# CHERE
# User Manual

Structure from Motion component is based on:

- VisualSFM (ccwu.me/vsfm) is used for feature detection and matching as well as reconstruction.
- SIFT - by using vlfeat (www.vlfeat.org)
- Yasutaka Furukawa's PMVS/CMVS (grail.cs.washington.edu/software/cmvs/)
- Screened Poisson Surface Reconstruction (http://www.cs.jhu.edu/~misha/Code/PoissonRecon)
- texrecon (https://github.com/nmoehrle/mvs-texturing)
- meshlab (www.meshlab.net)
- Nexus by Visual Computing Laboratory - ISTI - CNR (http://vcg.isti.cnr.it/nexus)

Web application and service is based on:

- Yii 2 PHP framework (www.yiiframework.com)
- 3DHOP library (vcg.isti.cnr.it/3dhop/)

## Using the Docker container

If the intended use is to perform SfM on local resources, relevant files are located at https://mis.etfbl.net/sfm and they represent Docker containers (two versions depending on installation and configuration of docker on target machine) sfm-v1.tgz and sfm-v2.tgz, as well as compiled and gathered basic tools used in container in sfm.tar.bz2 archive.

The container expects images in /input directory and will save output to /output directory. It will produce sparse and dense reconstruction as well as textured models in OBJ and PLY formats as well as NXS format suitable for streaming.

You can use binding mount or volumes, etc. for /input and /output. For example:

```
docker run --mount type=bind,source=/path/to/images,target=/input –mount
  type=bind,source=/path/to/output,target=/output sfm:latest /sfm/sfm
```

or via volumes

```
docker run -ti -v /input:/input -v /output:/output sfm:latest /sfm/sfm
```

Output of successful run is a set of files (with intermediary files left in place in order to facilitate troubleshooting or additional processing) including:

- Sparse and dense point clouds in PLY format.
- Reconstructed 3D surface (poisson) in PLY format.
- Texturized model in OBJ format (texture is in seaparate file).
- Object with texture transferred to mesh color in PLY format.
- NXS conversion of color PLY object.

# Accessing the web application

CHERE web application service has been installed at University of Banja Luka Faculty of Electrical Engineering ETFBL-CC01 servers. It accessible via the following URL: https://mis.etfbl.net/chere

In order to be granted access, please open the ticket at https://support.vi-seem.eu/ and please specify the intended use of the service and provide basic information on individual and institutional identity. If one already has VI-SEEM AAI account, please specify the account email as well in order to provide suitable Single Sign On access.

Upon visiting the service, the user will be presented with login page which enables entering the local username/password pair, use of VI-SEEM AAI login as well as support for longer term remebering of user credentials. This is illustrated in Figure 1.

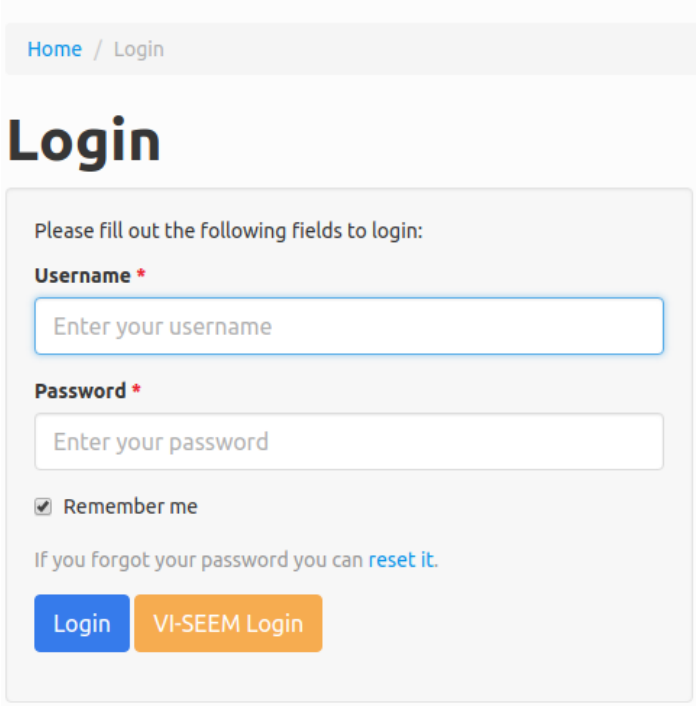


*Figure 1: CHERE Login page*

# Project management

Basic interaction is realized through "Projects". They are used to organize files and resources that users can host on the service. Each project has **input** and **output** directories, with input containing files uploaded by the user and output holding products of various transformations and processing workflows. This is illustrated in Figure 2.

