Pix2Model: A free, easy-to-use Photogrammetry Webservice

Philipp Erler* TU Wien
Maximilian Riegler† TU Wien
Johannes Eschner‡ TU Wien
Florian Steinschorn§ TU Wien
Sophie Pichler¶ TU Wien

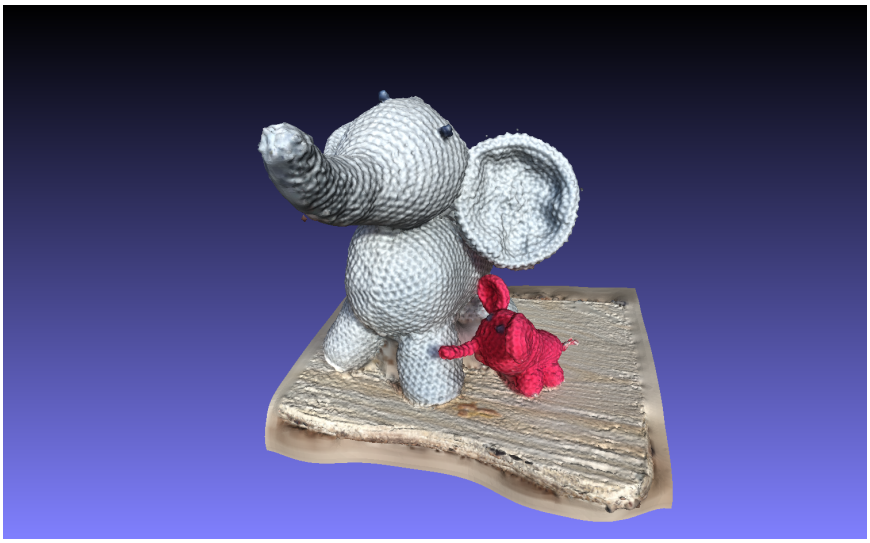


Figure 1: Reconstruction result created with Pix2Model.

Abstract

In short, we did good.

The system can be tested here:
`https://netidee.cg.tuwien.ac.at/`
The source code is available on Github:
`https://github.com/ErlerPhilipp/Pix2Model`

CR Categories: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

Keywords: photogrammetry, webservice, micro services, structure-from-motion, multi-view stereo, surface reconstruction

1 Introduction

We will tell you about how we did good.

1. focus on novice users -¿ as easy as possible
2. minimal maintenance
3. easy to exchange algorithms

*e-mail: perler@cg.tuwien.ac.at
†e-mail: name@example.com
‡e-mail: jeschner@cg.tuwien.ac.at
§e-mail: f.steinschorn@gmail.com
¶e-mail: sophie.pichler@ufficina.de

2 Related Work

Others did good too.

2.1 Frontend

2.1.1 Usability

The user interface of Pix2Model provides functionality to load different models, to apply basic transformations, to crop away parts of the model and to display the dimensions of the loaded model. This functionality is chosen based on the requirements of the pipeline to generate a 3D model based on images. The user needs to be able to view and gradually edit the pointcloud model, which is generated in a first step based on the input images. Additionally he needs to be able to iterate this process after reviewing the reconstruction model that is generated from the edited pointcloud.

The implemented UX design is as easy as possible to aid novice users. We compared the applications Blender, Autodesk Maya and MeshLab in the following aspects. The results of the comparison factor into the development of the interface of Pix2Model while the focus is to reduce complexity even if it does not match standards from existing 3D software.

Layout
The layouts of Blender, Maya and MeshLab are depicted in figure 2. They all provide an overview of the scene objects - which is called *outliner* and is not part of the default layout in Maya - listed in a separate panel next to the viewport. The tools are selectable in a different toolbar-panel. By selecting an object, attributes can be changed in an attribute panel within Maya and Blender.

Pix2Model adapts the idea of having a separate panel to see the loaded object, to set attributes and to choose tools. But it combines all three panels as the user can only load one object at a time. Therefore it is possible to display the loaded object with its attributes and with the features that are available for the specific object type of the loaded object, i.e. pointcloud or mesh without repetition of menu items or confusion which object is getting edited.

