

Motion Correction and Despike Functions

Paul Mazaika, June 11, 2009

The basic preprocessing pipeline contains two new methods: Motion Adjustment and Artifact Repair. The pipeline separates the motion adjustment and artifact removal algorithms into the preprocessing before GLM estimation, thus, motion regressors are not needed in the design matrix. The intermediate data can be reviewed at any step using the Contrast Movie, the RMS image, or plots of single voxel timeseries. The steps are implemented in separate modules to easily review the performance of each step of the algorithm.

This document describes the Motion Adjustment, Despike and Filter, and RMS image functions. Artifact Repair and Contrast Movie are described in the ArtRepairInstructions guide.

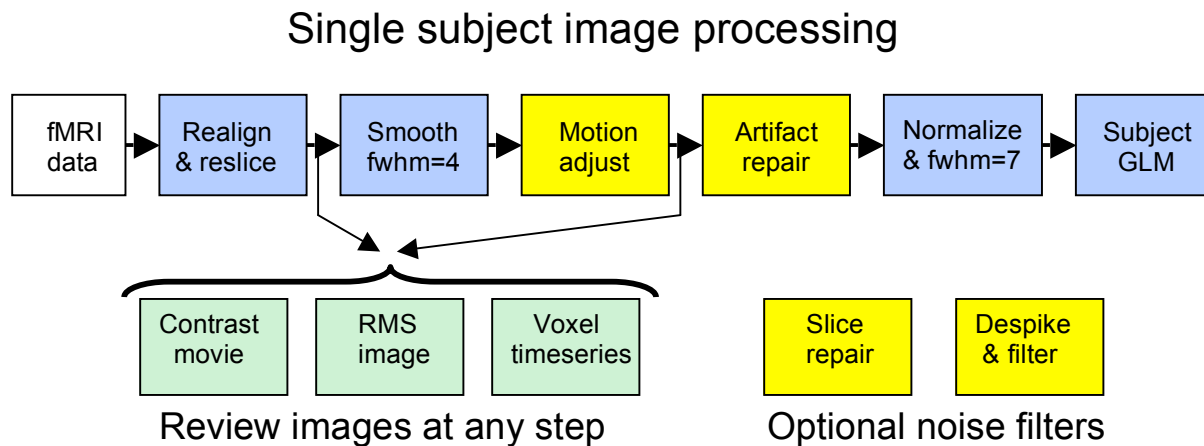


Figure 1: The pipeline uses single subject smoothing of FWHM = 4 mm, which helps the motion adjust algorithm. The best single subject result is after Artifact Repair. The normalized images are smoothed by 7 mm to merge results across subjects. The two smoothing operations of 4 mm and 7 mm are about equivalent to one 8 mm group smoothing. The outputs of any step can be reviewed by a movie and RMS summary.

Possible alternate pipelines include:

- Use Artifact Repair on the smoothed normalized images just before GLM estimation, and add motion regressors to the design matrix.
- Use Despike and Filter after Motion Adjustment and Artifact Repair, just before normalization.
- Use the slice repair (>>art_slice) program on the raw images before slice time and realignment. This algorithm can be effective when there is large transient electronic scanner noise.

Motion Adjustment

Motion adjustment is an alternative to adding motion regressors to the design matrix. This algorithm suppresses the residual fluctuations due to interpolation errors from large motions, and it applies stronger correction to edge voxels than to central voxels. The idea is that the linear and quadratic approximations used in motion regressors may not be sufficiently accurate for the large motions of clinical subjects. Note the pipeline removes the residual errors before any cognitive estimate because the effects of the motions can be far larger than cognitive BOLD effects. On the other hand, the estimates of the cognitive signals may be biased in cases of task-correlated motion because the motion is removed before the estimation step. The assumption of the method is that it is safer to remove the potentially large motion effects before the data enters into the estimation process.

Usage: >> art_motionregress

This function is applied to realigned, resliced images. The function works better on slightly smoothed images with FWHM = 4mm used as inputs. The algorithm runs somewhat faster than realignment itself. WARNING! The program will crash or run very slowly on normalized images which are usually much larger in size.

