

11/28/25, 7:58 AM

computervision-final-prep/lab/Lab 12 (3D Vision)-20251128/lab12.2\_working\_with\_Mesh.ipynb at main · limhpone/computervision-...



limhpone / computervision-final-prep



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

In

computervision-final-prep / lab / Lab 12 (3D Vision)-20251128

/ lab12.2\_working\_with\_Mesh.ipynb



limhpone 3D Vision- lab 12

daadd1a · 2 hours ago



279 lines (279 loc) · 8.41 KB

Preview

Code

Blame



Raw



# Mesh Geometry

Tutorial adopted from this [link](#)

```
In [1]: install_open3d = False

try:
    import open3d
except ModuleNotFoundError:
    install_open3d=True

if install_open3d:
    !pip install open3d
```

```
Jupyter environment detected. Enabling Open3D WebVisualizer.
[Open3D INFO] WebRTC GUI backend enabled.
[Open3D INFO] WebRTCWindowSystem: HTTP handshake server disabled.
```

```
In [2]: import open3d as o3d
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

```
In [3]: print("Testing mesh in Open3D...")

# Loading the Armadillo mesh
armadillo_mesh = o3d.data.ArmadilloMesh()
mesh = o3d.io.read_triangle_mesh(armadillo_mesh.path)

# Loading the Knot mesh
knot_mesh = o3d.data.KnotMesh()
mesh = o3d.io.read_triangle_mesh(knot_mesh.path)

# Printing the mesh
print(mesh)
print('Vertices:')
print(np.asarray(mesh.vertices))
print('Triangles:')
print(np.asarray(mesh.triangles))
```

```
Testing mesh in Open3D...
[Open3D INFO] Downloading https://github.com/isl-org/open3d_downloads/release
s/download/20220201-data/ArmadilloMesh.ply
[Open3D INFO] Downloaded to C:\Users\rbipl/open3d_data/download/ArmadilloMesh/
ArmadilloMesh.ply
[Open3D INFO] Downloading https://github.com/isl-org/open3d_downloads/release
s/download/20220201-data/KnotMesh.ply
[Open3D INFO] Downloaded to C:\Users\rbipl/open3d_data/download/KnotMesh/KnotM
esh.ply
TriangleMesh with 1440 points and 2880 triangles.
Vertices:
[[ 4.51268387  28.68865967 -76.55680847]
 [ 7.63622284  35.52046967 -69.78063965]
```

```
computervision-final-prep/lab/Lab 12 (3D Vision)-20251128/lab12.2_working_with_Mesh.ipynb at main · limhphone/computervision-...
[ 6.21986008 44.22465134 -64.82303619]
...
[-22.12651634 31.28466606 -87.37570953]
[-13.91188431 25.4865818 -86.25827026]
[-5.27768707 23.36245346 -81.43279266]]
Triangles:
[[ 0 12 13]
 [ 0 13 1]
 [ 1 13 14]
 ...
 [1438 11 1439]
 [1439 11 0]
 [1439 0 1428]]
```

The `TriangleMesh` class has a few data fields such as `vertices` and `triangles`. Open3D provides direct memory access to these fields via numpy.

## Visualize the 3d mesh

```
In [5]: print("Try to render a mesh with normals (exist: " +
      str(mesh.has_vertex_normals()) + ") and colors (exist: " +
      str(mesh.has_vertex_colors()) + ")")
o3d.visualization.draw_geometries([mesh])
print("A mesh with no normals and no colors does not look good.")
```

Try to render a mesh with normals (exist: False) and colors (exist: False)  
A mesh with no normals and no colors does not look good.

You can rotate and move the mesh but it is painted with uniform gray color and does not look 3d. The reason is that the current mesh does not have normals for vertices or faces. So uniform color shading is used instead of a more sophisticated Phong shading.

## Surface normal estimation

Let's draw the mesh with surface normals.

```
In [6]: print("Computing normal and rendering it.")
mesh.compute_vertex_normals()
print(np.asarray(mesh.triangle_normals))
o3d.visualization.draw_geometries([mesh])
```

Computing normal and rendering it.  
[[ 0.79164373 -0.53951444 0.28674793]
 [ 0.8319824 -0.53303008 0.15389681]
 [ 0.83488162 -0.09250101 0.54260136]
 ...
 [ 0.16269924 -0.76215917 -0.6266118 ]
 [ 0.52755226 -0.83707495 -0.14489352]
 [ 0.56778973 -0.76467734 -0.30476777]]

## Crop mesh

In [ ]:

```
import copy
print("We make a partial mesh of only the first half triangles.")
mesh1 = copy.deepcopy(mesh)

# Crop the mesh by keeping only half of the triangles

mesh1.triangles = o3d.utility.Vector3iVector(
    np.asarray(mesh1.triangles)[:len(mesh1.triangles) // 2, :])

mesh1.triangle_normals = o3d.utility.Vector3dVector(
    np.asarray(mesh1.triangle_normals)[:len(mesh1.triangle_normals) // 2, :])

print(mesh1.triangles)
o3d.visualization.draw_geometries([mesh1])
```

We make a partial mesh of only the first half triangles.