# CHERE Tools
# User Manual

## About CHERE Tools

CHERE Tools stands for Cultural Heritage Repository Tools and represents a set of web based tools aimed at people working in cultural heritage preservation and digitization, but is not limited to those uses as individual services can be used in a variety of ways. The service currently provides following functions:
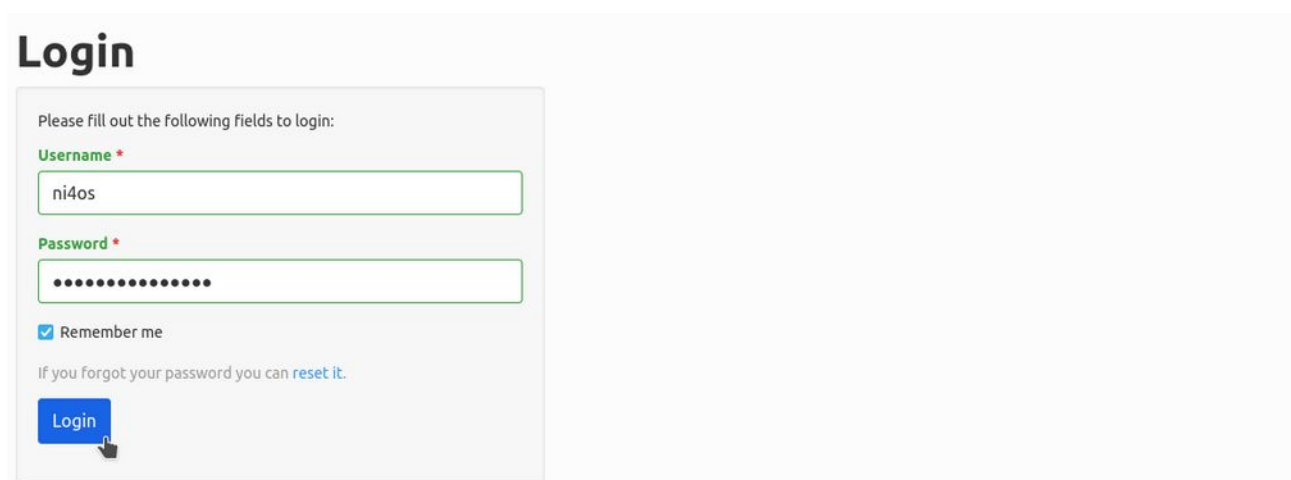
- Structure from Motion - reconstruction of textured 3D object from a series of photographs of the object.

- Measurement of 3D objects - either reconstructed or user-provided.

- Conversion of BigBlueButton meetings' recordings to standalone video files.

## Accessing the web application

CHERE web application service has been installed at University of Banja Luka Faculty Electrical Engineering ETFBL-CC01 compute cluster as a part of NI4OS-Europe project. The service is accessible via the following URL: https://chere.etfbl.net/

In order to be granted access, please contact Mihajlo Savić at mihajlo.savic@etf.unibl.org and please specify the intended use of the service and provide basic information on individual and institutional identity.