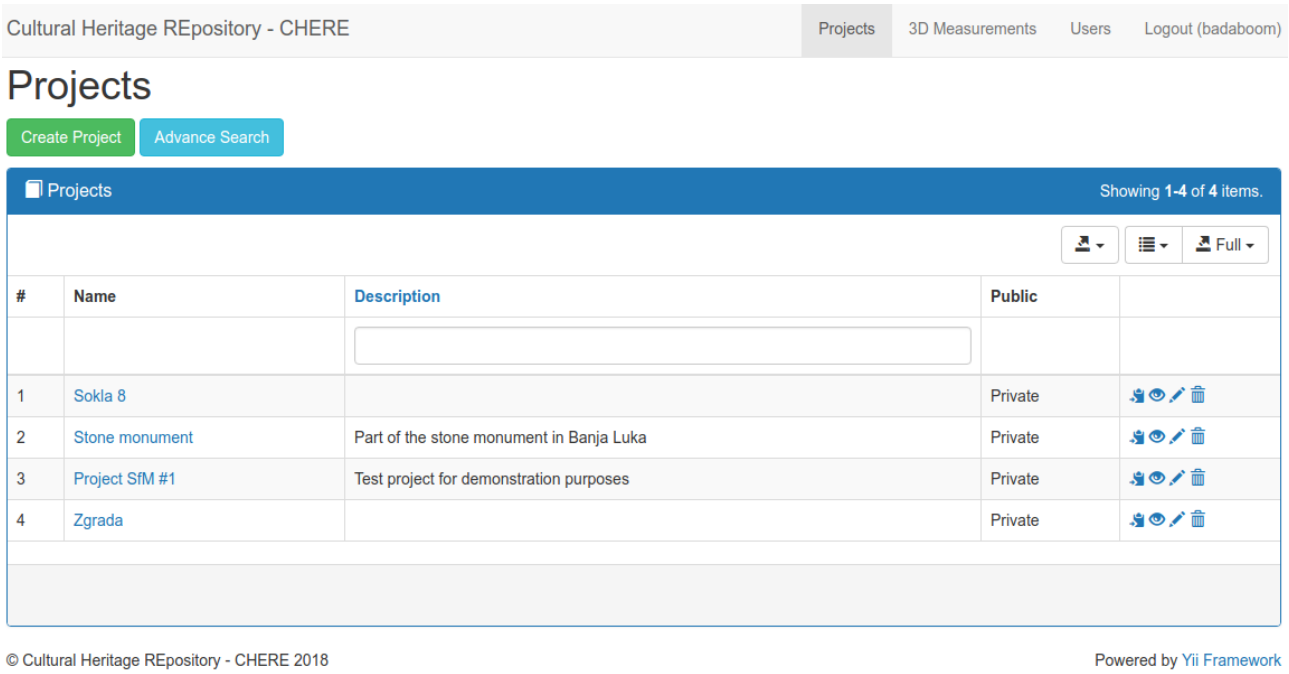


*Figure 2: Project management view*

Once the user select creating a new project or updating the existing one, the user is presented with interface that allows for editing basic data on project as well as uploading new files as seen in Figure 3.

