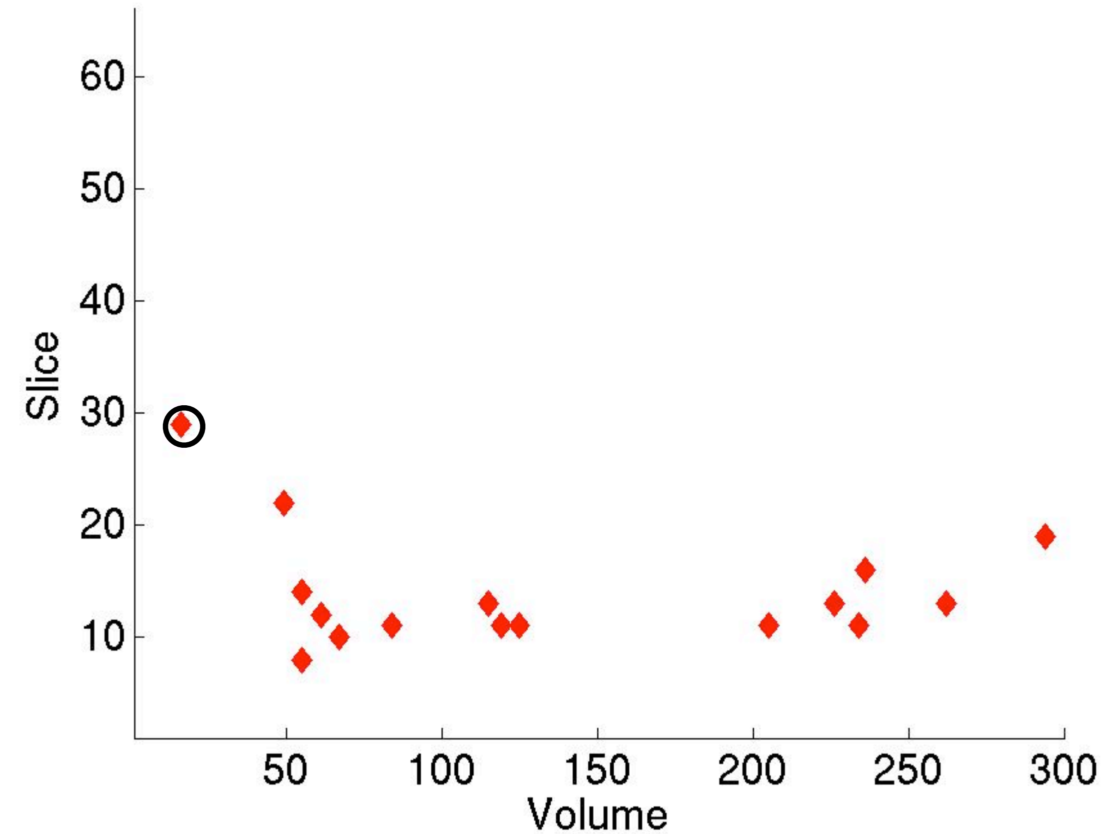
# Outlier detection

## Original data



## Data after replacement