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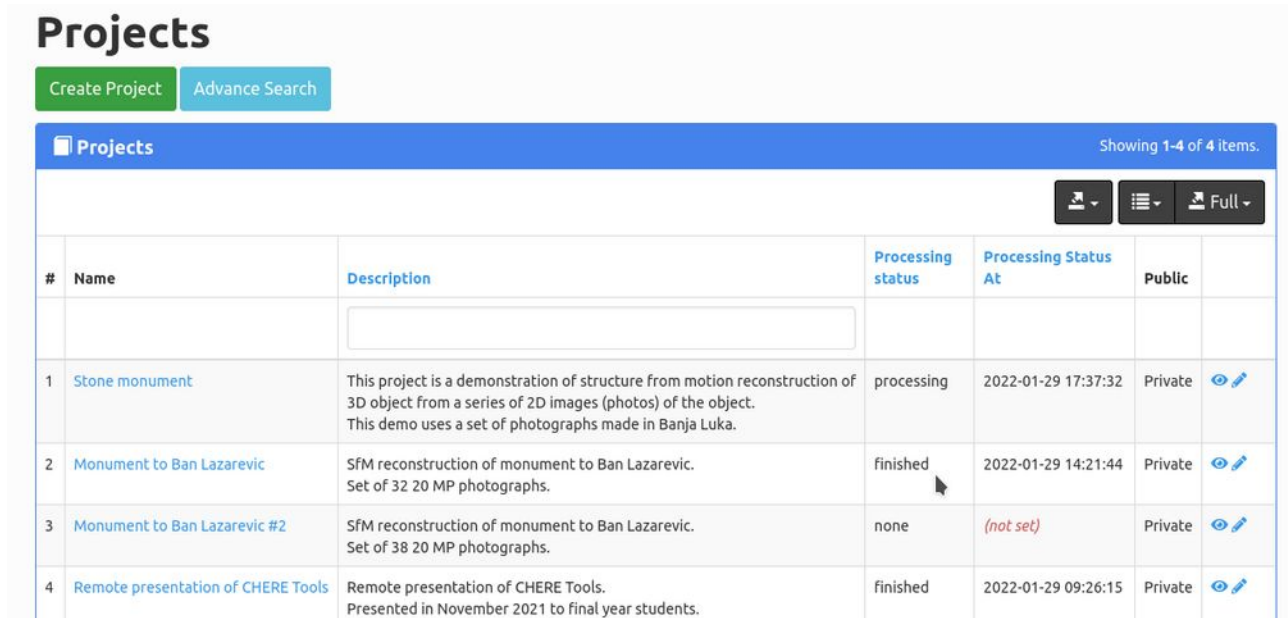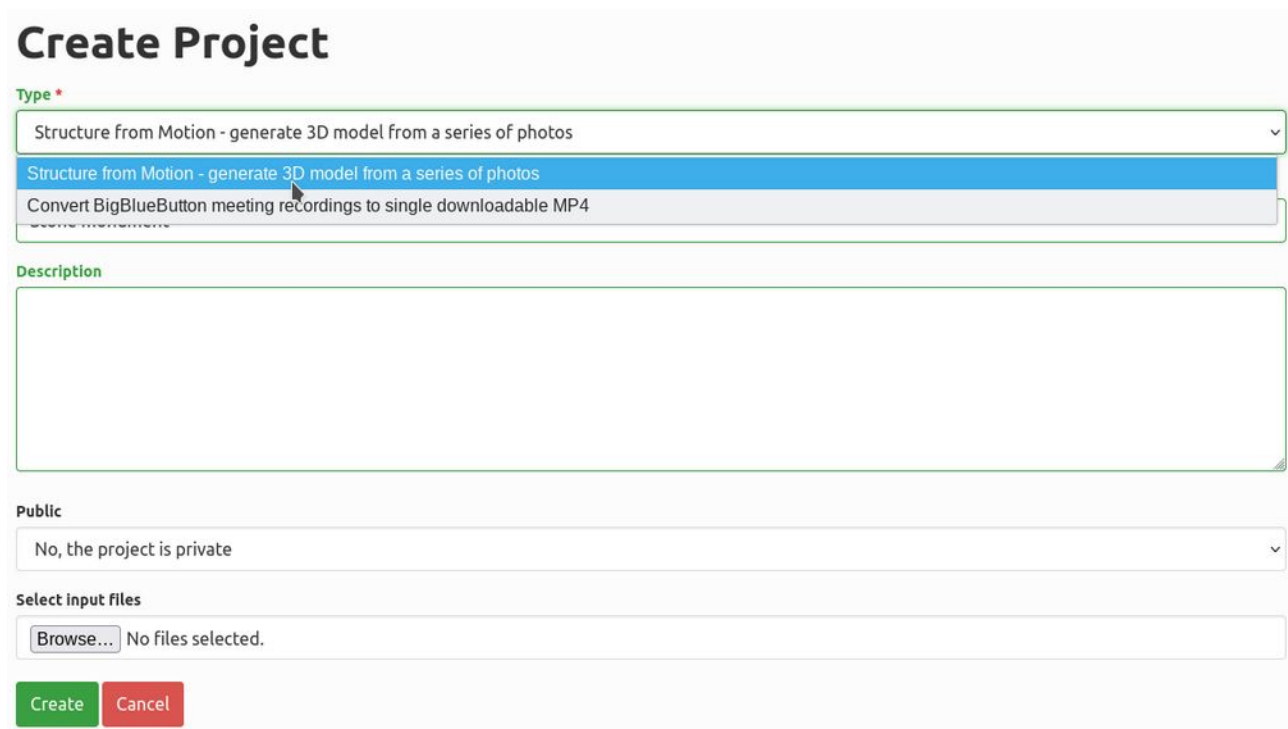


