

Date: 9/12/2013

This archive contains binary executables that can be used to extract, match and visualize scale-invariant features in volumetric images. Additionally, input data and expected output data are provided for verification.

Executables were compiled for Windows (\*.exe) and Linux (Ubuntu, \*.ubu).

This code provides an implementation of applications and functionality described in the following references:

- M. Toews, W.M. Wells III. "Efficient and Robust Model-to-Image Alignment using 3D Scale-Invariant Features", Medical Image Analysis, 17(3), 2013, pp 271-282.
- M. Toews, L. Zöllei, W.M. Wells III. "Invariant Feature-based Alignment of Volumetric Multi-modal Images", Image Processing in Medical Imaging (IPMI), 2013, pp. 25-36, 2013.

Please direct questions or comments to Matthew Toews ([mt@bwh.harvard.edu](mailto:mt@bwh.harvard.edu), [matt.toews@gmail.com](mailto:matt.toews@gmail.com)) or William Wells III ([sw@bwh.harvard.edu](mailto:sw@bwh.harvard.edu)).

Further updates and information can be found at [www.matthewtoews.com](http://www.matthewtoews.com)

#### DISCLAIMER

THIS SOFTWARE IS PROVIDED "AS IS" FOR NON-COMMERCIAL RESEARCH USE ONLY. ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

#### **OASIS Data Usage Agreement (DUA)      [www.oasis-brains.org](http://www.oasis-brains.org)**

Several images from the Open Access Structural Imaging Series (OASIS) are used as example data.

Data users must recognize & agree to the following points:

1. The quality and completeness of the data cannot be guaranteed. Users employ these data at their own risk.
2. Users shall respect restrictions of access to sensitive data. Users will make no attempt to identify the individuals whose images are included in OASIS data sets.
3. Users must acknowledge the use of OASIS data and data derived from OASIS data when publicly presenting any findings or algorithms that benefited from their use. Such presentations include but are not limited to papers, books, book chapters, conference posters, and talks.  
When publishing findings that benefit from OASIS data, please include the following grant numbers in the acknowledgements section and in the associated Pubmed Central submission: P50 AG05681, P01 AG03991, R01 AG021910, P20 MH071616, U24 RR021382.
4. Redistribution of original OASIS data is permitted so long as the data are redistributed under the same terms and conditions are described in this DUA.
5. Data derived from original OASIS data may be distributed under terms and conditions established by the creators of the data. Users must comply with the terms and conditions of use set by the creators of the data.

\*\*\*\*\* **Files** \*\*\*\*\*

Binary Executables

featExtract.(exe/ubu) – feature extraction  
featMatchMultiple.(exe/ubu) – feature matching  
featView.(exe/ubu) – feature viewing  
featResample.(exe/ubu) – image resampling

Sample Brain Images (from the OASIS data set)

OAS1\_0001.hdr OAS1\_0001.img  
OAS1\_0002.hdr OAS1\_0002.img  
OAS1\_0003.hdr OAS1\_0003.img

Extracted keypoints for each image (generated by featExtract.exe)

OAS1\_0001.key  
OAS1\_0002.key  
OAS1\_0003.key

Keypoints transformed to the reference image OAS1\_0001.key (generated by featMatchMultiple.exe)

OAS1\_0002.key.update.key  
OAS1\_0003.key.update.key

Report on number of inliers found for each image match (generated by featMatchMultiple.exe)  
report.txt

Lists of matching features found for each image and reference image (generated by featMatchMultiple.exe)

OAS1\_0002.key.matches.img1.txt – matches in reference image (OAS1\_0001.key)  
OAS1\_0002.key.matches.img2.txt – matches in image (OAS1\_0002.key)  
OAS1\_0003.key.matches.img1.txt – matches in reference image (OAS1\_0001.key)  
OAS1\_0003.key.matches.img2.txt – matches in image (OAS1\_0003.key)

Global similarity transform between image and reference image (generated by featMatchMultiple.exe)

OAS1\_0002.key.trans.txt – transform mapping OAS1\_0002.key to OAS1\_0001.key  
OAS1\_0002.key.trans-inverse.txt – transform mapping OAS1\_0001.key to OAS1\_0002.key  
OAS1\_0003.key.trans.txt – transform mapping OAS1\_0003.key to OAS1\_0001.key  
OAS1\_0003.key.trans-inverse.txt – transform mapping OAS1\_0001.key to OAS1\_0003.key

