Robust Realignment of fMRI Time Series Data

Ben Dodson
bjdodson@stanford.edu

Olafur Gudmundsson
olafurg@stanford.edu

December 12, 2008

Abstract

FMRI data has become an increasingly popular source for exploring brain activity for a variety of research purposes. Doing so with automated tools requires the series of images to be aligned as accurately as possible, accounting for any motion that may have occurred during the scan. This is typically done by applying a rigid body transformation to each three dimensional image of the series, often done with a Least Squares cost function as a similarity metric between a given figure and the reference image. However, by itself, this technique can be biased by outlying voxel intensities caused by random noise during the scan and, even worse, caused by the very brain activation that is ultimately being detected. Bias in motion estimation due to activation signals can ultimately result in the detection of false activation regions or true activation regions remaining undetected. We apply various techniques of Machine Learning to try and improve upon this algorithm, building on top of common tools when possible.

1 Introduction and approach

The problem of motion correction for fMRI time series is of particular importance for the analysis of brain activation. As described in [2] there are two main issues with this task. On one hand an accurate realignment algorithm is needed so that relatively large motions won't hinder the detection of signal variation due to brain activity. On the other hand if the brain activation is relatively high it can bias the motion estimate and possibly result in false activation regions being detected in later steps of the analysis.

In this paper we will turn most of our attention towards the second issue. This issue is known to be especially severe when using a least-square cost function as a similarity metric between two images.[2] This is because the L_2 norm puts small weight on small residuals but strong weight on large residuals.

We will approach the problem in two ways. First we will investigate whether using image preprocessing techniques to detect features in the image will help ignoring the effect of bias due to brain activation. Specifically we will focus on detecting the edge of the brain image and use that to align all images to the reference. Our second approach will focus on using different cost functions that are less sensitive to outliers and should thus be less sensitive to high activation signals.

2 Data and Simulations

The fMRI data studied is readily represented as a series of three dimensional matrices. Typically a slice of size 64x64 voxels, at a depth of 30 slices, over a period of 40 samples was used. Here, the conversion between voxels and millimetres is a fixed, known quantity. The underlying data at each voxel was a single 16-bit unsigned integer, representing the intensity at that location.

Obviously in a real fMRI image series the true motion and activation of the subject can not be known exactly. Instead, all work was done using simulations. A single, fixed three dimensional fMRI scan was duplicated 40 times, representing a static time series. Then, simulated activation was applied by increasing the intensity of a select grouping of voxels by a given amount ranging from 0-10%. Finally, we altered the orientation of a given image by applying an affine transformation to the original image. To best approximate true motion, a smooth motion function was used and, to simplify our analysis, it was fixed to be a sine wave. Similarly, a square wave was chosen for the injected activation profile over our samples.

We explored various combinations of affine motion over the six degrees of freedom, but to ease analysis, these functions were required to be parametrized by a single value determining the peak motion expressed over our time series. Likewise, our activation had a single parameter determining the amplitude of the square wave applied. The overall method used was similar to that of [6].

Finally, all of our work was done in the open source Matlab-based platform SPM. By doing so, we hoped to both avoid repeating work that others had already done, and also to have a clear channel of distribution in the event that our work turned out useful results. SPM's realignment method was a useful baseline for determining if our injected motion was what we desired.

To investigate different cost functions we used `CVX`, a package for specifying and solving convex programs [3, 4], to solve the resulting convex optimization problems.

3 Derivatives of Least Squares

SPM, like many other initiatives that correct for real-world motion over time, uses the method of least squares to compensate for unwanted movement. Between two given images in the series, the motion is assumed to be affine, and the correction matrix is computed by solving a linear system of equations for the mapping of voxels. However, it is known that least squares is sensitive to outliers. Our first intuition was to mask such outliers.

3.1 Pre-processing of Images

Figure 1(a) shows a typical 2D slice from a 3D fMRI image. It seems apparent that most voxels belonging to the brain are of relatively high intensity compared to the background. This is clearly demonstrated in figure 1(c) which shows a histogram of the pixel intensities in the image. Thus a simple way to separate the background from the brain is to use a clustering algorithm, like K-means with $K = 2$, to find the means of the two distributions. Then one can use the results to mask out the entire brain and do the realignment based on the edge between the two regions.