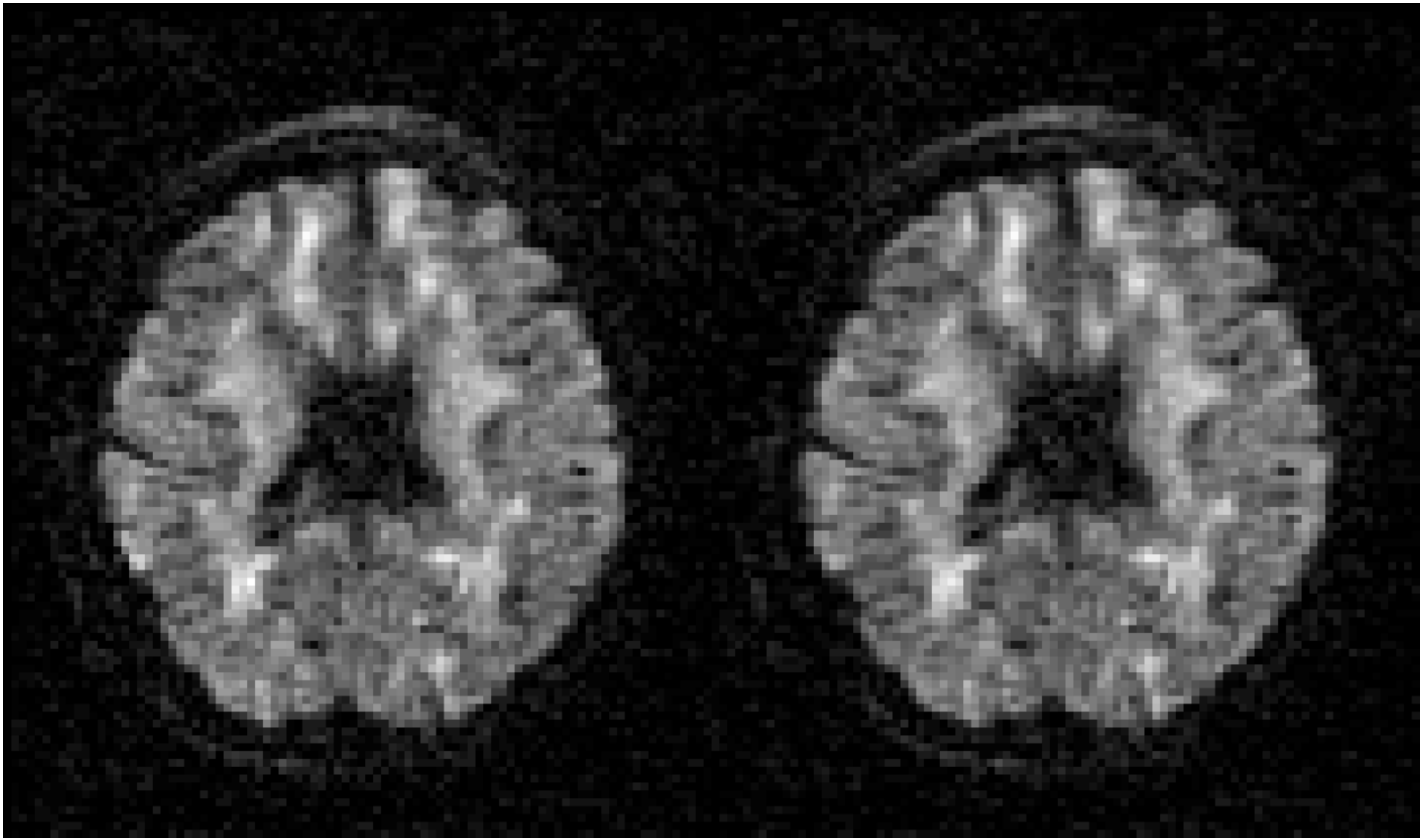



This outlier <u>probably</u> due to subject movement.

