Introduction

The CheckFrame scripts are a quality checking tool for 4D fMRI images saved in the nifti format. They are intended to find frame outliers in fully preprocessed scans. Here frame refers to one three-dimensional volume in a time series of fMRI images. The scripts are written in Matlab and include a template and central file like other MethodsCore scripts. The user only needs to set the variable values in the template file to run CheckFrame.

Template File

In the template file, each variable includes a description of its purpose so it is self-explanatory. Most of the variables used in the template script are used to create the file path to the images. Only 4D nifti files can be used. The variable **Opt.OutlierText** refers to a text file that is generated which contains all scans labeled as outliers. Detecting outliers is controlled by the variable **Opt.Thresh**. This refers to the number of standard deviations beyond which an observed value is declared to be an outlier. The recommended value is either 3 or 4. A lower threshold value causes more frames to be labeled as outliers. The user is responsible for determining whether the frame is an actual outlier.

Ouput

In each run directory where the 4D nifti file is located, a pdf file is generated named frameReport.pdf. The report graphically displays calculated metrics for the nifti file in five figures. Table 1 summarizes what is plotted in each figure and example figure are given in figure 1-4. If any frames are labeled as outliers, a csv file is also generated in the run directory. The csv file contains regressors that can be used in the first level analysis to account for the variability caused by the outlier frames. The regressors are set equal to one at the outlier frames and zero for the remaining frames.

Table 1. frameReport.pdf summary

Figure Title	Metric	Utility
Frame Mean Values	mean voxel value for frames	The user can compare mean voxel values between frames.
Z scored frame means	mean frame values z-scored over time	This metric can help identify frame outliers.
Standardized diff between frame means	difference between mean frame values z-scored over time	This metrice is used to identify frame outliers.
MSE between frames	mean squared error between frames across time	This metric can help identify scan outliers.

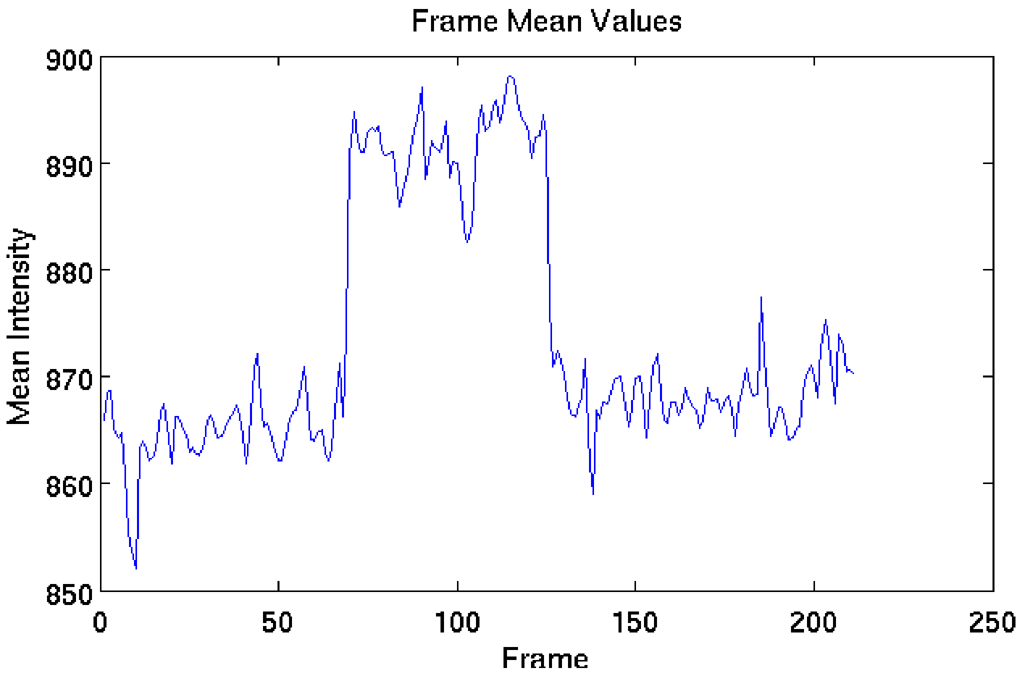


Figure 1. An example scan mean values figures. The mean voxel intensity should not change greatly between frames.

