

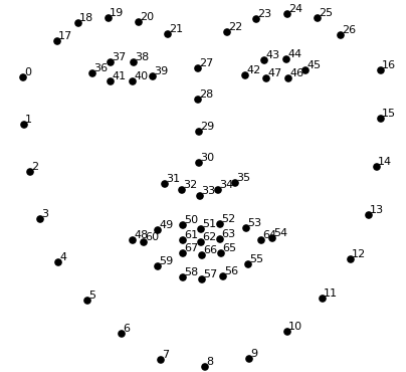
LMCface

Plugin for Rhino/Grasshopper

LMCfaces (Land Mark Creator - Face) is a plugin designed for Grasshopper in Rhino 8 for Windows. It takes in untextured mesh data from 3D scans of a user's face and outputs 68 landmarks as 3D points using the Dlib library. Input scan can be any un-textured mesh of a face and does not need 360 degree coverage. Clean ear to ear scans of a face with no back of head data works best.

This plugin is open source:

<https://github.com/PyottDesign/LMCface>



USECASE

This plugin was made for creating custom on demand generative products from face scan data in Grasshopper. I've used it for taking in 3D scan data captured using an iPhone's FaceID and making custom 3D printed VR face gaskets. I usually use the Heges 3D Scanning app on iPhone and export it as an STL file for my source scans. Currently this plugin will only work with Rhino 7 or 8 for Windows only.

PACKAGE MANAGER (YAK) DOWNLOAD IN RHINO

LMCface can be downloaded and installed directly in Rhino for Windows using the built-in package manager.

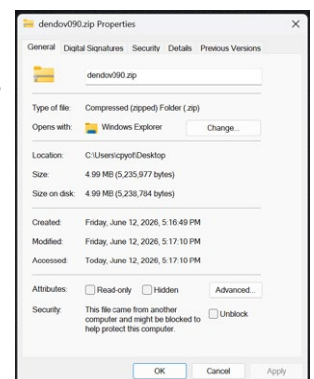
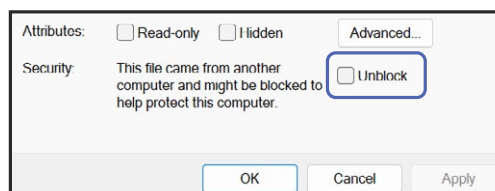
- Type "PackageManager" in Rhino
- Search for "LMCface"
- Select install

ZIP DOWNLOAD INSTALLATION

Alternatively, you can manually install the the plugin by downloading the zip file of it from Food4Rhino

- Download the ZIP file from **LMCface on Food4Rhino**
- Right click on the zip and select "Properties"
Windows 11: Right click and select "More Options" and then "Properties"
- Check "Unblock" box and "Ok"
- Copy files from zip to your Grasshopper Library folder
Usually: C:\Users\USERNAME\AppData\Roaming\Grasshopper\Libraries

You must Unblock the zip file before copying the files or the plugin will not work!



FACE LANDMARKS

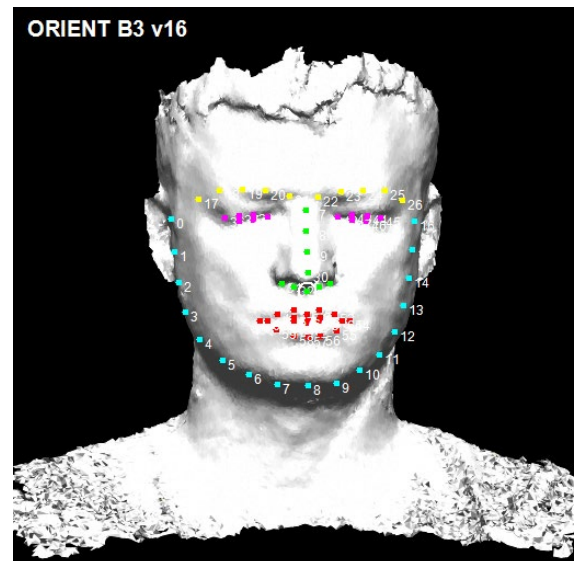
All landmarks are output to the standard 68 point Dlib structure (0-16: Jaw, 17-26: Brow, 27-35: Nose, 36-47: Eyes, 48-67: Mouth) as 3D points in Grasshopper. This allows for the creation of generative designs that can update based on face scan input inside GH.

