import numpy as np

import trimesh

# Function to embed data into 3D model

def embed\_data\_into\_model(model\_path, data):

# Load the 3D model

mesh = trimesh.load(model\_path)

# Convert data into binary format

binary\_data = ''.join(format(ord(char), '08b') for char in data)

# Embed data into mesh vertices

for i, char in enumerate(binary\_data):

if i < len(mesh.vertices): # Ensure we don't exceed the number of vertices

if char == '1':

mesh.vertices[i] += 0.001 # Example manipulation for embedding 1

else:

mesh.vertices[i] -= 0.001 # Example manipulation for embedding 0

# Save the modified mesh

mesh.export(model\_path.replace('.obj', '\_steganographic.obj'))

# Function to extract data from 3D model

def extract\_data\_from\_model(modified\_model\_path):

# Load the modified 3D model

modified\_mesh = trimesh.load(modified\_model\_path)

# Extract data from mesh vertices

extracted\_data = ''

for vertex in modified\_mesh.vertices:

if vertex[0] > 0: # Example condition for extracting 1

extracted\_data += '1'

else:

extracted\_data += '0'

# Convert binary data back to characters

extracted\_text = ''.join(chr(int(extracted\_data[i:i+8], 2)) for i in range(0, len(extracted\_data), 8))

return extracted\_text

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

# Path to the 3D model

model\_path = 'example\_model.obj'

# Data to be embedded

data\_to\_embed = "Hello, this is a secret message!"

# Embed data into the 3D model

embed\_data\_into\_model(model\_path, data\_to\_embed)

# Extract data from the modified 3D model

extracted\_data = extract\_data\_from\_model('example\_model\_steganographic.obj')

print("Extracted data:", extracted\_data)