USER INPUTS

Select realigned and resliced images, eg. 'rI*', or after smoothing 'srI*'.
Select realigned, unresliced images, e.g. 'I*img' where there

are associated .mat files. The .mat files describe the realignment results, and are used for the regression calculation.

OUTPUT

Writes new image files with prefix 'm', that are the corrections of the input images after motion adjustment.

Also writes some information images:

Writes maprior image, showing logarithm of regularization value.

Writes six mgamma images files containing motion regressors.

Writes file art_suspects.txt listing files omitted during calculation of the regression parameters.

See Figure 4 as an example of motion adjustment reducing time series variation.

Background

It was previously shown that a trigonometric form adjustment could correct for mathematical interpolation errors when there was a poor interpolation algorithm (Groontoonk, et.al. 2000). In separate experiments, we confirmed that the same interpolation error is present even with the best interpolation algorithms, and that it could be well-estimated and corrected with a similar periodic form (Mazaika, et.al. HBM2007).

The algorithm first approximates the translations and rotations of the whole head as three translational distances locally on each voxel. The distances are expressed relative to voxel size, so a translational distance of x mm in the X -direction is expressed as $x' = 2*\pi*x/v_x$ where v_x is the voxel size in the X -dimension. Translations in the Y and Z directions are expressed similarly. The voxel time-series \mathbf{u} after realignment and reslicing is an $nx1$ vector, where n is the number of images in the scan session. The motion correction estimator, \mathbf{g} , a $7x1$ vector, is the linear regression solution to the equation:

$$\mathbf{u} = \mathbf{M}\mathbf{g}$$

where \mathbf{M} is the $nx7$ matrix with i th row:

$$\mathbf{M}_i = [\sin x_i' \quad 1-\cos x_i' \quad \sin y_i' \quad 1-\cos y_i' \quad \sin z_i' \quad 1-\cos z_i' \quad 1]$$

and x_i' , y_i' , and z_i' may be different for every image of data, and different for every voxel. The motion correction estimator \mathbf{g} is calculated from the “normal equation”,

$$\mathbf{g} = (\mathbf{D}^2 + \mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{u}$$

For accuracy, the matrix $\mathbf{M}^T\mathbf{M}$ is regularized by a parameter \mathbf{D}^2 that depends on the observed RMS variation on a voxel. The regularization value is small (~ 0.003) for edge voxels to strongly suppress motion-induced fluctuations, and large (~ 5) for central voxels where motion-induced errors are small. Images (named `maprior` and `mgamma`) are created showing the regression solution (see Figure 2). The motion corrected timeseries \mathbf{v} on a voxel, an $nx1$ vector, is calculated from

$$\mathbf{v} = \mathbf{u} - \mathbf{M}\mathbf{g} + u_0$$

where the matrix \mathbf{M} is as before, and u_0 is the constant term of the estimate. All new images are written for use by succeeding stages of fMRI processing.

Scans with high scan-to-scan motion are excluded from the regression estimation, although the regression result is applied to all scans. It is assumed these high motion scans will be fixed later by Artifact Repair or Despiking. The method is safe to use on low motion subjects, because the first six columns of \mathbf{M} are near zero, and the motion correction smoothly transitions to an estimate of the mean signal.

