README for Mesh Simplification Script

Overview

This script provides a straightforward way to simplify 3D mesh models by reducing the number of triangles, using the edge collapse method through the Open3D library. It's designed to be an easy-to-use tool for both 3D modeling enthusiasts and professionals who require efficient mesh simplification for their projects. The script loads an original mesh file, performs simplification to a target number of triangles, and then outputs both the original and simplified meshes for comparison and further use.

Features

Mesh simplification using the edge collapse method.

Easy specification of the target number of triangles for the simplified mesh.

Supports various mesh file formats (.ply, .obj, etc.).

Visual comparison between the original and simplified meshes.

Requirements

To run this script, you will need:

Python 3.x

Open3D library (can be installed via pip with pip install open3d)

Installation

Ensure you have Python 3.x installed on your system.

Install Open3D using pip:

pip install open3d

Usage

Place the script in your project directory.

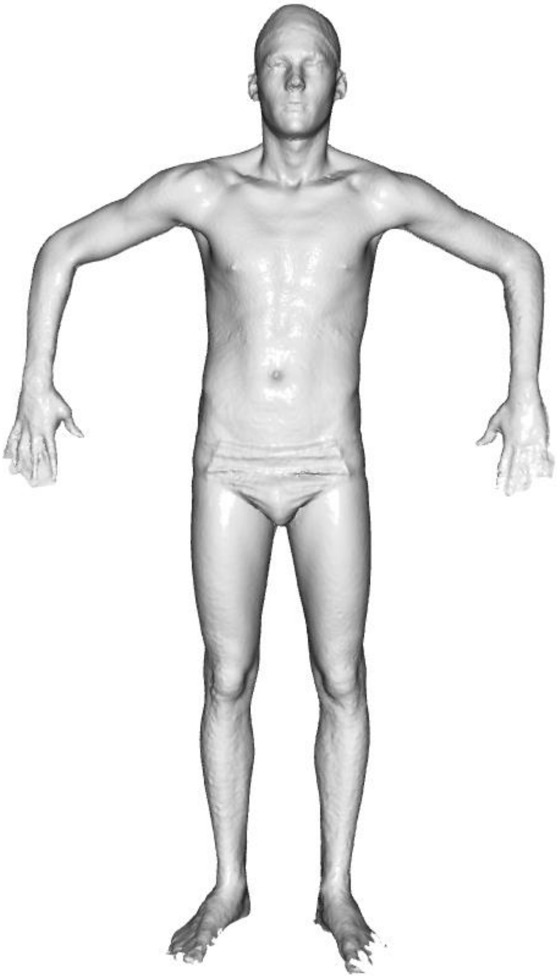
Modify the mesh_path variable in the script to point to your original mesh file. The path should be a string, e.g., r"path\to\your\meshfile.ply".

Set target_triangles to the desired number of triangles in your simplified mesh.

Run the script. Two windows will appear visualizing the original mesh and the simplified mesh, respectively.

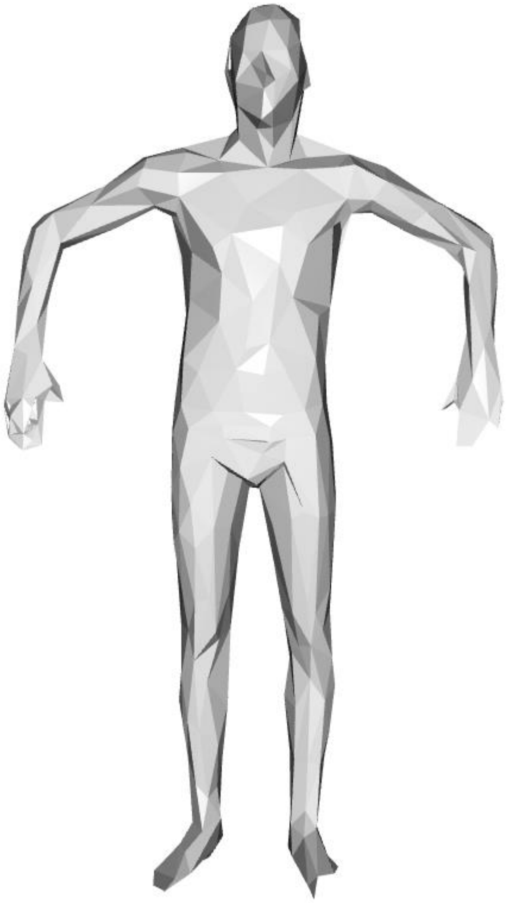
Results

The image below shows he original 3D model



\\

This image shown below is a 3D model of a human figure that has been rendered using a low-polygon or 'low-poly' mesh. This artistic style is characterized by a simplified geometry with a minimal number of polygons.



README for Vertex Clustering Mesh Simplification Script

Overview

This script is designed to simplify 3D mesh models by using a vertex clustering method. The script not only reduces the complexity of the mesh but also provides the option to visualize both the original and the simplified models. It is particularly useful for scenarios that require a balance between mesh detail and computational efficiency, such as in graphics rendering or in the preprocessing stages of 3D data analysis.

Features

Simplification of 3D meshes using vertex clustering.

Adjustable voxel size for fine-tuning the level of simplification.

Optional visualization of the original and simplified meshes for immediate comparison.

The script supports mesh files in formats readable by Open3D, such as .ply and .obj.

Requirements

Python 3.x

Open3D library

Installation

Make sure you have Python 3.x installed on your machine.

Install the Open3D library using pip if you haven't already:

pip install open3d

Usage

Save the script in your desired directory.

Change the mesh_path variable to the path of your mesh file, ensuring it is in the correct format (e.g., .ply or .obj).

Set the voxel_size parameter as needed to adjust the simplification level. A smaller voxel size retains more detail, while a larger size results in a more simplified mesh.

Run the script. If visualize is set to True, the original and simplified meshes will be displayed in separate windows.

Results

The image below shows a 3D model human figure represented in a Vertex Clustering Mesh Simplification, similar style is marked by the use of a relatively small number of large flat polygons, which gives the model a faceted look. This kind of model is often used where real-time rendering performance is a priority, such as in video games or interactive applications, because it requires less computational power to render compared to a high-polygon model.