Pix2Model outsources some of the tools that are connected to shortcuts as well to a separate toolbar window on the top left corner. In retrospect this functionality should have been part of the object panel as well to serve the purpose of simplicity and to show the user only the functionality that is available as soon as it is useful, i.e. as soon as a mesh is loaded.

Orbiting Scrolling the middle mouse button (MMB) is a standard user interaction for zooming. With the left mouse button (LMB) the user is orbiting around the scene and panning is implemented in Maya and Blender with a combination of Alt / Shift + LMB and in MeshLab by pressing the MMB while navigating through the scene. Pix2Model also implements scrolling, orbiting and panning and uses the established user interactions realized in Maya.

Transformations The shortcuts used for *Translation*, *Rotation* and *Scale* are listed in table 1. MeshLab does use key-combinations to press while interacting with the scene instead of enabling transformation tools upfront. As there is no general assignment between transformations and shortcuts, Pix2Model implemented the shortcuts according to the initials of the transformations. The center operation in Pix2Model uses *F* as a shortcut for *focusing* or *framing* a selected object.

Operations Maya and Blender provide *boolean operations* for cropping. A boolean operation is performed between two objects. One object can be used to cut away the intersecting area between the two objects or to only keep the intersecting area and cut away the parts that do not belong to that intersection. Pix2Model provides a box that can be activated to crop intersecting points from a pointcloud. Cropping can be applied iteratively to make complex cut-outs.

Undo Redo Maya provides multiple options to undo and redo an action. Buttons within the UI can be used, the functionality can be chosen from a dropdown in the upper menu and the common shortcuts Strg + Z and Strg + Shift + Z are connected to the undo and redo commands. Blender has as well menu items to undo and redo an action and shortcuts to call those. We chose a combination of buttons in the UI and the established shortcuts Strg + Z and Strg + Shift + Z for the undo / redo implementation in Pix2Model. The buttons are part of the implementation to inform the user that this functionality is available.

Mobile Support While Maya and Blender are complex applications, MeshLab has reduced functionality. Maya and Blender are cross-platform, but are not supported on mobile phones. MeshLab on the other hand has a simplified mobile version in which the user can view an object. Pix2Model as well functions as a 3D viewer on the mobile version without editing options.

	Blender	Maya	Pix2Model
Translation	G	W	T
Rotation	R	E	R
Scale	S	R	S
Centering	Numpad .	F	F

Table 1: Shortcuts for Transformations and centering an object

2.2 Backend

Flask (Build process) [**?**].
(nvidia) docker [**?**].
COLMAP [**?**].
Screened Poisson Surface Reconstruction [Kazhdan and Hoppe 2013].
Chamfer Distance [Barrow et al. 1977].
Hausdorff Distance [Huttenlocher et al. 1993]. Texturing [**?**].

3 Implementation

This is how we did good.

3.1 Structure From Motion

In the first step of our reconstruction pipeline, the Structure From Motion (SFM) implementation COLMAP[1] is used to extract a point-cloud from the input images. COLMAP utilizes an incremental SFM implementation [Schönberger and Frahm 2016], [Schönberger et al. 2016] to generate a dense point-cloud of the input scene. This implementation works as follows:

First, a correspondence search is performed in the input images to find overlaps. Then an incremental reconstruction is performed where, starting from an initial image pair, new images are iteratively registered into the scene to add additional data points for the final point-cloud output.