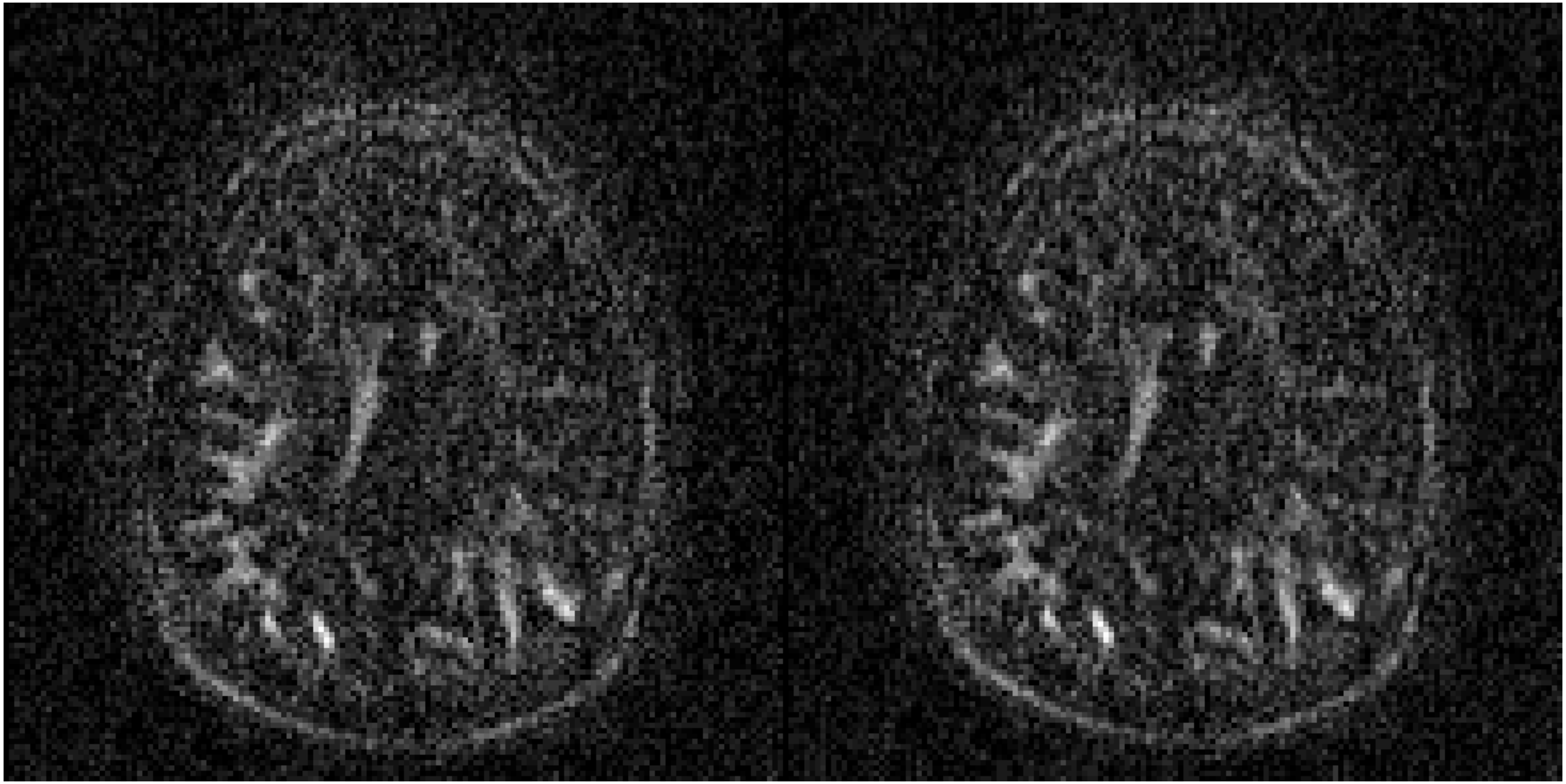
# Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field ⇒ Distortions
- How to fix, and mess up, data
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- Zoltar -- The prediction maker
- How eddy works
- Under the hood of Zoltar
- Outlier detection
- Some results

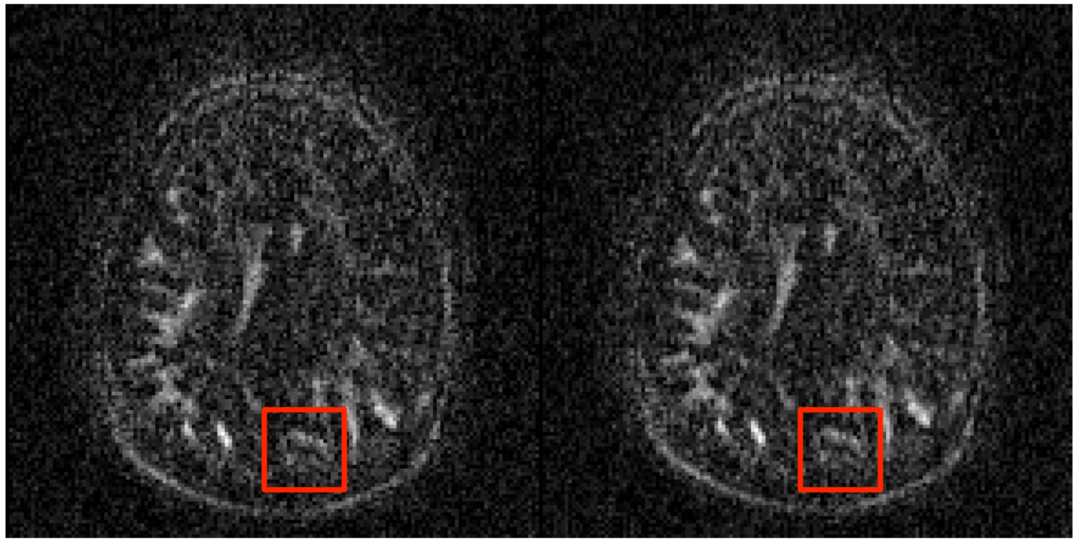
# HCP-data, 150 directions, b=3000, blip-up-blip-down

