Medical imaging in the web browser with the A* Medical Imaging (AMI) toolkit.

N. Rannou[1], J.L. Bernal-Rusiel[2], D. Haehn[3], P.E. Grant[2], R. Pienaar[2]

[1]Eunate Technology S.L, Spain, [2]Boston Children's Hospital,Fetal-Neonatal Neuroimaging and Developmental Science Center (FNNDSC), USA, [3]Harvard University, USA

Purpose of the Software

We present AMI (A* Medical Imaging) ([1]), a new JavaScript toolkit for medical imaging on the web.

Popular currently available libraries are often very specialized and may lack some core imaging features commonly required by applications: some are 2D only with no support for popular 3D models ([2], [3], [4]), others might support concurrent 2D and 3D visualization but often these are not in the same rendering window ([5], [6]), many have limited UI tool elements ([7], [8]), GPU acceleration is either limited or not present in most existing solutions ([3], [4], [5], [6], [7]), etc.

AMI is a single toolkit that includes 2D and 3D visualization of most popular medical imaging data formats, the ability to render 2D/3D images and models concurrently in the same scene, allows for real time interaction and also provides a set of useful UI elements (such as rulers, painters, selectors, etc).

Methods and Implementation

The framework is written in JavaScript and is developed through continuous integration for quality assurance. It has a modular architecture that makes it easily extensible, testable and understandable.

WebGL and WebVR support is provided by layering onto three.js ([8]).

A flexible design means that actual image and model data parsing is not part of the core library but external parsers can be easily linked to the library.

Data parsers: DICOM, NIFTI, NRRD, MHD/RAW.
Model parsers: VTK, STL, TRK.

UI tools are designed to work seamlessly in 2D and 3D. Existing UI elements can also be simply combined to create new elements.

Tools available (2D and 3D): distance measurements, angle measurements, annotations.

AMI is open source to promote transparency, collaboration and improve code quality. New features submission and bug fixes are encouraged. All contributions are carefully reviewed and tested before integration to the main repository.

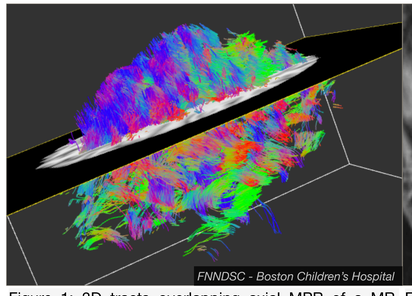

Figure 1: 3D tracts overlapping axial MPR of a MR DICOM dataset.

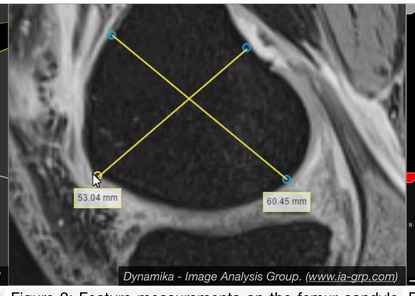

Figure 2: Feature measurements on the femur condyle MRI.

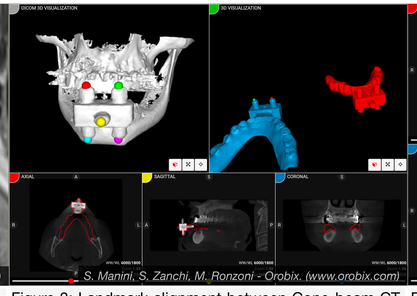

Figure 3: Landmark alignment between Cone-beam CT and surgical guide.

Figure 4: DICOM and Volume rendering visualization of a Cone-beam CT.

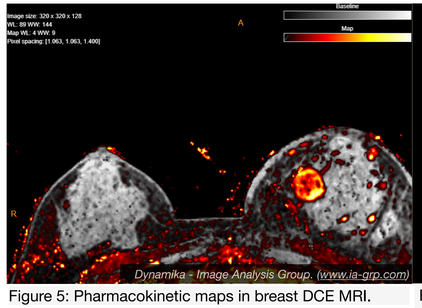

Figure 5: Pharmacokinetic maps in breast DCE MRI.