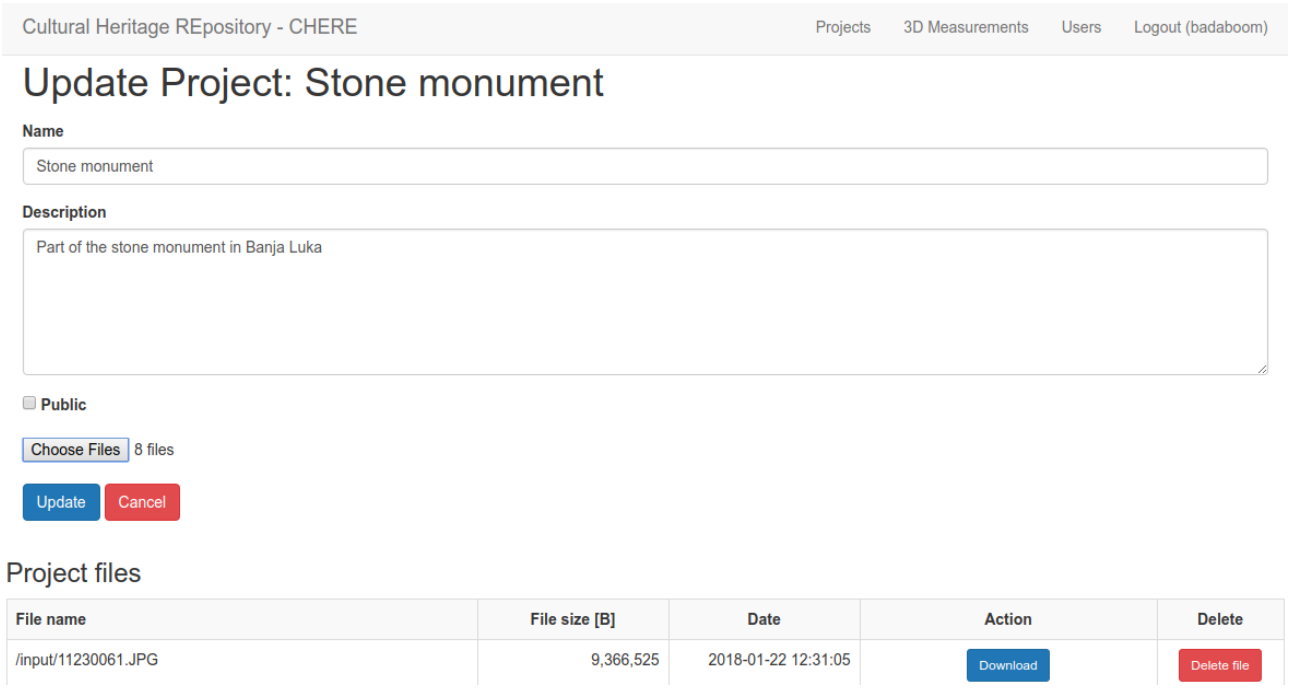


*Figure 3: Creating/Updating a project*

Individual project view page enables the user to initiate Structure from Motion reconstruction process (Figure 4). This assumes that **input** directory contains only the images to be used in reconstruction process whose products will be stored in **output** directory. Once initiated, the process submits the job to local cluster infrastructure. Multiple submission in parallel are not supported and will fail immediately.