PLUGIN LOGIC OVERVIEW

The plugin works by creating a bounding sphere around the input mesh. The sphere surface is populated with virtual cameras pointing inwards and images of the scan data are internally captured. The plugin goes through an initial low resolution image pass that attempts to locate and orient a face in the scene. This includes incrementally rotating the camera images if no face is detected (Dlib does not detect upside down faces on its own). Once detected, a series of high resolution images are rendered from cameras pointing straight at the scan and from the left and right sides. These images of the scan data are passed through Dlib for landmarks to be created on the 2D pictures. The points are then projected back onto the 3D model. This process is repeated and refined with all the cameras that detect a face and the landmark points are weighted averaged to create the final 68 point landmark set in 3D space.

The plugin can be thought of as working in two separate phases. An orientation phase that is only interested in locating a face in the scan data, creating a bounding box around that face and orienting it in the scene so it is upright, facing forward.

The second phase is the detailed capture and landmark generation where a series of high-resolution renderings are done to make and project accurate landmarks in 3D space. The second phase can't reliably happen until the first orientation phase is completed accurately.



LMCface (LMCf) COMPONENT

This is the base component for the plugin that takes in a 3D scan and outputs landmarks as a 3D point set.

INPUTS

Scan Mesh (M) - The 3D scan input mesh. Recommend a clean ear to ear headscan with no back of head scan data saved as an STL.

Resolution Final (RF) - Final resolution of capture images from virtual cameras around scan used to create landmarks. 1024 is a good starting number input for bust scans. "1024" input would capture a square 1024x124 pixel image sets. Higher resolution images have diminishing returns so don't go crazy.

Resolution Orientation (RO) - Lower resolution camera settings for initial face detection and orientation phase. 512 is good for busts but this number may need to be larger for full body scans. Like the RF input, there are diminishing returns with this input number.

Orientation Iterations (OI) - The number of times the face detection and orientation cycle runs without the final landmark generation phase. If your scanning workflow brings scans in upside down or your scan scene has significant data outside of the face, this may need to be run multiple times. If OI input is 0, the scan mesh will remain in its original position in the scene. If the OI was set to 1 or greater, the scan will be placed with the nose on 0,0,0 in the Rhino world space.

Settings (S) - This is the input for LMCf Settings. Usually can be left defaults (disconnected).

OUTPUTS

Mesh (M) - Oriented mesh output.

Landmarks (LM) - This is a list of the 68 generated landmark points placed on the scan as X,Y,Z coordinates (0-16: Jaw, 17-26: Brow, 27-35: Nose, 36-47: Eyes, 48-67: Mouth)

Transform (T) - The transformation vector of the original scan to the oriented position. Useful if you reduced the mesh input for the Landmark processing and want to orient the original high-resolution scan. Again, if OI input was set to 0, there will be no Transform vector.

Raw Points (RP) - All raw points used for calculating the final Landmark set. Can be used for debugging in your Grasshopper definition but otherwise ignored.

Confidence (C) - Score of confidence for raw points output in R. Confidence is based on a few factors but mainly comes down to the angle of the point projection onto the final scan. Perpendicular projections onto the scan are high confidence, extremely low angle "glancing" projections are low confidence. This data is also used for advanced scan analysis debugging and can be ignored in normal use.

LMCface Settings (LMCf Settings) COMPONENT

The advanced setting input for more detailed control and debugging of the plugin. This is an optional component that is used for fine tuning the base LMCface component.

Most of these inputs are for controlling the final image generation and Landmark calculation phase.

INPUTS

Cluster Radius (CR) - How far apart a group of Landmarks can be before they are considered part of another group. Default: 25 Rhino global units

Outlier Rejection (OR) - How aggressively outlier landmarks are discarded. Low values are strict with higher values being more lenient. Default: 2.0

View Angle (VA) - Angle on a 2D arc across the eyeline from center for the left and right camera camera clusters are for capturing the jawline landmarks. Default: 25.0°

View Cone (VC) - Distribution of cameras angle for front, left and right views. A tighter cone will keep the cameras clustered closer together. Default: 35.0°

Camera Number (CN) - Number of virtual cameras that populate the front, left and right camera view cones. Default 12