*Figure 1: CHERE Login page*

# 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 a name, a description, processing status as well as a type and private or public status. This is illustrated in Figure 2.



*Figure 2: Project management view*



*Figure 3: Creating/Updating a project*

Once the user selects 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. During creating the project, the user chooses the type of the project which defines the processing possible on the input data.

Individual project view page enables the user to initiate processing of the project by external worker tasks (Figure 4). This processing assumes that **input** directory contains input files to be used in processing whose products will be stored in **output** directory, while the log files are stored in **logfiles** directory. Once initiated, the process submits the job to local cluster infrastructure. Multiple submission in parallel of the same project are not supported, while the user is free to submit processing of multiple projects simultaneously.



*Figure 4: Viewing project details*

The project status changes during the workflow as illustrated in Figure 5. Upon creation or update, the project is put in **ready for processing** status which enables the user to send it to processing queue (**queued** state). Once an available worker takes over the project to process, it is transferred to **processing** state (the project passes through **processing-init** state while the worker is obtaining input files during which the user can remove the project from processing queue). While the project is in **processing** state, the user can observe the log files which are sent by worker in real time. Upon completion of processing, the project is in **finished** state and the user can download or view processing output. This is illustrated in

If the user observes some deviations in logfiles or simply chooses to give up on processing, the user can abort the project in which case the project is put into **aborting** state until the assigned worker acknowledges that the processing is terminated and the project is then put into **aborted** state.

If the project is not being currently processed, the user can update the project at which point it passes to **ready for processing** state and can be sent to processing queue again.
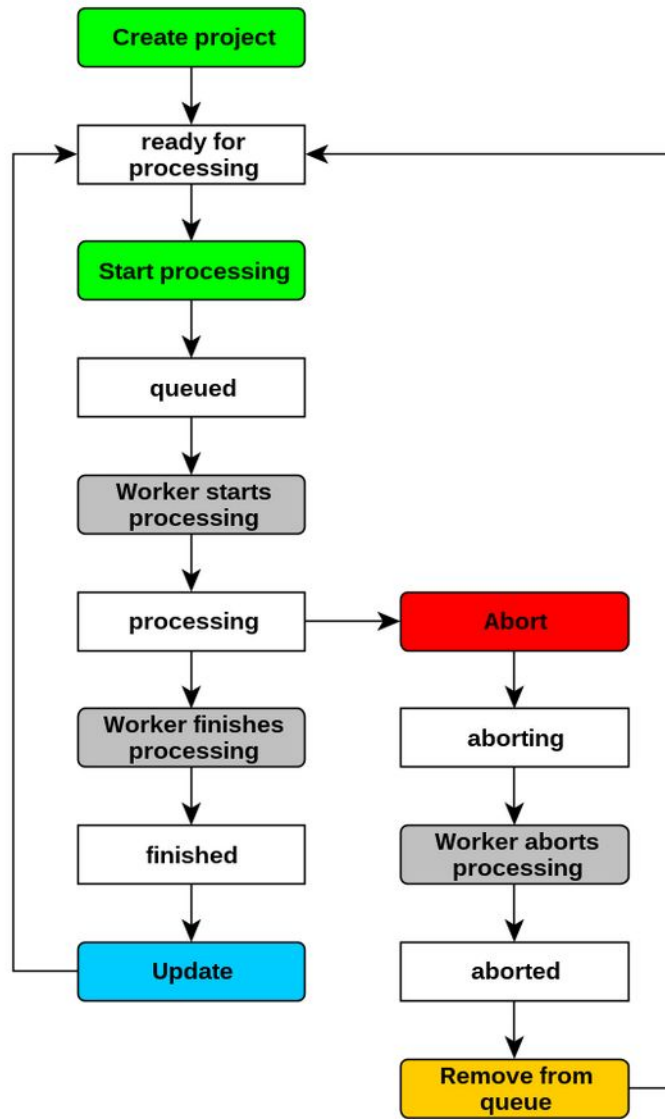
*Figure 5: Project workflow statuses*