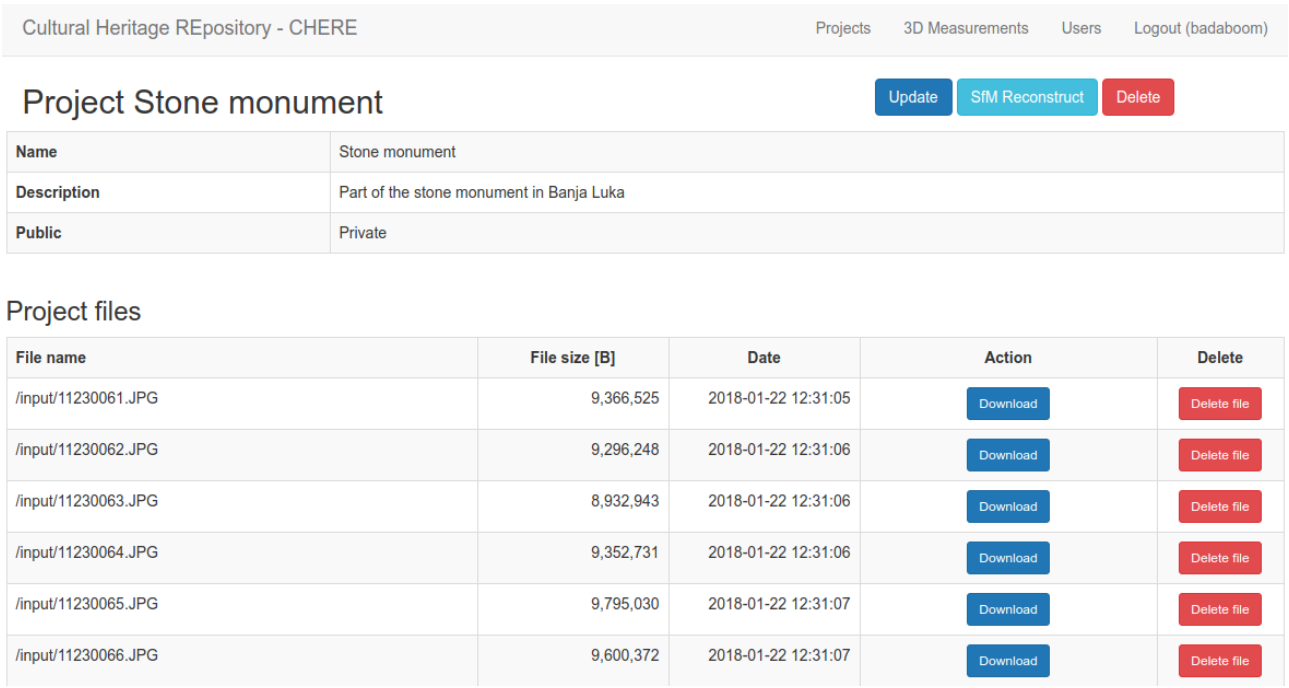


*Figure 4: Viewing project*

Once the reconstruction process is finished (or partially finished), the detailed individual project view allows for inspecting and performing other actions on existing files as is presented in Figure 5. User has the option to download or delete any file, to measure NXS and PLY files and to open for processing in Meshlab and PLY, STL, OBJ and OFF file.

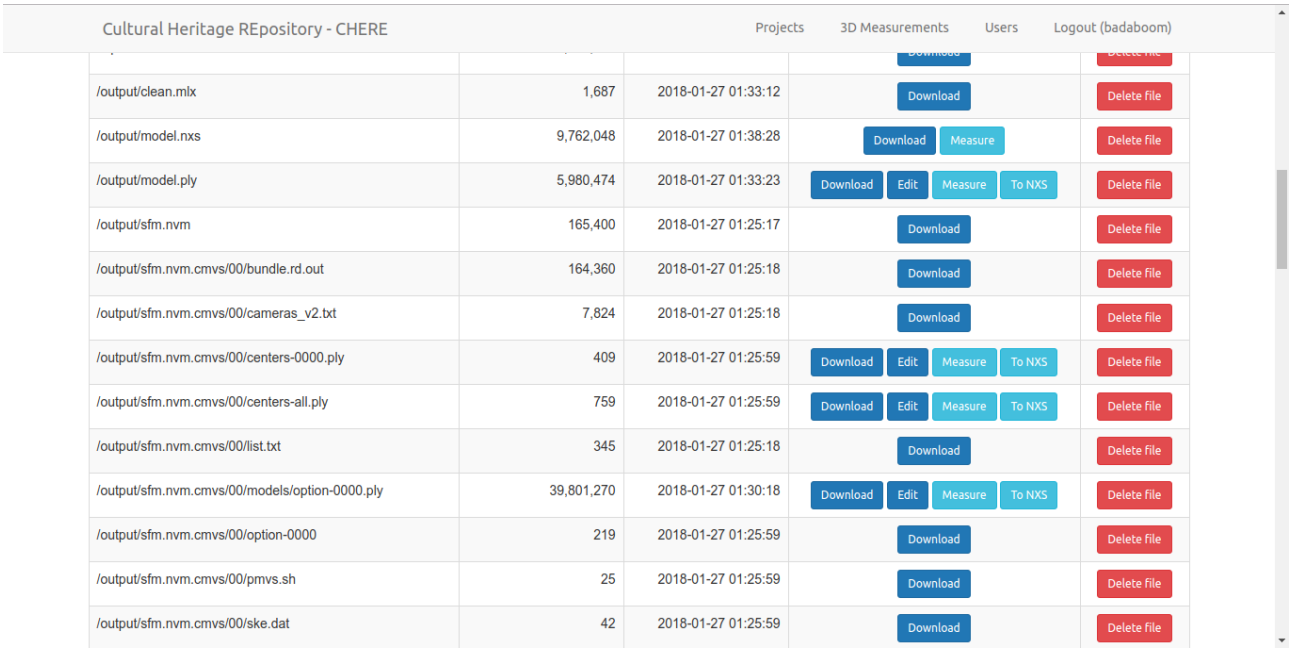


*Figure 5: Output of SfM reconstruction*

The application also supports transforming PLY input files to NXS format. NXS format represents the mesh model in streaming form, with first step containing the rough outlines of the object, with every successive step being more detailed. This allows for faster initial render for end users as they don't have to wait for the complete model to download in order to see anything on the screen. Output of one example is shown in Figure 6.