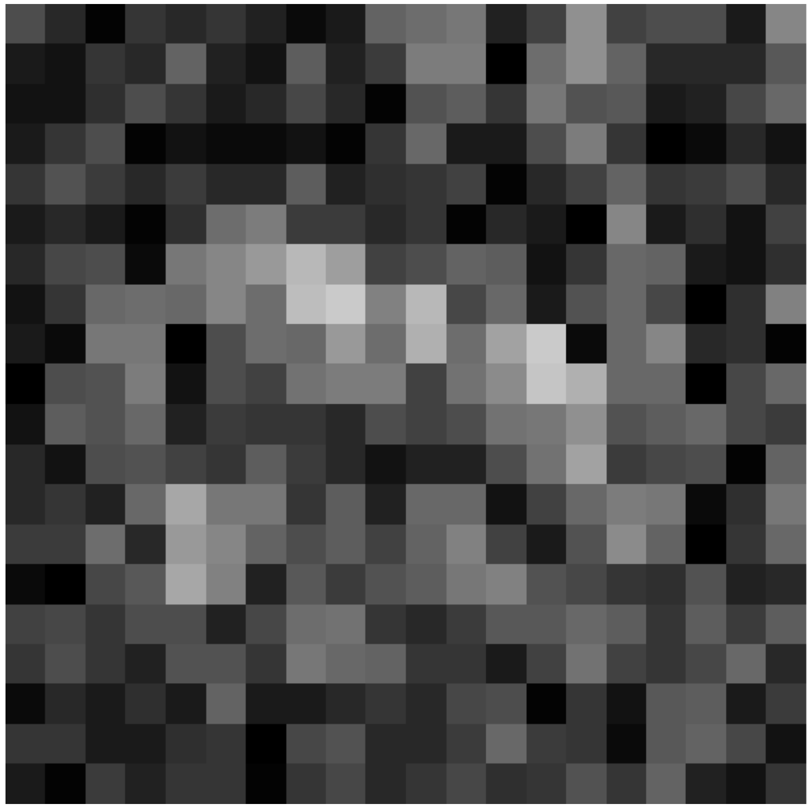


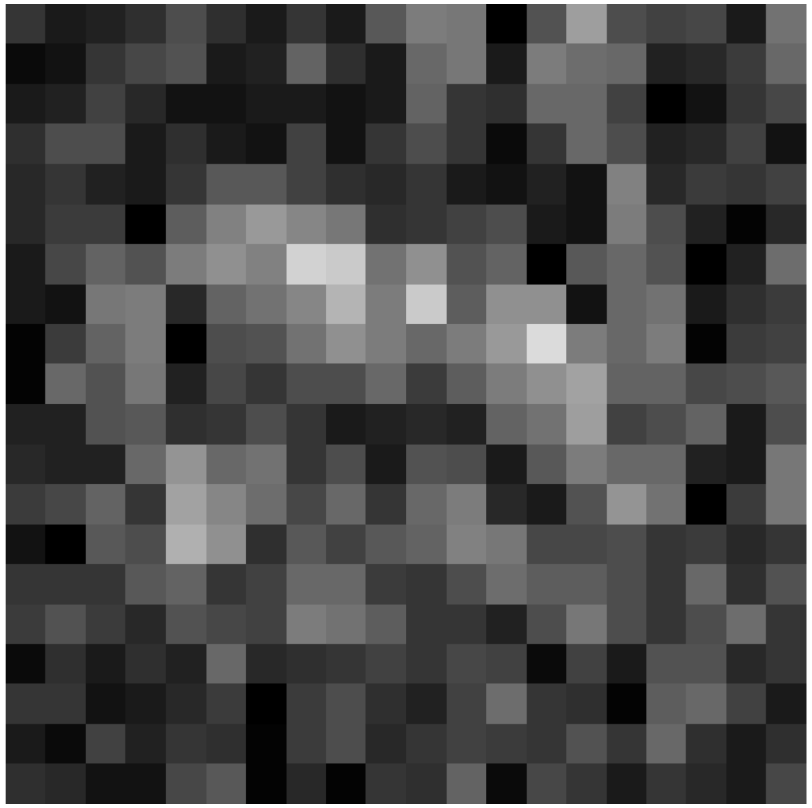
# MGH-data, 198 directions, b=10000!