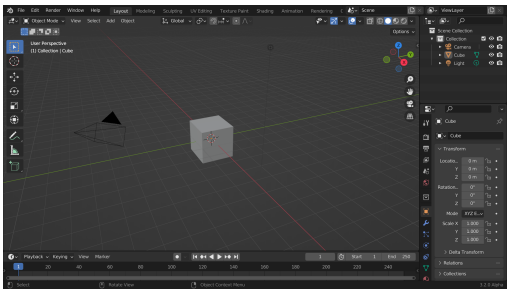
To integrate COLMAP into our multi-staged webservice, we created a Python wrapper to interface with the COLMAP commands. By not modifying the existing source code and, instead, building our own interface we can incorporate possible future COLMAP updates into our pipeline without the need of adapting the COLMAP source. Furthermore, due to this interface nature, exchanging the existing SFM step with a different implementation can be done seamlessly.

In contrast to COLMAP, which by itself also covers mesh reconstruction, our implementation does not use this feature. Instead, our final of step 1 is a dense point-cloud, which is then used as a basis for mesh reconstruction in step 2. The final output generated by out solution is a folder which contains the point-cloud in .PLY format as well as a log file.

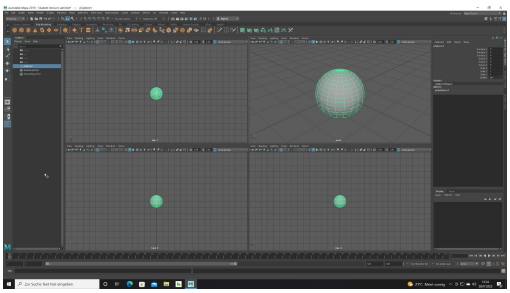
3.2 Mesh Reconstruction

Step 2 of our reconstruction pipeline is Mesh Reconstruction. We get the point-cloud files generated in step 1 and apply Screened Poisson Surface Reconstruction (SPSR) to them. This is done using the implementation of Meshlab[2]. SPSR is described in the following Section 3.2.1. Since we do not have any information about the input cloud, we just use the default arguments for this reconstruction instead of adjusting them specifically. For meshes with over one million faces we additionally apply the Quadric Edge Collapse Decimation Simplification to reduce the face count to one million. This algorithm is based on Quadric Error Metrics Simplifications by Garland and Heckbert [1998]. This reduces the memory usage of the resulting mesh significantly, which is especially important for the limited performance and data transfers of mobile devices.

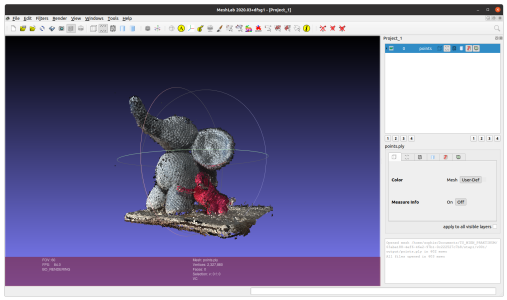
[1] https://colmap.github.io/
[2] http://www.meshlab.net/



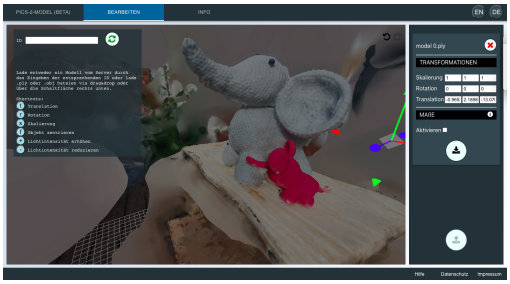
(a) Blender Layout



(b) Maya Layout



(c) Meshlab Layout



(d) Pix2Model Layout

Figure 2: Comparison of the layouts of Blender, Maya, MeshLab and Pix2Model

3.2.1 Screened Poisson Surface Reconstruction

Screened Poisson Surface Reconstruction by Kazhdan and Hoppe [Kazhdan and Hoppe 2013] is an algorithm to create watertight surfaces from oriented point clouds. This is done by first converting it into a vector field that describes the gradient of an indicator function. This function is positive inside the model and negative outside. To find this function, it's error is minimized by fitting b-splines into the leaves of an octree. The big advantage of SPSR over standard Poisson Surface Reconstruction is, that a term is added to the error function that discourages it's deviation from zero at the sampling locations, therefore sticking closer to the points of the point cloud. The resulting zero-set does then represent the models surface. After this reconstruction, some global corrections are applied to reduce the overall global.
SPSR is highly configurable with multiple parameters, but for our work, we stick to the default values, since not enough is known about the input clouds to better adjust any parameters.