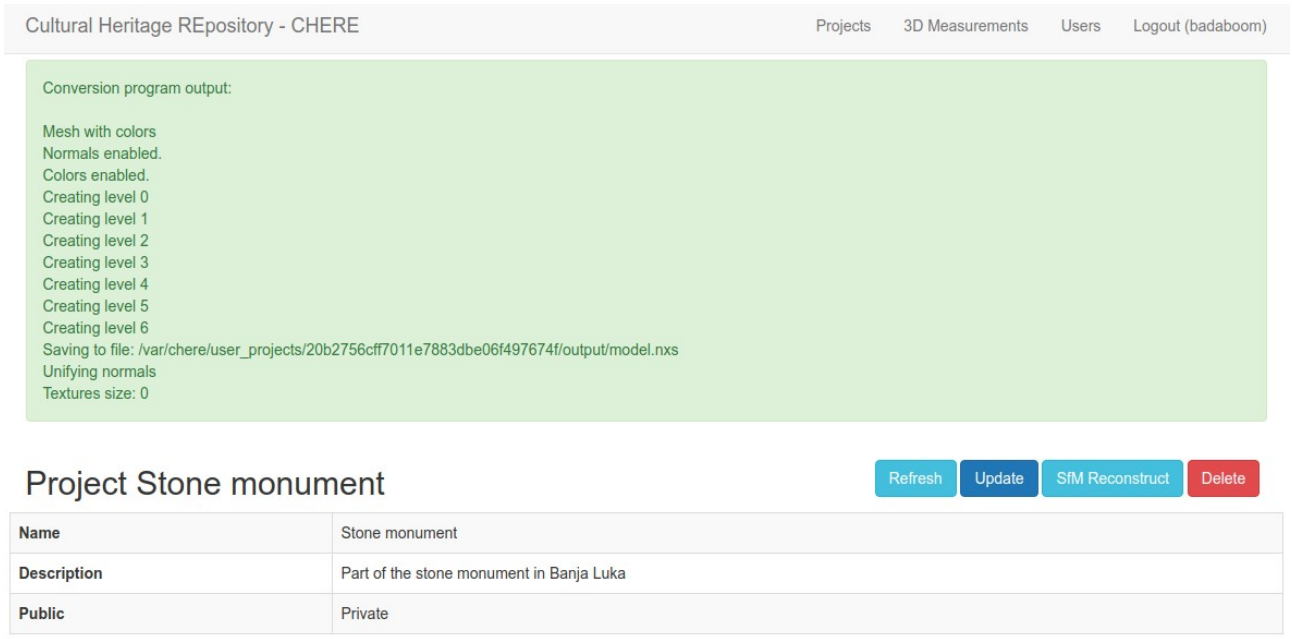


*Figure 6: Generating NXS file from PLY mesh*

While using 3D mesh processing software is preferred solution, it is often useful to be able to quickly inspect generated meshes and perform simpler global adjustments without the need to install, sometimes complex, software on end-user computers. In order to enable for this, we have installed and integrated an excellent web based version of popular Meshlab software named Meshlabjs. It can be launched from project files view by clicking on **Edit** button. For example, one can smooth and edit the mesh generated by SfM process as seen in Figure 7.