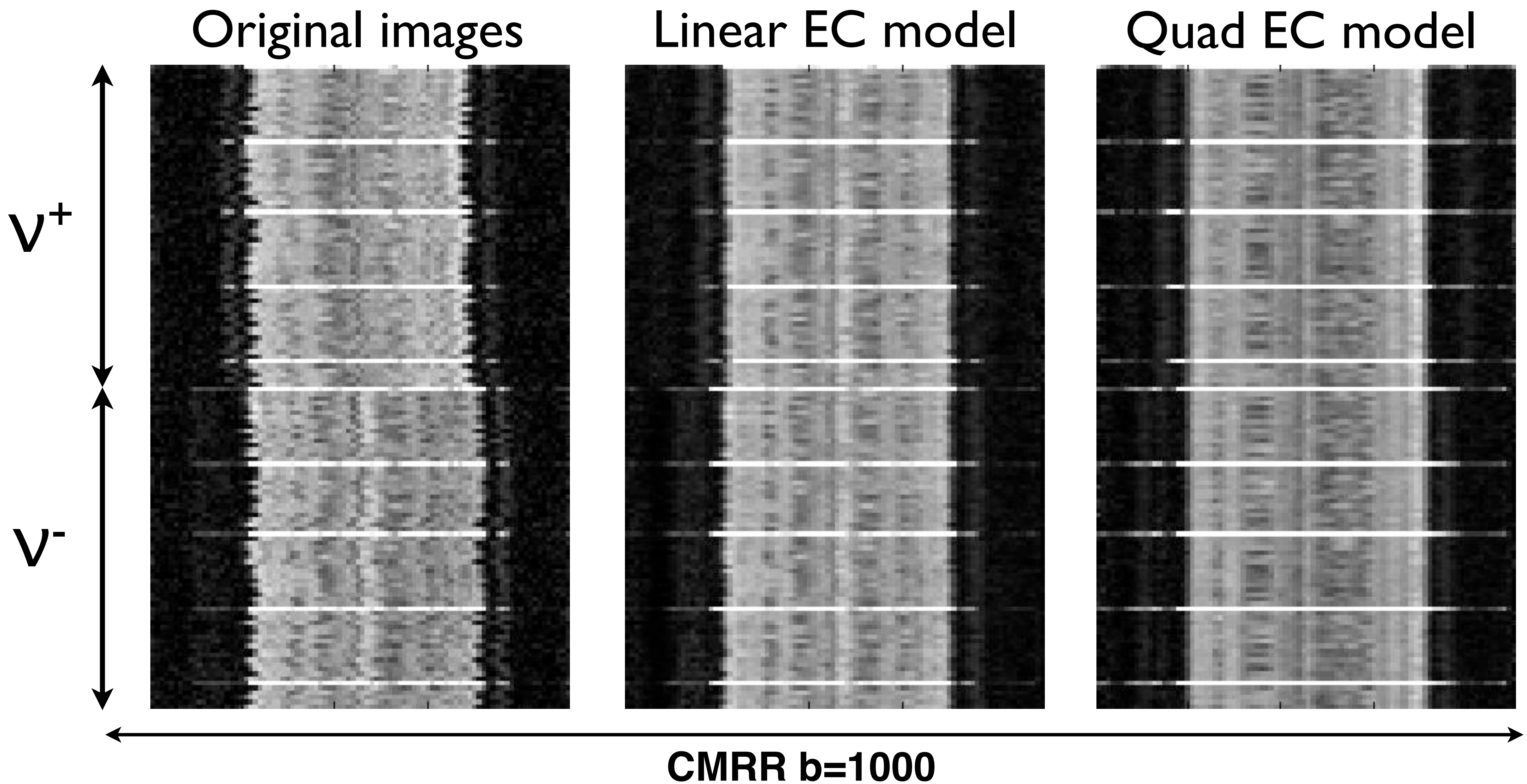
# MGH-data, 198 directions, b=10000!







# Some result in case the movies don't work



# Some result in case the movies don't work