## Project files

| File name | File size [B] | Action | | | Delete |
|---|---|---|---|---|---|
| **Output files** | | | | | |
| /output/model.nxs | 7,095,296 | Measure | View | Download | Delete file |
| /output/model.obj.tgz | 11,374,068 | | View | Download | Delete file |
| /output/model.ply | 4,322,916 | Measure | View | Download | Delete file |
| **Logfiles** | | | | | |
| /output/sfm.log | 30,359 | | View | Download | Delete file |
| /output/sfm.status | 879 | | View | Download | Delete file |
| **Input files** | | | | | |
| /input/11230061.JPG | 9,366,525 | | View | Download | Delete file |

*Figure 6: Output of SfM reconstruction*

Once the processing is finished, the detailed individual project view allows for inspecting and performing other actions on existing files as is presented in Figure 6. User has the option to view (in browser if supported), download (to local computer) or delete (from server) any file, as well as to measure NXS and PLY files in provided viewer based on 3D HOP system as shown in Figure 7.



*Figure 7: Measuring 3D objects in 3D HOP viewer*

The SfM system also produces a high quality fully textured 3D object in OBJ format suitable for further processing on local computer. As this format consists of multiple files, it is compressed into an archive (model.obj.tgz) in gzipped tar format. Upon download and extraction, the model can be opened in any compatible software, such as MeshLab shown in Figure 8.
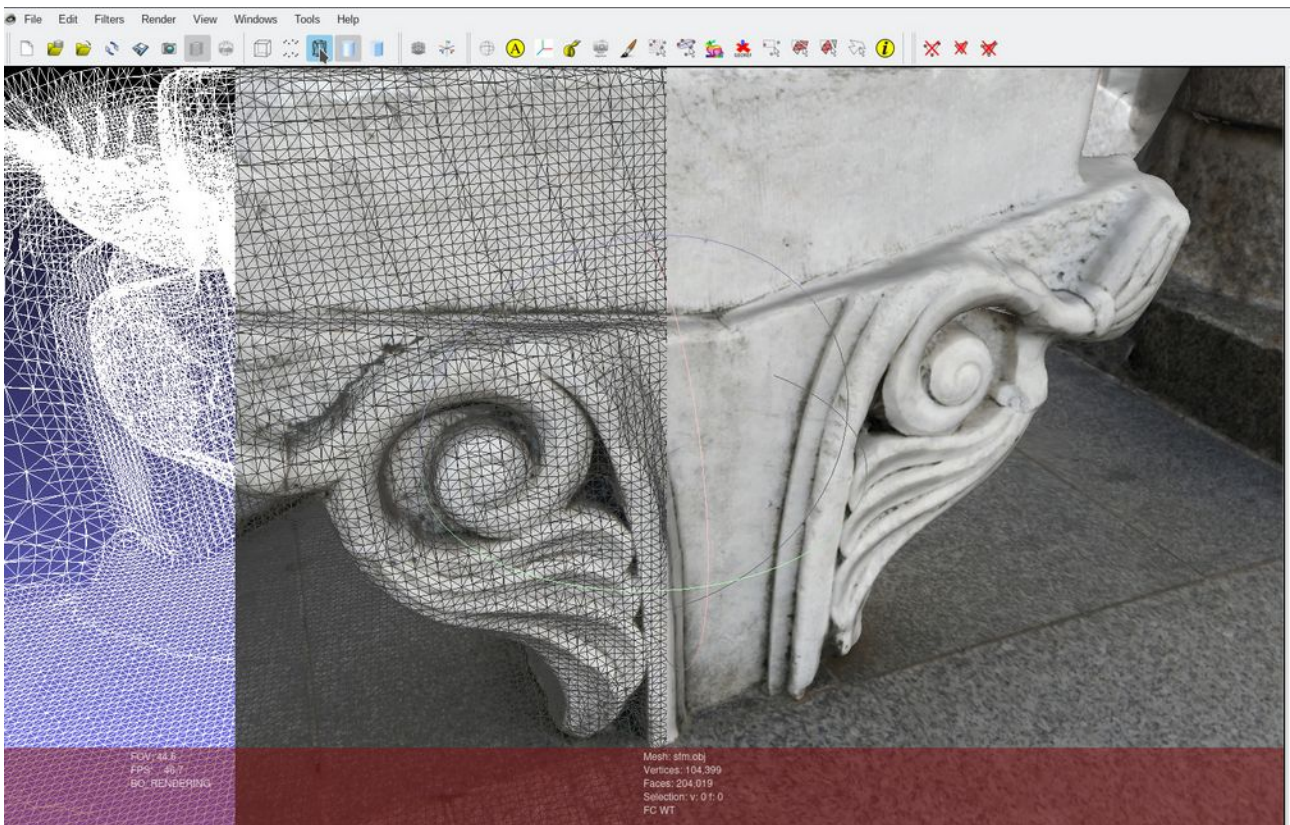


*Figure 8: MeshLab processing generated 3D model*

# BigBlueButton conversion to video

CHERE Tools also provides users with a means to convert a saved BigBlueButton meeting to a standalone video file. While BigBlueButton meetings are efficiently and effectively stored and procesented from server, many users prefer having the ability to make local copies of meetings in a familiar format, such as MP4 or WEBM.