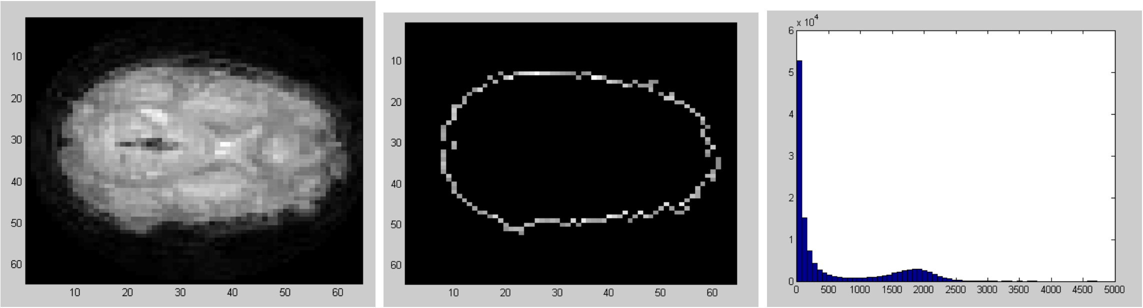
To further elaborate on this idea we also attempted to use higher values for K. By reducing our color space to a much lower dimension, our voxels would be more likely to have an intensity equivalent to its counterpart in the realignment process; the activated regions should be indistinguishable from their unactivated counterparts. With $K = 4$, for example, we see good distinction between regions of our image, and hence improved robustness when considering activation levels. On a particular slice, we found our means to be $[1020.7, 2617.8, 4214.9, 5812.0]$. These clusters are separated beyond the levels of activation produced in the brain, which were at most 10% greater than the original value, so we were likely to assign coordinated voxels to the same cluster whether or not activation was applied.

We discussed several methods of determining clusters, taking into account efficiency and efficacy. Because we would be mapping voxels across slices and across time samples, we required the cluster values to match in each image. And, because our simulations were generated off of a single image, we used one image to determine these values.

We found that the addition of K-Means did achieve our desired result of masking the induced activation; During our error analysis, there was no indication that activation was present in these series. However, the loss of detail in the images resulting from applying K-Means introduced substantial error to our least squares algorithm. Although comparable, our results were typically worse than the standard least squares method.

We explored another intuition for how we could reduce the effect of activation. Rather than applying a lossy filter to each voxel of each image, we could detect a region of the image that would be readily reconciled between images, and would also be immune to activation. The boundary region of the brain was a likely candidate.

We tried several well known edge detection methods, such as Canny and Sobel for our preprocessing. However, during our K-Means analysis, we saw that the method was acceptable for separating regions of the



(a) An unaltered slice taken from an fMRI series.

(b) Simple boundary detection based on K-Means using 2 clusters.

(c) Histogram of pixel intensities in a single fMRI scan.

brain from the background, and built a simple boundary detector from there.

4 Different cost functions

The role of the cost function is to assign a value to how well a given image matches to the reference image given a certain transformation. The most common choice of a cost function in practice, and the one that is implemented in SPM, is the L_2 norm. Other currently used methods are the correlation between images [1] or Mutual Information [5]. For the purpose of this paper we consider using the L_1 norm and the deadzone-linear penalty function instead of least-squares. The deadzone-linear penalty function puts zero cost on residuals less than a given value M (which we chose to be 0.04) and is defined as:

$$\phi(x) = \begin{cases} 0 & , x \leq 0 \\ |x| - M & , x > 0 \end{cases}$$

Both of them put relatively small weight on large residuals and should thus be less sensitive than least-squares to bias due to activation.