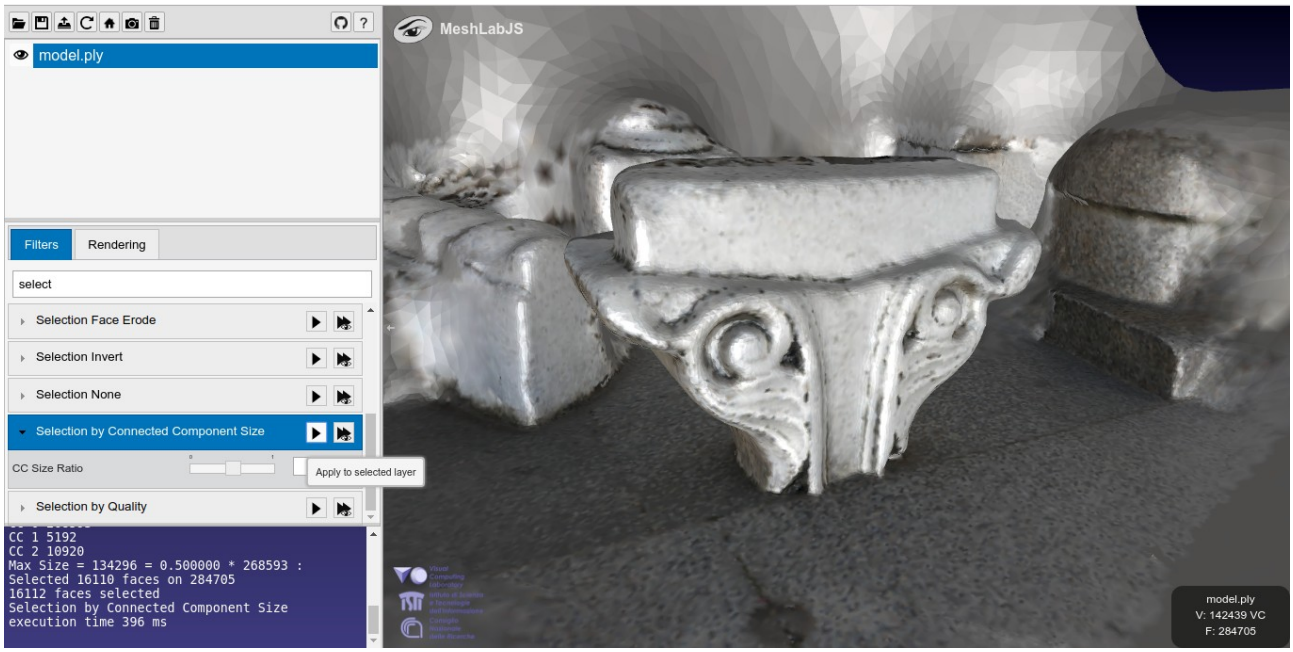


*Figure 7: Using Meshlabjs for mesh editing*

When we are working with PLY or NXS meshes, we can use adapted version of 3DHOP environment to perform basic measurements and inspections on meshes as seen in Figure 8. This environment allows for working with cross sections with visible planes and/or edges which can be very usesful in certain circumstances, such as isolating the object, analysing internal structure, etc. Example is presented in Figure 9.


*Figure 8: Measuring 3D objects*

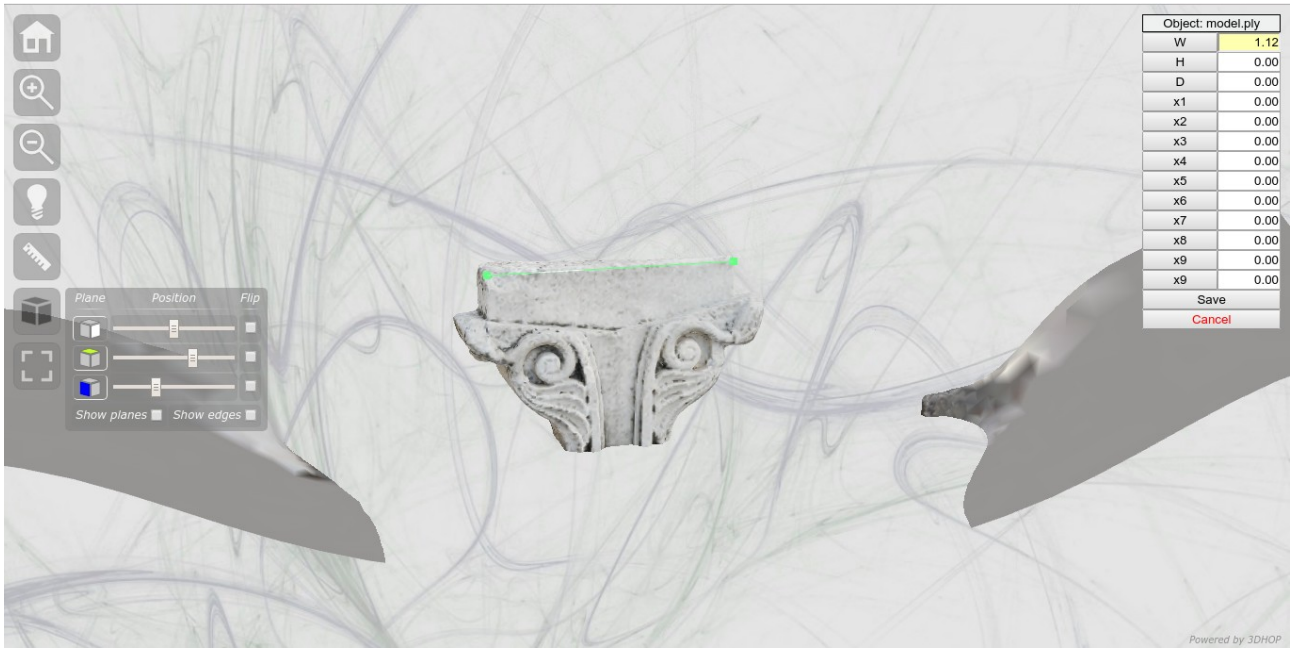


*Figure 9: Inspecting PLY mesh in 3DHOP*