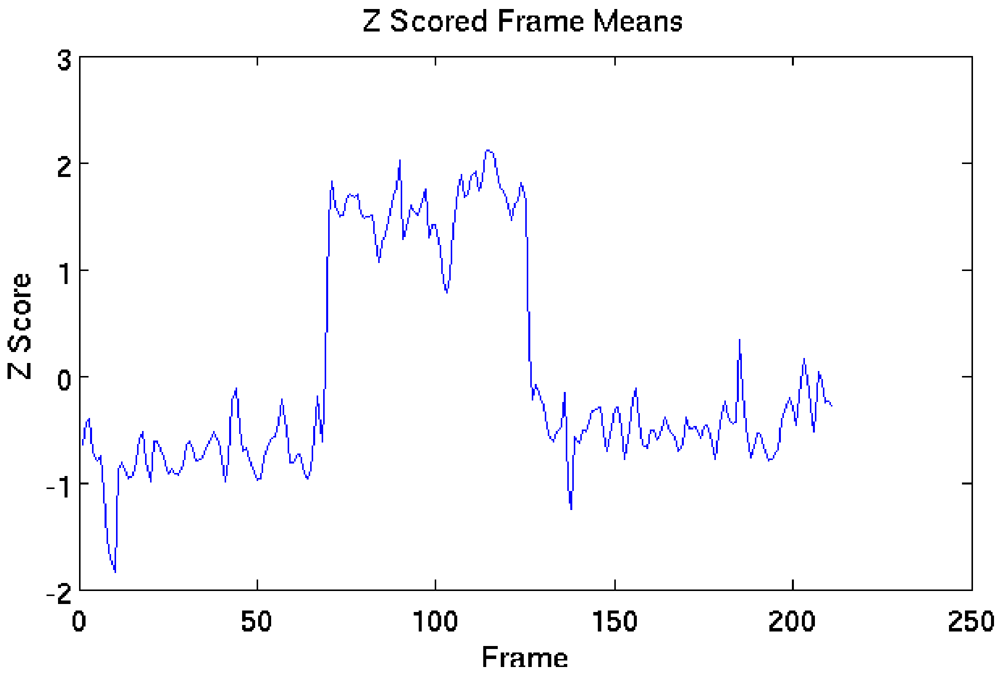


Figure 2. An example z-scored frame means figure. This can help identify outlying frames in a time series.

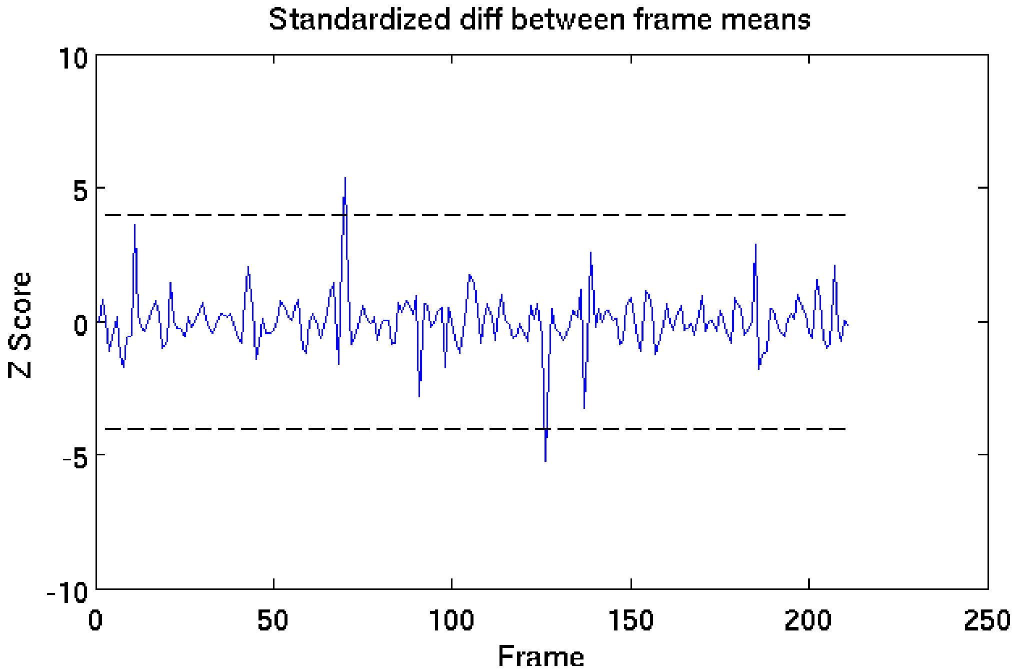


Figure 3. An example standardized difference between frame means figure. The dashed indicates the threshold value for outlying frames.