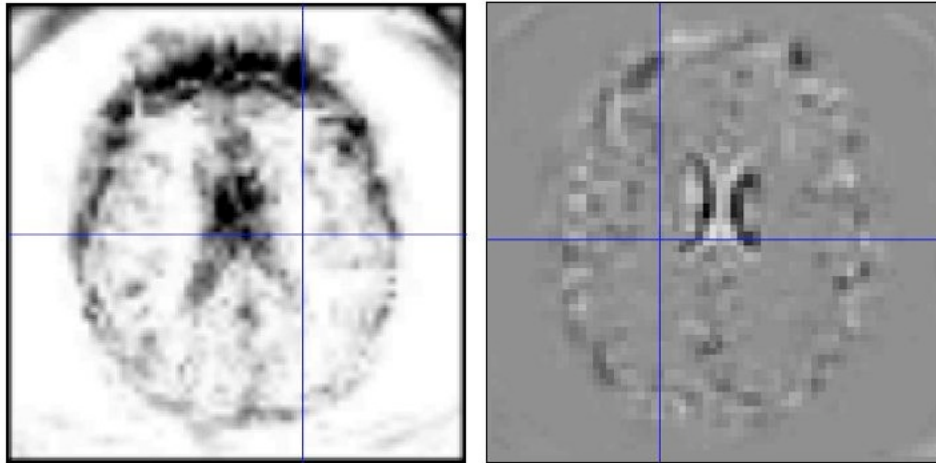


Figure 2: LEFT: maprior image showing log of regularization factor. Values range from -5.6 (black) to 1.8 (white). Motion reduction is stronger on brain edges and near ventricles. Data in the corner of the images is an artifact of a spiral scan pulse sequence. RIGHT: mgamma05 image shows coefficient of sin z term of g for all voxels (on a different brain). Z-correction is strong on the superior edge of ventricles, and can vary sign on neighboring voxels.

RMS Image

Computes the RMS fluctuation image for a set of images, for all voxels within a head mask. Writes out the RMS image. Summarizes the result as average RMS fluctuation over the mask in percent signal change, as well as the mean RMS and mean image over the mask. Percent signal change is measured relative to the mean of the image within a head mask.

Usage: `>> art_rms`

USER INPUTS

Images is a list of images, in a single session.

MaskImage. Specified by user or generated automatically.

OUTPUTS

RMS image, named rmsINPUT.img, where INPUT are the input image names. No percent scaling is done on this image.

Values returned:

Avg RMS variation over mask in percent signal change
 $= 100 * (\text{avg RMS variation} / \text{Imagemean})$

Avg RMS variation over mask, and mean of all images over mask

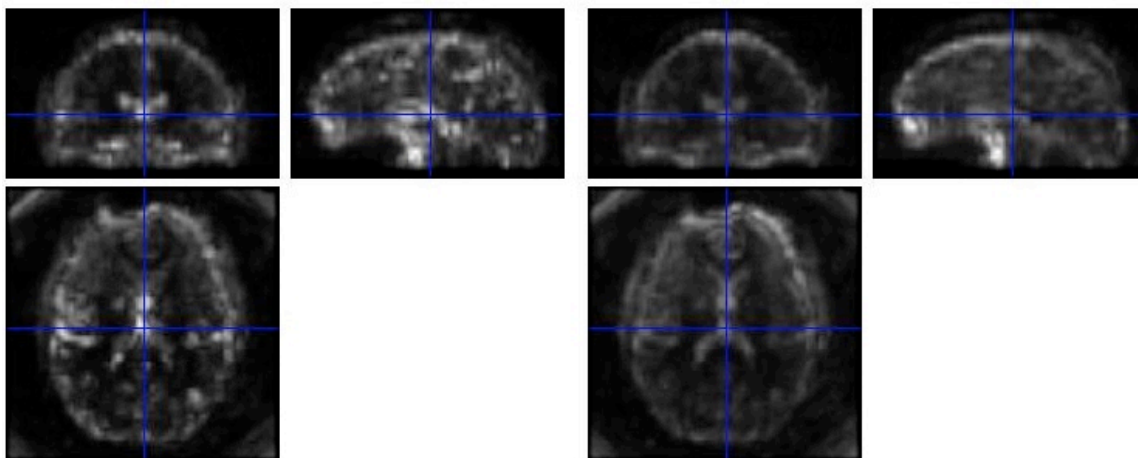


Figure 3: The RMS image of the smoothed realigned images ('srI') on LEFT, and motion adjusted smoothed realigned images ('msrI') on RIGHT. Mean RMS over image is reduced from 3.98% to 2.51% after motion adjustment.