3.3 Frontend

Figures 3 and 4 depict the user interface of the application.

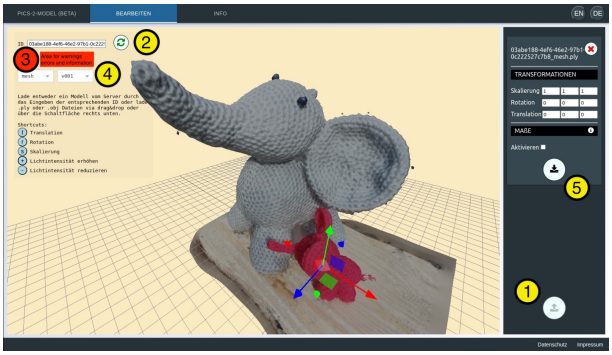
The frontend is implemented in React and the 3D editor is based on threejs. It is a single-page application, which enables the user to upload images (Pix2Model) - Figure 4 - that are further used to reconstruct the 3D mesh, to edit 3D models (EDIT) - Figure 3 - and to receive further information about the project and the team (INFO).

Pix2Model
The upload as depicted in figure 3 provides the option to initiate only the pointcloud reconstruction as described in Section 3.1 or both the pointcloud reconstruction and the mesh reconstruction, discussed in Section 3.2. Additionally the user can request an email notification after the upload has terminated.
After successfully uploading the images, the upload page displays a link to the edit page. The redirecting link will update the URL to contain the process ID and it will navigate the user to the edit page. The reconstruction takes several minutes, depending on the amount and resolution of the images. For 100 photos, the reconstruction takes about one hour. Therefore, the mesh won't be accessible immediately after redirecting to the edit page.

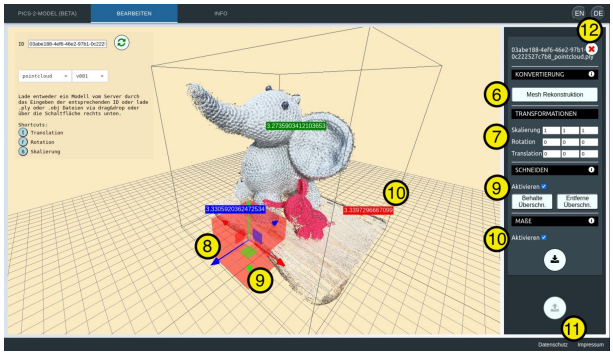
EDIT: Load and Download Models
The user can load models from his local machine or from the server. The former can be performed with drag&drop or by using the *Upload button* (Figure 3, 1). The latter is done by entering the ID to the ID textfield and pressing the *Refresh button* (Figure 3, 2). In case the user navigated to the Edit page via the the previously mentioned redirection link, the ID is already copied to the ID textfield. The model reconstruction might still be in progress, such that the UI displays an error message that no model is available for the requested ID (Figure 3, 3). To pull updates, the refresh button needs to be pressed actively again (figure 3, 2).
The requested reconstruction ID can be linked to multiple models. Initially, it points to the most recently created model, which is the output of step 1, in case that step 2 was not created (yet) or to the output of step 2 otherwise. The user can choose between *pointcloud* and *mesh* and different versions for those model types in the *dropdown menu* (Figure 3, 4).
The loaded mesh can be downloaded with the *download button* (Figure 3, 5).

EDIT: Version Up
The initial version is v000. Each new version is an increment of the previously created version. New meshes are created for a reconstruction ID, if a user loads a pointcloud (output of step 1) and presses the *reconstruct mesh button* (Figure 3, 6). This enables the user to make adjustments on a pointcloud before the reconstruction of the mesh. Those adjustments can include the removal of background points and transformations.