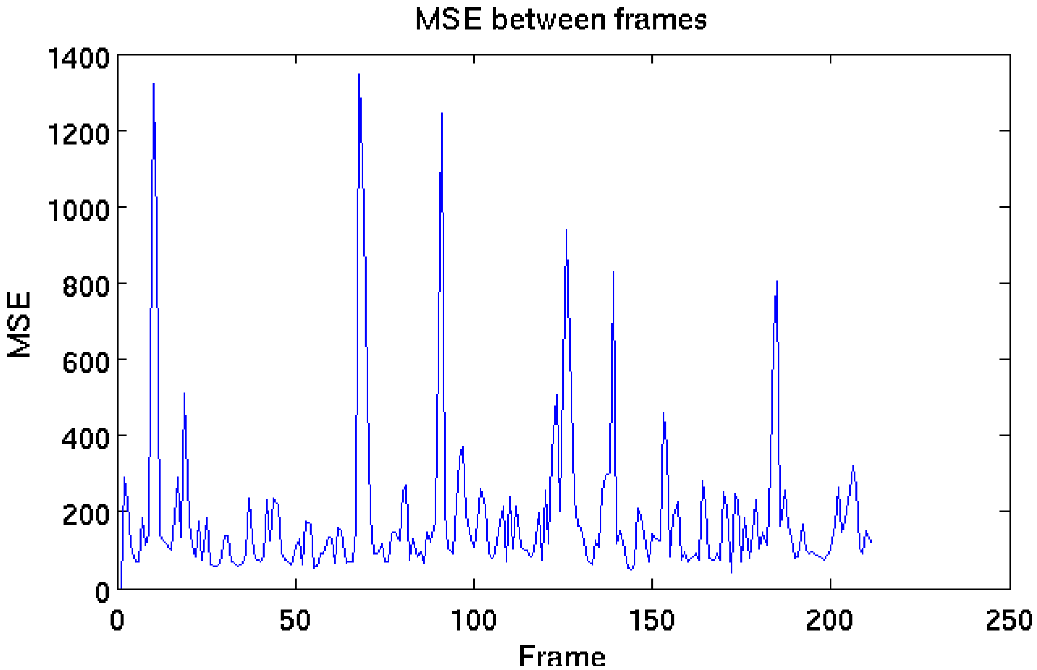
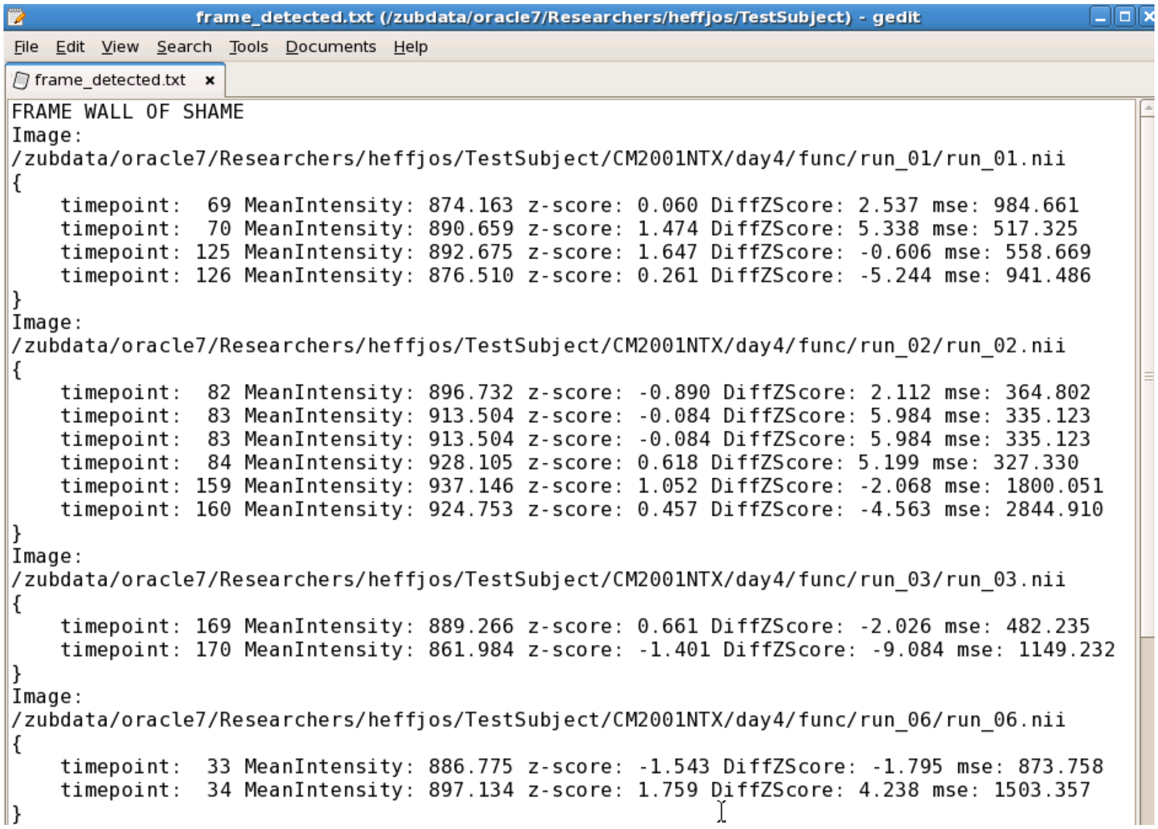