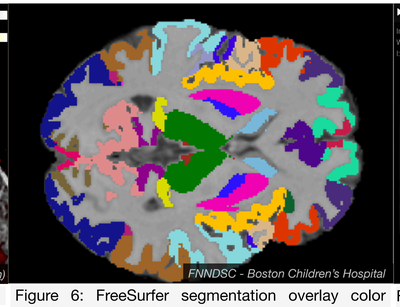

Figure 6: FreeSurfer segmentation overlay color coded by anatomical region.

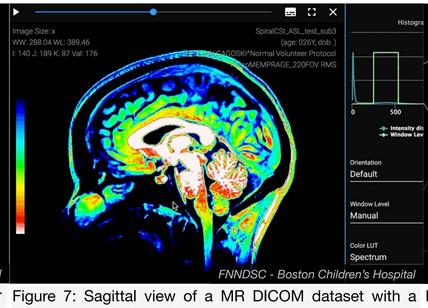

Figure 7: Sagittal view of a MR DICOM dataset with a custom color lookup table.

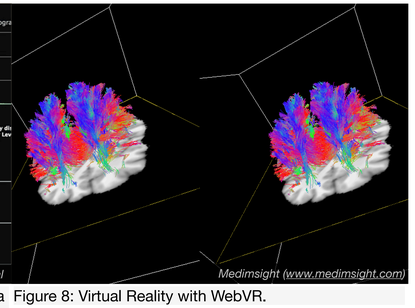

Figure 8: Virtual Reality with WebVR.

Target Audience

AMI's target audience are JavaScript developers seeking to leverage the latest WebGL and WebVR technologies in Medical Imagining. The API is friendly and easy to use. A complete application can be created in 15 lines of code (Figure 9).

Future Directions and Conclusion

AMI will incorporate more support for medical image formats and models (e..g FreeSurfer mesh support) and trk file support. Additional directions include VR support and a wider selection of tools. It is hoped that AMI will help accelerate the viewing and processing of medical data on the web.

```
// AMI imports
import XRenderer3D from '../../src/helpers/x/helpers.x.renderer3d';
import XVolume from '../../src/helpers/x/helpers.x.volume';

window.onload = function() {
    // INIT THE RENDERER
    const renderer = new XRenderer3D();
    renderer.animate();

    // CREATE THE 3D VOLUME
    const xVolume = new XVolume();
    xVolume.file =
      'https://cdn.rawgit.com/FNNDSC/data/master/nifti/marc_avf/avf_float_32.nii.gz';
    xVolume.progressbarContainer = renderer.container;

    // LOAD AND RENDER THE 3D VOLUME
    xVolume.load().then((volume) => {
        renderer.add(volume);
        renderer.center(volume.centerLPS);
    }).catch((error) =>
      console.log('ERROR: something went wrong with the volume load.', error));
};
```

FNNDSC - Boston Children's Hospital

Figure 9: Implementation of a fully functional 3D NIFTI viewer in 15 lines of code with AMI.

References

[1] AMI "https://github.com/FNNDSC/ami"
[2] Cornerstone "https://github.com/chafey/cornerstone"
[3] OHIF Viewers "https://github.com/OHIF/Viewers"
[4] Bernal-Rusiel et al. "Reusable Client-Side JavaScript Modules for Immersive Web-based Real-time collaborative Neuroimage Visualization." Frontiers in Neuroinformatics, 2017
[5] Sherif et al. "BrainBrowser: distributed, web-based neurological data visualization." Frontiers in Neuroinformatics, 2015
[6] Papaya "https://github.com/rii-mango/Papaya"
[7] Haehn et al. "Neuroimaging in the Browser using the X Toolkit." Frontiers in Neuroinformatics, 2014
[8] three.js "https://github.com/mrdoob/three.js/"