All the user needs to do is to select a suitable project type (Convert BigBlueButton meeting recordings to single downloadable MP4) and enter a valid BigBlueButton meeting URL like https://somedomain.com/playback/presentation/2.3/somelongidentificationstring and send the project to processing queue. Processing workflow is identical to previously described SfM workflow and can be interrupted at any time.

Important thing to consider is that the conversion is happening in real time which means that conversion of a meeting that lasted two hours will take at least two hours (duration of meeting plus any additional processing and recompression time).

Upon completion, the user is provided with a single file representing the meeting recording. The file is an MKV container with video (H.264 codec 1280x720 resolution) and stereo audio (Opus codec).

# Using the Docker container for SfM reconstruction

If the intended use is to perform SfM reconstruction on local computing resources, relevant files are located at https://mis.etfbl.net/sfm and they contain:
- Dockerfile used for building the image
- Docker image in tgz format
- Compiled basic tools used in images in sfm.tar.bz2 archive
- Test files in test directory
- This user manual

The container expects images in /input directory and will save output to /output directory. It will produce dense model reconstruction as well as textured models in OBJ and PLY formats. Since the generated models can be quite large, the container will also produce an NXS format representation suitable for streaming and viewing in 3D HOP service.

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

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

and 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.
- Object with texture transferred to mesh color in PLY format.
- NXS conversion of color PLY object.

# Tools used in development of CHERE SfM process

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