Figure 4. An example MSE between frame figure. This can help identify scan outliers in a time series. Large spikes indicate possible frame outliers.

The script also generates a text file which contains all outlying frames found during a run of the CheckScan scripts. The frame locations are included in the text file and are written in SPM voxel coordinates. All metrics calculated for a frame are saved in the text file to help the user decide what to do with the frame. An example output text file is displayed in figure 5.


frame_detected.txt (/zubdata/oracle7/Researchers/heffjos/TestSubject) - gedit
File Edit View Search Tools Documents Help
frame_detected.txt

```
FRAME WALL OF SHAME
Image:
/zubdata/oracle7/Researchers/heffjos/TestSubject/CM2001NTX/day4/func/run_01/run_01.nii
{
    timepoint:  69 MeanIntensity: 874.163 z-score: 0.060 DiffZScore: 2.537 mse: 984.661
    timepoint:  70 MeanIntensity: 890.659 z-score: 1.474 DiffZScore: 5.338 mse: 517.325
    timepoint: 125 MeanIntensity: 892.675 z-score: 1.647 DiffZScore: -0.606 mse: 558.669
    timepoint: 126 MeanIntensity: 876.510 z-score: 0.261 DiffZScore: -5.244 mse: 941.486
}
Image:
/zubdata/oracle7/Researchers/heffjos/TestSubject/CM2001NTX/day4/func/run_02/run_02.nii
{
    timepoint:  82 MeanIntensity: 896.732 z-score: -0.890 DiffZScore: 2.112 mse: 364.802
    timepoint:  83 MeanIntensity: 913.504 z-score: -0.084 DiffZScore: 5.984 mse: 335.123
    timepoint:  83 MeanIntensity: 913.504 z-score: -0.084 DiffZScore: 5.984 mse: 335.123
    timepoint:  84 MeanIntensity: 928.105 z-score: 0.618 DiffZScore: 5.199 mse: 327.330
    timepoint: 159 MeanIntensity: 937.146 z-score: 1.052 DiffZScore: -2.068 mse: 1800.051
    timepoint: 160 MeanIntensity: 924.753 z-score: 0.457 DiffZScore: -4.563 mse: 2844.910
}
Image:
/zubdata/oracle7/Researchers/heffjos/TestSubject/CM2001NTX/day4/func/run_03/run_03.nii
{
    timepoint: 169 MeanIntensity: 889.266 z-score: 0.661 DiffZScore: -2.026 mse: 482.235
    timepoint: 170 MeanIntensity: 861.984 z-score: -1.401 DiffZScore: -9.084 mse: 1149.232
}
Image:
/zubdata/oracle7/Researchers/heffjos/TestSubject/CM2001NTX/day4/func/run_06/run_06.nii
{
    timepoint:  33 MeanIntensity: 886.775 z-score: -1.543 DiffZScore: -1.795 mse: 873.758
    timepoint:  34 MeanIntensity: 897.134 z-score: 1.759 DiffZScore: 4.238 mse: 1503.357
}
```

Figure 5. Sample output text file that contains outlier frames. Each image with frame outliers is written. The outlier locations and calculated metrics for the frame follow the image name within the brackets.

The following video demonstrates how to use the CheckFrame toolbox:

http://youtu.be/gwaVL5nlkZ0