Light Angle (LA) - Controls the virtual lighting angle from the same position as a camera (0°) to directly overhead (90°). Default 30.0°

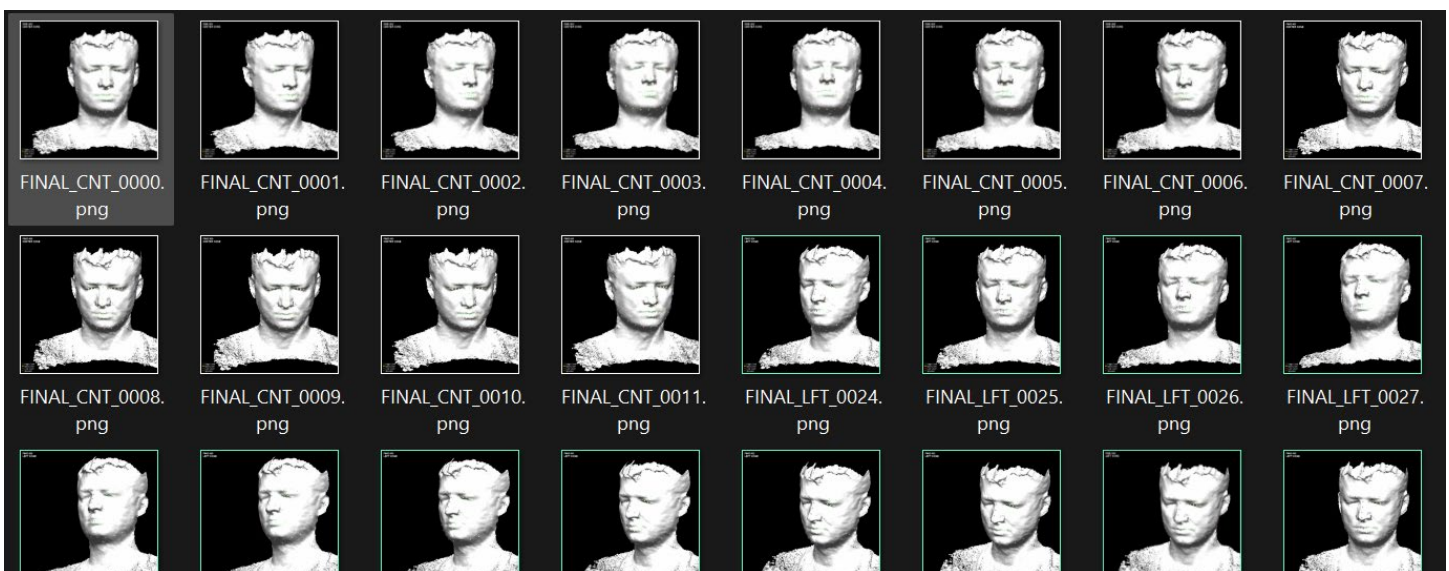
Light Intensity (LI) - How bright the light is for each rendering. Default: 1.5

Light Fill (LF) - Fill light amount for parts of the model in shadow. Default: 0.01

DebugImages (DBG) - When enabled, saves annotated render images from every camera pass to AppData/Local/Temp/LMCface_debug. This is a very effective tool for visually identifying where the plugin is struggling. Default: False (off)

OUTPUTS

Settings (S) - Settings data, connects to LMCf (S) input



LMC faces

KNOWN ISSUES

Dlib is a pretty old library and there are better ways of doing landmark generation with higher fidelity in 2026. That said, Dlib is very stable and was easy to implement in this case, so it worked well as a quick and dirty first go as a face landmark generator.

Dlib works well with black and white images as the input for face identification. This matches up mostly ok with untextured scan data that's captured with one key issue; finding the center of the eyes. Dlib usually uses the high contrast area of the pupil to localize the eye landmarks. This data is not available on untextured 3D scans so the eye landmarks (36-47) can get weird.

Recommend additional post processing logic for finding the eye centers if higher precision is needed.

This plugin also starts to really struggle and can crash at higher mesh face count scans. Highly recommend using the included ReduceMesh component and decimating your input scan mesh to under 100k faces. You can use the transform data out of LMCf to move the original scan into position and then Mesh Closest Point to snap the landmarks into position.

BASIC HD LANDMARK GENERATION DEFINITION