EDIT: Transformation



(a) Mesh Model: Editor Features



(b) Pointcloud Model: Editor Features

Figure 3: 3D Editor: (1) Upload model, (2) Load model from server, (3) Information, warnings and errors, (4) Version selection, (5) Download, (6) Create new version for the model to reconstruct mesh with current adjustments, (7) Transformations, (8) Gizmo which is attached to the model if the crop features is not activated and attached to the crop-box otherwise, (9) Crop feature, (10) Measurement feature, (11) Privacy and Impressum, (12) Change language

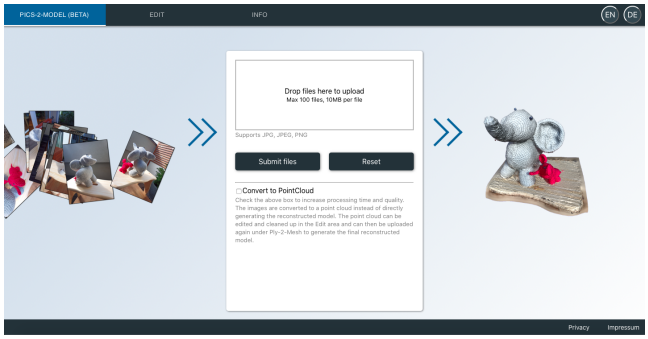


Figure 4: Upload images to start the reconstruction of a 3D model

Transformations are *scaling*, *rotation* and *translation*. They can be applied textually (Figure 3, 7) or by using the gizmo (Figure 3, 8). The keyboard shortcuts 'T' for Translation, 'R' for Rotation and 'S' for Scale can be used to switch between the Transformation modes.

EDIT: Crop
Cropping is only available for pointclouds. The crop functionality needs to be activated first. This spawns a crop-box that can be transformed to cover a subset of the points in the pointcloud (figure 3, 9). These points can either be removed or singled out. 3 shows the the cropbox, which intersects with the points that form the table that the figures stand on.

EDIT: Measurement
When the measurement feature is activated, the bounding box that includes all the points indicates the size of the 3D model and the labels attached to the bounding box display the exact values of the bounding box in each dimension (figure 3, 10).

INFO
Information about the project's goal, its usability and the team members are listed under *Info*. Privacy information and the Impressum as well as a support section that links to Github to file issues are always accessible on all subpages with the links in the bottom right corner (figure 3, 11). The user can toggle between English and German with the buttons in the top right corner (Figure 3, 12).

3.4 Backend

1. how calls to step 1 and 2 are started
2. data protection: UUID
3. microservices
4. redis
5. ssh

4 Comparison

compare Chamfer and Hausdorff distance of our results with raw COLMAP and Meshroom

user study comparing visual quality of results?

user study using the editor?

crowdworker for user study?

5 Conclusion

We did good.

6 Acknowledgements

Thanks to xxx for funding this project.

References

BARROW, H. G., TENENBAUM, J. M., BOLLES, R. C., AND WOLF, H. C. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, Science Applications, Inc Arlington, VA, 21–27.

GARLAND, M., AND HECKBERT, P. S. 1998. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of the Conference on Visualization '98*, IEEE Computer Society Press, Washington, DC, USA, VIS '98, 263–269.

HUTTENLOCHER, D. P., KLANDERMAN, G. A., AND RUCKLIDGE, W. J. 1993. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 9, 850–863.

KAZHDAN, M., AND HOPPE, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG) 32*, 3, 29.

SCHÖNBERGER, J. L., ZHENG, E., POLLEFEYS, M., AND FRAHM, J.-M. 2016. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*.

SCHÖNBERGER, J. L., AND FRAHM, J.-M. 2016. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4104–4113.

A Contributions

1. Philipp Erler: Project Management
2. Maximilian Riegler: Backend
3. Sophie Pichler: Frontend
4. Johannes Eschner: COLMAP
5. Florian Steinschorn: Reconstruction