UPC
Universidad Peruana
de Ciencias Aplicadas

Ciencias de la Computación 2022-02

Machine Learning - CC57

Trabajo

Investigación de modelos de representación de objetos 3D, para impresión 3D asociado con Machine Learning

Integrantes :

Daniel Carlos Guillén Rojas - U201920113

Sebastián Alonso Gonzales Sotomayor - U201923816

Antonio Salinas Roca - U201924931

Sección:

CC72

Profesor :

Luis Martín Canaval Sánchez

Septiembre, 2022

Índice

1. Introducción

La impresión 3D es una tecnología creada con el fin de obtener creaciones hechas en una computadora en la vida real. Asimismo, esta tecnología influyó positivamente en la reconstrucción de materiales dañados por el paso del tiempo. Tomando en cuenta de esto, en el paper *3D Reconstruction of Incomplete Archaeological Objects Using a Generative Adversarial Network,* elaborado por Renato Hermosa e Ivan Sipiran (2018), nos brindan información de como el machine learning beneficia este suceso de manera categórica. Por eso, en el presente proyecto se brindará información con respecto a los formatos de objetos 3D y cómo se podrá implementar esta información para lograr la impresión de estos objetos.

2. Formatos de los modelos

a. Formato OFF (Object File Format)

Los archivos que terminan en extensión OFF son objetos 3D o 2D basados en texto ASCII que sirve para describirlos. Estos archivos definen sus superficies geométricas mediante polígonos y vértices. Asimismo, la coloración de los objetos se rige mediante los valores RGB. La información que nos brinda es de manera organizada según el siguiente orden:

- La primera línea contiene la palabra OFF..
- La segunda línea muestra el número de vértices, seguido del número de caras y luego el número de bordes del objeto.
- Desde la tercera línea se muestran ***"n"*** filas con las respectivas coordenadas X,Y,Z de cada vértice.
- Desde la enésima línea siguen ***"m"*** líneas dando la información de las caras.
- Por último, desde la emésima línea se muestran ***"l"*** líneas dando información de los bordes.

```
OFF
8 12 0
1 1 -1
1 -1 -1
1 1 1
1 -1 1
-1 1 -1
-1 -1 -1
-1 1 1
-1 -1 1
3 0 4 6  220 220 220
3 6 2 0  220 220 220
3 3 2 6  220 220 220
3 6 7 3  220 220 220
3 7 6 4  220 220 220
3 4 5 7  220 220 220
3 5 1 3  220 220 220
3 3 7 5  220 220 220
3 1 0 2  220 220 220
3 2 3 1  220 220 220
3 5 4 0  220 220 220
3 0 1 5  220 220 220
```

Ejemplo de formato **OFF** de un cubo

b. Formato OBJ

Este tipo de formato fue desarrollado por Wavefront Technologies, esta extensión nos comenta que es un archivo tipo imagen 3D. En estos tipos de archivos se almacenan datos importantes del objeto, como la posición de los vértices, el vector normal de los vértices y las caras del objeto.

La composición para la lectura de los archivos OBJ pueden llegar a ser algo dificultosos, pero conociendo la composición de las columnas se simplifica de gran manera.

- Como primer valor aparece la letra v, esta variable representa la posición de cada vértice.
- Continua la variable vt, esta nos identifica coordenadas UV.
- También la variable vn aparece, esta nos muestra el vector normal.
- Por último la variable f identifica la creación de las caras del objeto con polígonos de mínimo 3 vértices, y cada vértice con su respectiva coordenada UV y vector normal de manera opcional.

```
# Blender v2.93.6 OBJ File: ''
# www.blender.org
mtllib untitled.mtl
o Cube
v 1.000000 1.000000 -1.000000
v 1.000000 -1.000000 -1.000000
v 1.000000 1.000000 1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 1.000000 -1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 1.000000
v -1.000000 -1.000000 1.000000
vt 0.625000 0.500000
vt 0.875000 0.500000
vt 0.875000 0.750000
vt 0.625000 0.750000
vt 0.375000 0.750000
vt 0.625000 1.000000
vt 0.375000 1.000000
vt 0.375000 0.000000
vt 0.625000 0.000000
vt 0.625000 0.250000
vt 0.375000 0.250000
vt 0.125000 0.500000
vt 0.375000 0.500000
vt 0.125000 0.750000
vn 0.0000 1.0000 0.0000
vn 0.0000 0.0000 1.0000
vn -1.0000 0.0000 0.0000
vn 0.0000 -1.0000 0.0000
vn 1.0000 0.0000 0.0000
vn 0.0000 0.0000 -1.0000
usemtl Material
s off
f 1/1/1 5/2/1 7/3/1 3/4/1
f 4/5/2 3/4/2 7/6/2 8/7/2
f 8/8/3 7/9/3 5/10/3 6/11/3
f 6/12/4 2/13/4 4/5/4 8/14/4
f 2/13/5 1/1/5 3/4/5 4/5/5
f 6/11/6 5/10/6 1/1/6 2/13/6
```