Visualization of a feature in OAS1\_0001.hdr in pgm format (featView.exe)

view\_test.pgm.OAS1\_0001.hdr\_x100\_y100\_z100.pgm

\*\*\*\*\* **Command Line Examples** \*\*\*\*\*

Run the following command line examples to extract features, match features, view feature matches, and to resample images.

```
>featExtract.exe .\OAS1_0001.hdr .\OAS1_0001.key  
>featExtract.exe .\OAS1_0002.hdr .\OAS1_0002.key  
>featExtract.exe .\OAS1_0003.hdr .\OAS1_0003.key
```

These commands extract features in each image, and produce output keypoint files \*.key

```
>featMatchMultiple.exe OAS1_0001.key OAS1_0002.key OAS1_0003.key
```

These commands will match all (\*.key) feature files to a reference image / feature file. The reference image / feature file is the first on the command line, here OAS1\_0001.key.

The output will be several text files detailing transforms, corresponding features, etc., see below.

Note: wildcards can be used, e.g. featMatchMultiple.exe \*.key. Be careful here as additional updated/transformed keypoint files (\*.update.key) maybe be present.

```
>featResample.exe OAS1_0001.hdr OAS1_0002.hdr OAS1_0002.key.trans-inverse.txt  
OAS1_0001_0002.hdr
```

This will resample OAS1\_0002.hdr into reference space OAS1\_0001.hdr, using the OAS1\_0002.key.trans-inverse.txt file generated by featMatchMultiple.exe  
The output file is OAS1\_0001\_0002.hdr, it should be a slightly shifted/rescaled version of OAS1\_0002.hdr

```
>featView.exe OAS1_0001.hdr 100 100 100 10 view_test
```

This will output a visual display image of a feature at xyz=(100,100,100), scale 10, located in image OAS1\_0001.hdr.

Produces an output image in pgm format view\_test.OAS1\_0001.hdr\_x100\_y100\_z100.pgm

```
>featView.exe OAS1_0002.key.matches.img2.txt
```

This will output a visual display of all features from OAS1\_0002.key that matched to OAS1\_0001.key during featMatchMultiple.exe

Input is text file OAS1\_0002.key.matches.img2.txt created by featMatchMultiple.exe

Produces 33 output images in \*.pgm format

```
>featView.exe OAS1_0003.key.matches.img2.txt
```

This will output a visual display of all features from OAS1\_0003.key that matched to OAS1\_0001.key during featMatchMultiple.exe

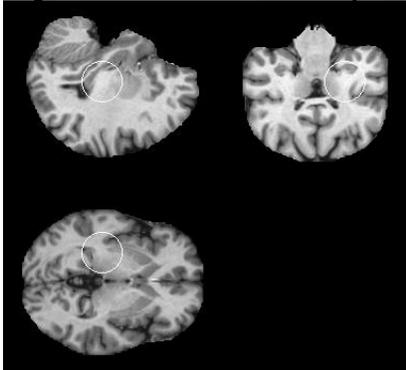
Input is text file OAS1\_0003.key.matches.img2.txt created by featMatchMultiple.exe

Produces 70 output images in \*.pgm format

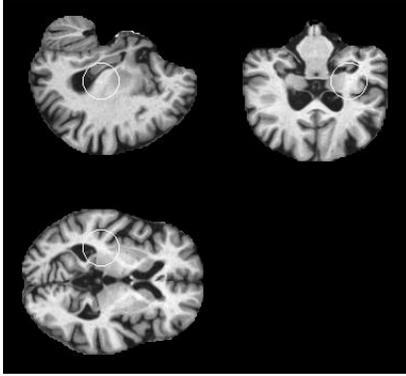
\*\*\*\*\* **Notes** \*\*\*\*\*

**Note 1:** There should be two visualization images beginning with 'img1\_002070'. This means that there are features in OAS1\_0002.key and OAS1\_0003.key that mapped to the same feature 2070 in the reference image OAS1\_0001.key. This indicates the presence of image structure that is similar in geometry and appearance in all three images, see white circles shown below in sagittal, axial and coronal views.

img1\_002070.OAS1\_0002.key\_x114\_y093\_z076.pgm



img1\_002070.OAS1\_0003.key\_x118\_y092\_z076.pgm



**Note 2:** as of this code release, all executables operate in voxel space and not in world coordinate space. Input images should be resampled isotropically if necessary.

**Note 3:** algorithms results differ slightly from one operating system to another, apparently due to small differences in floating point math processing.

Note 4:

\*\*\*\*\* **Future ToDo** \*\*\*\*\*

1. Feature-based morphometry application.
2. Feature-based model construction application.
3. Conversion between voxel and world coordinates.