Real Time Motion Monitoring During a Functional MRI

WRIGHT STATE UNIVERSITY
Biomedical Engineering
Industrial and Systems Engineering

Diana Johnson, Samuel Manuszak, Kareem Khasawneh

DEPARTMENT OF NEUROSCIENCE, CELL BIOLOGY, & PHYSIOLOGY

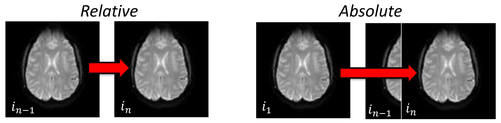
Background

Patient motion during Functional MRI scans at Wright State University's MRI research facility causes inconsistencies in image data, often leading to rescanning and unusable data. The current motion plots are small and difficult to read. An efficient way to monitor patient motion in real time would give MRI staff the ability to react to excessive motion and eliminate the need for rescanning.

Current system output of real-time patient highlighting motion plots

Problem Statement

An algorithm developed in Python will measure patient movement using Framewise Displacement (FD). The absolute and relative FD can then be plotted in real-time on a microprocessor-based visual display.



Relative Motion: Most recent (i_n) vs. previous MRI image (i_{n-1})
Absolute Motion: Most recent (i_n) vs. first MRI image (i_1)

Materials and Methods

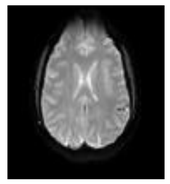
- GUI Framework: Developed using PyQt5 for real-time visualization
- Microprocessor: Raspberry Pi 5 for data processing
- Display: 7" Raspberry Pi screen (1024x600 resolution) for GUI output
- Drin Data Dumper: Philips Research tool used under WSU-Philips Health Research Agreement
- Networking: 50 ft LAN cable for data transmission
- Engineering Standard: Follows ISO/IEC/IEEE 14764 for software maintainability and lifecycle support

7" Display Screen

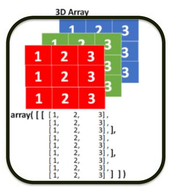
Raspberry Pi 5

Calculating Patient Motion

Drin Tool



MRI Image Data



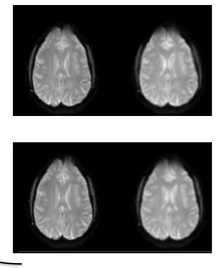
Convert to 3D NumPy array

Convert to 3D NiBabel Image

NiBabel image is stored in RAM to reduce I/O overhead and improve processing speed

FD is the sum of absolute value differences between consecutive MRI frames across six degrees of freedom (three for translation and three for rotation)

Slight changes in motion between consecutive MRI scans



FD Calculation:

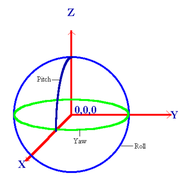
$$FD_i = |\Delta d_{ix}| + |\Delta d_{iy}| + |\Delta d_{iz}| + |\Delta \alpha_i| + |\Delta \beta_i| + |\Delta \gamma_i|$$

Where…

dx, dy, dz = translation (x, y, z)

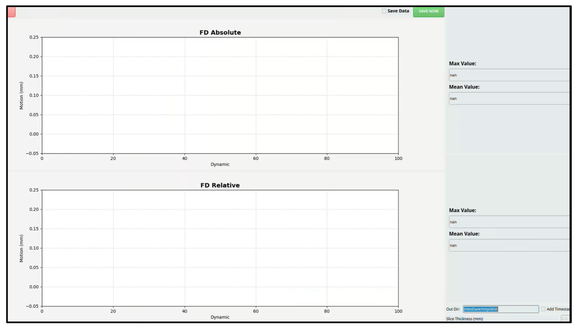
and…

α, β, γ = rotation (pitch, roll, yaw)

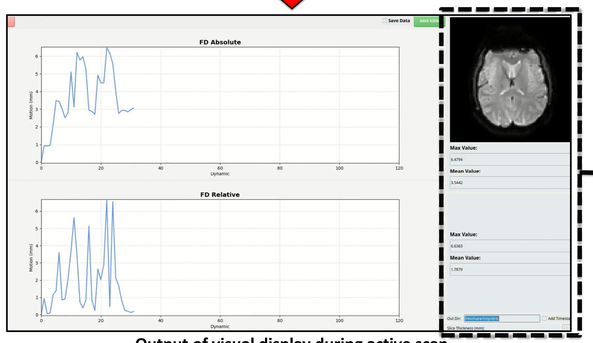


PyQt5-Based GUI Display

Real-time motion tracking begins at scan start



Output of visual display before scan start



Output of visual display during active scan

Current MRI Scan:
Most recent MRI image frame

Real-Time Statistics:
Maximum and mean motion values for relative and absolute motion plots

Output Directory:
Path used for saving output files

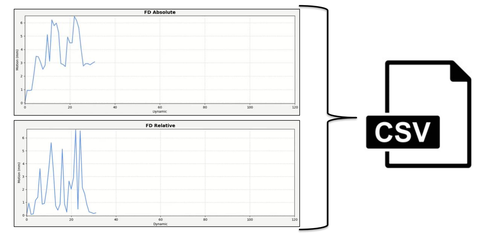
Slice Thickness:
User-defined input for scan slice thickness

Add Timestamp Checkbox:
Appends the current timestamp to saved files

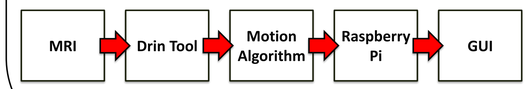
PYQT5

Results

A visual display system was developed to visualize patient motion during an FMRI. The motion plots reflect the relative and absolute framewise displacements of the patient in real-time. The motion data can be saved as a CSV file.



The system uses the Philips Research tool for image data and performs framewise displacement calculations on the image data.



System Workflow

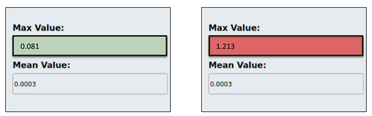
Conclusions

- The developed system visualizes real-time patient motion using framewise displacement
- Plots for absolute and relative motion provide MRI staff with intuitive feedback
- The Python-based algorithm runs on the Raspberry Pi 5 with a GUI implemented using PyQt5

Future Work

- Validate motion accuracy with offline analysis
- Consider alternate processor options for increased speed
- Implement conditional formatting of statistical elements (Max/Mean) for when motion exceeds thresholds

Max motion threshold: 1.000



Example of conditional formatting on excessive motion

Acknowledgments

Thank you to Dr. Matthew Sherwood, Dr. Hugh Salehi, and Dr. Bob Myers for their advice and ongoing support throughout this project.