Ejemplo de formato **OBJ** de un cubo

c. Formato STL

Conocido también como *Standard Triangle Language* o *Standard Tessellation Language*, este formato fue creado por la compañía de tecnologías 3D *3D Systems* en 1987 y es nativo de sus software CAD de estereolitografía. El formato funciona empleando teselado triangular para lo cual almacena información sobre las coordenadas de los vértices de cada triángulo. Cada archivo de dicho formato se encuentra compuesto por una descripción de la geometría de una superficie u objeto 3D a partir de cadenas de triángulos conectados. Actualmente se usa en la impresión 3D, donde es un formato reconocido universalmente y en prototipado rápido. El formato es ligero y veloz, sin embargo no trabaja bien con colores y texturas ni tampoco incluye metadata.

Un archivo STL sigue el siguiente formato:

i. Inicia con la línea:
   solid name
   Donde name es el nombre del objeto a crear

ii. El cuerpo contiene declaraciones de los triángulos mencionados anteriormente en la forma:
   facet normal n_i n_j n_k
   outer loop
   vertex $v1_x$ $v1_y$ $v1_z$
   vertex $v2_x$ $v2_y$ $v2_z$
   vertex $v3_x$ $v3_y$ $v3_z$
   endloop
   endfacet
   Donde n y v son números de tipo punto flotante.

iii. Finalmente el archivo se finaliza con la declaración el fin del objeto de la forma:
   endsolid name

Cuadro 1: Cuadro comparativo entre los formatos OFF, OBJ y STL

	OFF	OBJ	STL
Antigüedad	1986	1990	1987
Utilidad	Formato creado originalmente para visualización, describe los polígonos de un objeto 2D o 3D, registrando el número de vértices, caras y bordes	Formato de visualización e impresión de imágenes 3D. Describe la geometría 3D a través de la posición de los vectores, UV mapping y texturas	Formato de impresión y CAD el cual describe un modelo empleado teselado triangular

Ventajas	-Fácil de procesar en código -Netamente numérico	-Incluye color -Incluye texturas	-Fácil de comprender -Ligero
Desventajas	- No incluye color -No incluye texturas	-Las coordenadas no contienen unidades -Difícil de comprender	- No incluye color -No incluye texturas

3. Transformaciones y conversiones de formatos

Para la decisión de qué formato sería mejor para su transformación al formato STL tomamos como referencia al dataset brindado en el paper elaborado por Hermoza y Sipiran. El primer grupo llamado ModelNet10 contiene objetos caseros, como sillas, mesas, camas, jarrones, etc. El segundo grupo llamado 3D Pottery, contiene los objetos con los cuales trabajaremos, cerámicas. Como el objetivo principal es lograr una reconstrucción de un objeto mediante el uso del machine learning y nos enfocaremos principalmente en cerámicas, optamos a elegir el segundo grupo de dataset para entrenar mejor nuestro equipo. Por ese motivo se eligió el formato OBJ como el tipo de archivo preferencial para implementar funciones de conversión a STL.

4. Codificación de funciones para la transformación de formatos

Repositorio: https://github.com/antoniosalinas2000/MachineLearning-TA1
Archivo: obj_converter.py

```python
@dataclass
class Obj:
    positions: list[list[float]]
    normals: list[list[float]]
    tcoords: list[list[float]]
    faces: list[list[list[int]]]

    def __repr__(self):
        return f"Wavefront\n(v:{len(self.positions)}, vn:{len(self.normals)}, vt:{len(self.tcoords)}, f:{len(self.faces)})"

    def to_stl(self) -> Stl:
        stl = Stl.generate_empty()

        for face in self.faces:
            face = list(zip(*face))
            v1, v2, v3 = [Vec3(*self.positions[a]) for a in face[0]]
            n = Vec3(*self.normals[face[2][0]])

            stl.add_triangle_normal(n, v1, v2, v3)

        return stl
```

5. Bibliografía

FILExt (s.f.). Todo sobre los archivos OFF. Recuperado de: https://filext.com/es/extension-de-archivo/OFF [Consulta: 27 de septiembre de 2022].

Hermoza, R. & Sipiran, I. (2018). 3D reconstruction of incomplete archaeological objects using a generative adversarial network. Recuperado de: https://dl.acm.org/doi/abs/10.1145/3208159.3208173 [Consulta: 27 de septiembre de 2022].