Despike and Filter

Removes spikes and slow variations using clipping and a digital high pass filter. This filtering can reduce the “ringing” side effects that may occur when standard high pass filters encounter large spikes in the data. Generally, despiking and tight digital filters give more robust processing in large noise at the expense of slightly reducing the contrast between conditions and not being as optimal in low noise conditions.

The DESPIKE function clips all values further than DESPIKE % from a rolling mean to exactly a DESPIKE % deviation. For example, if DESPIKE = 4, a spike of size 6% will be reduced to size of 4%. Percent signal change is measured relative to the mean of the image within a head mask, and is the same for all voxels. Despiking is performed voxel-wise, since spikes may occur only on some voxels. Set DESPIKE = 0 to not apply this function (i.e., only apply filtering).

The high pass FILTER removes slow variations on each voxel with an AC-coupled filter, and adds the mean input image to write the output images. There are choices of two high pass filters, a smoothing and high pass filter, or none at all.

If both despike and filtering are chosen, images will be despiked before entering the high pass filter.

Usage: >> art_despike

WARNING! FOR UNNORMALIZED IMAGES ONLY. The large size of normalized images may cause this program to crash or go very slow.

USER INPUTS

Images is a list of images, in a single session.

Images must all have the same orientation and size.

FiltType has 4 options:

1. 17-tap filter, aggressive high pass filter.
Suitable for ER designs in noisy data.
2. 37-tap filter, long filter for block designs. SPM high pass filter may be better since it preserves gain.
3. No high pass filtering is done. Despiking is done based on a 17-point moving average of unfiltered data.
4. Matched filter for single events using temporal smoothing.
Perhaps useful for movies of ER fMRI

Clip Threshold is used to despike the data, in units of percentage signal change computed relative to the mean image.

If Despike = 0, no clipping is done. Despike=4 is the default.

OUTPUTS

Images with a prefix "d" (for despike or detrend) in the same directory.

The filtered images have the same mean as the original series.

Mean input image, named meenINPUT.img, where INPUT is input image name. The misspelling of mean is deliberately different from the mean image produced by SPM realignment.

The algorithm runs through all the images sequentially like a pipeline. It clips and filters about 1 volume per second with default settings. The ends of the dataset are padded by reflection for the filter process. SPM high-pass filtering is OK, but not required after this filter. Note these filters are applied to only the data, so there is a filter gain to try to preserve quantitative amplitude information. This works OK for ER designs, but not as well for block designs. The SPM high pass filter applies to both the data and design matrix, and may be better for block designs.

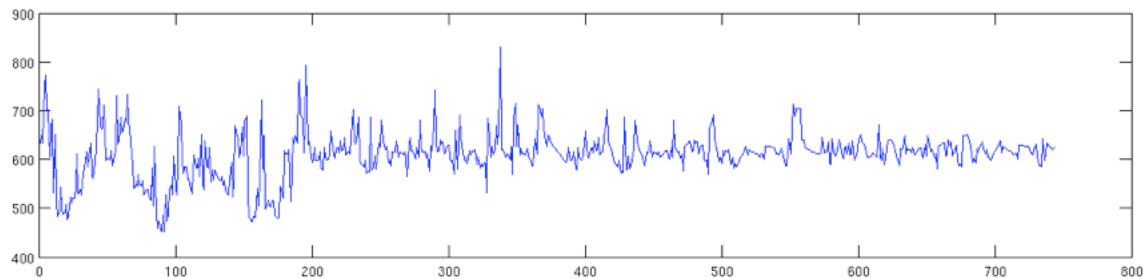


Figure 4: Output of `>> art_plottimeseries` of a single voxel (near the edge of L-DLPFC) concatenated over four processing steps. Scans 1-186 are realigned, resliced 'srI' images. Scans 187-372 are motion adjusted 'msrI' images. Scans 373-558 are artifact repaired 'vmsrI' images. Scans 559-744 are despiked images ('dvmsrI') with 17-tap filter and clip threshold of 4%. Each step removes more variation from the realigned time series.