5 Results

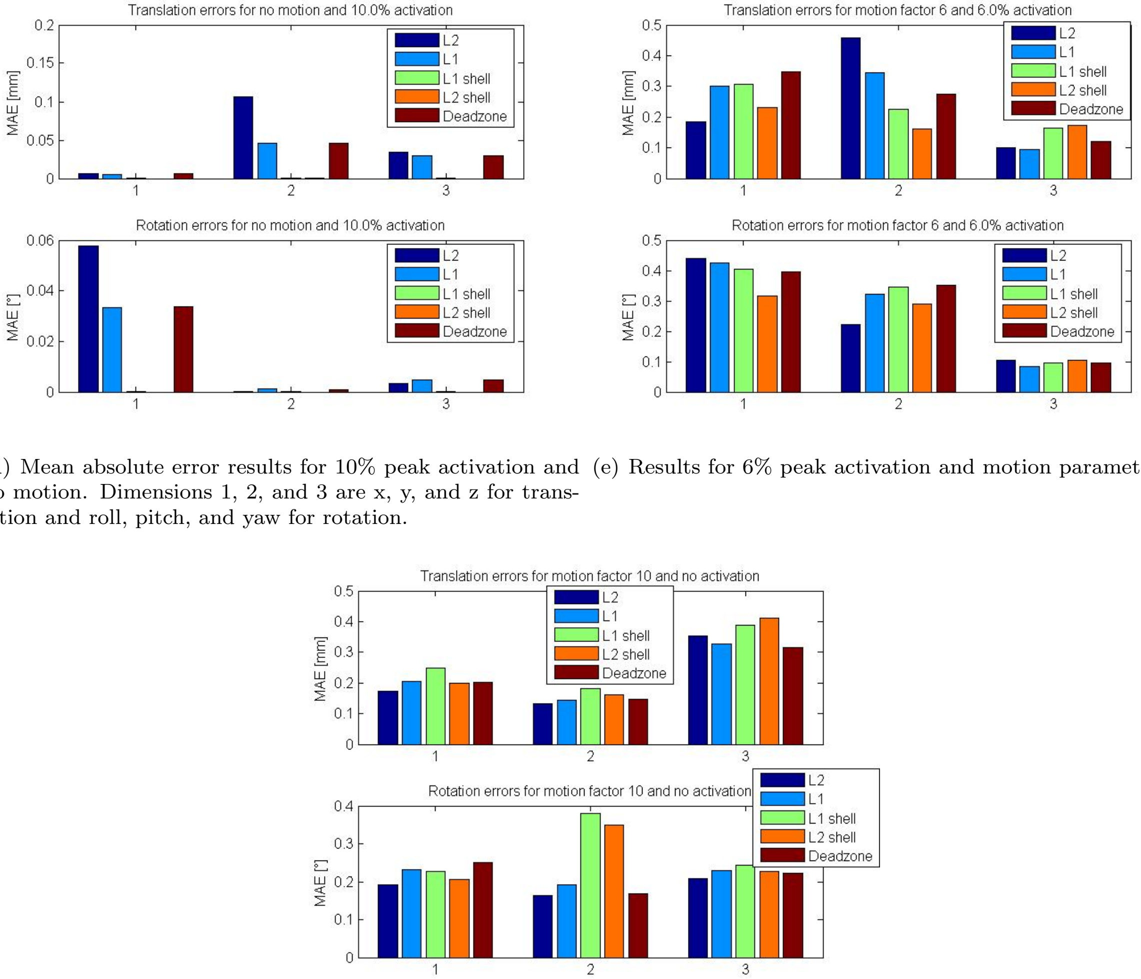
Figure 5 shows the results from using all methods on three different cases: little or no motion injected and high activation (10% peak activation), medium motion and activation (motion factor 6 and peak activation 6%) and a case with high motion but little or no activation (motion factor 10). The motion factor defines the amplitude of the sine wave motion function injected, thus a motion factor 10 injects a translation of 5 mm, 2.5 mm and 1 mm in x, y, and z direction and 5° and 2.5° rotation around the x and y axis.

From 1(d) we see that using the L_1 norm or the deadzone function as a cost function results in less sensitivity to high activation compared to using the L_2 norm. Furthermore, the boundary detection methods are more or less immune to the activation as expected since we are masking out the activated brain regions. For the two cases with higher motion, where bias due to activation is less critical, all methods are mostly comparable.

While our methods are not perfect, they do provide superior results for the common case where the effect of activation may outweigh that of motion. With a real time series, it is not possible to determine whether or not motion is an issue, and our methods protect against introducing bias due to a false correction.

6 Acknowledgments

Many thanks to Catie Chang for her inspiration and continual support on this project. Also thanks to Andrew Ng and the many TAs of CS229 for guidance.



(d) Mean absolute error results for 10% peak activation and no motion. Dimensions 1, 2, and 3 are x, y, and z for translation and roll, pitch, and yaw for rotation.

(e) Results for 6% peak activation and motion parameter 6.

(f) Results for no activation and a motion parameter of 10.

References

[1] Collins DL, Evans AC, Holmes C, Peters TM. *Automatic 3D segmentation of neuro-anatomical structures from MRI.* Inform Process Med Imaging 1995;139-152

[2] Luis Freire, Jeff Orchard, Mark Jenkinson, and Jean-Francois Mangin. *Reducing Activation-Related Bias in FMRI Registration.* MIAR 2004, LNCS, pp. 278-285.

[3] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming (web page and software). `http://stanford.edu/~boyd/cvx`, December 2008.

[4] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[5] Maes F, Vandermeulen D, Suetens P. *Comparative evaluation of multiresolution optimization strategies fro multimodality image registration by maximization of mutual information.* Med Image Anal 1999;4:373-386.

[6] Jeff Orchard, Chen Greif, Gene Golub, Bruce Bjornson, and M. Stella Atkins, *Simultaneous Registration and Activation Detection for fMRI.* IEE Transactions on Medical Imaging, Vol. 21, No. 5 May 2002;470-484.