International Journal of Advance and Applied Research
www.ijaar.co.in
ISSN – 2347-7075 Impact Factor – 8.141
Peer Reviewed Bi-Monthly
Vol. 7 No. 5 January – February - 2026

Improving Existing Algorithm Using Artificial Intelligence: Generating Three Dimensional Curves and Surfaces

Mr. Gorakhanath Rambhau Karade
Abasaheb Marathe Arts and New Commerce, Science College Rajapur (Vikhare-Gothane), Tal: Rajapur Dist: Ratnagiri
DOI - 10.5281/zenodo.18897583

Abstract:

Three-dimensional (3D) curve and surface generation is an important area of research, having various applications, such as computer-aided design, robotics, and virtual reality. The generation of 3D objects is currently done by algorithms, which require manual human interaction, thereby taking up more time. This paper discusses the utilization of artificial intelligence (AI) to enhance current algorithms that specifically focus on 3D curve and surface generation. Recent advances involving the usage of generative adversarial networks (GANs), diffusion models, and neural radiance fields (NeRF) will also be reviewed, mentioning its advantages as well as limitations.

Keywords: Sphere, Cone, Cylinder Python Programming.

Introduction:

The generation of three-dimensional curves and surfaces is a complex operation that demands great knowledge and computing capabilities. Conventional approaches need manual human intervention. Artificial intelligence-based three-dimensional generation is one of the most effective solutions for such problems. Advances in artificial intelligence have made it possible to build sophisticated models that efficiently produce natural three-dimensional models.

Methodologies:

Some of the AI-driven methodologies that have been conceptualized for generating 3D output include:

Generative Adversarial Networks (GANs): GANs entail the use of two neural networks that combine to produce 3D models. GANs have been utilized in the creation of 3D models of objects such as cars and buildings.

Diffusion Models: Diffusion models employ a technique called score distillation sampling (SDS) to produce 3D models. Diffusion models have been utilized to produce high-quality 3D shapes.

Neural Radiance Fields: A continuous volume representation of a 3D scene.

It has been applied in creating realistic 3D models:

The volumetric nature accounts for both position and

Benefits and Risks

Product Name

Some of the advantages of using AI in the creation of three-dimensional images include:

Increased Efficiency: An AI-driven 3D creation system can potentially remove the need for human input in creation.

Enhanced Realism: The use of AI in 3D design and generation enables the production of realistic 3D images similar to those designed by humans.

Customization Through AI:

Flexibility: The capability of AI-generated 3D generation can be applied for the generation of various 3D entities, ranging from simple objects to complex environments.

However, there also exist challenges in employing AI in 3D generation. Some of these challenges include:

Data Quality: The generation of 3D using AI should have high-quality data in order for it to look realistic.

Computational Resources: It may require considerable computational resources to compute the generated image by the artificial intelligence capabilities of 3D image creation.

Applications:

The applications of AI 3D generation range from:

AI-assisted 3D reconstruction in:

Computer-Aided Design: 3D generation by computers can be harnessed by artificial intelligence to create designs automatically, thereby reducing human input.

Robotics: Artificial intelligence algorithms in 3D generation could be utilized in applications that require generation of three-dimensional models of robots and their settings.

Virtual Reality: Artificial Intelligence enabled 3D generation can be applied for the creation of realistic 3D models for virtual reality.

1. Sphere:

A sphere is a three-dimensional object that is perfectly round and symmetrical about its center. The mathematical equation for a sphere is:

x = r * sin(u) * cos(v)
y = r * sin(u) * sin(v)
z = r * cos(u)

where:

- (x, y, z) are the coordinates of a point on the surface of the sphere
- r is the radius of the sphere
- u is the angle between the positive z-axis and the line segment connecting the origin to the point (polar angle)
- v is the angle between the positive x-axis and the projection of the line segment onto the xy-plane (azimuthal angle)

Using Algorithm in Python Programming (AI Generated):

```
# Define the sphere's parameters
   u = np.linspace(0, 2 * np.pi, 100)
   v = np.linspace(0, np.pi, 100)
# Create a meshgrid of the parameters
   x = 10 * np.outer(np.cos(u), np.sin(v))
   y = 10 * np.outer(np.sin(u), np.sin(v))
   z = 10 * np.outer(np.ones(np.size(u)), np.cos(v))
# Create a 3D plot
   fig = plt.figure(figsize=(8, 8))
   ax = fig.add_subplot(111, projection='3d')
   # Plot the surface of the sphere
   ax.plot_surface(x, y, z, color='b')
# Set the plot's title and labels
   ax.set_title('3D Sphere')
   ax.set_xlabel('X')
   ax.set_ylabel('Y')
   ax.set_zlabel('Z')
# Display the plot
   plt.show()
```

2. Torus:

A torus is a doughnut-shaped object. The mathematical equation for a torus is:

x = (R + r * cos(u)) * cos(v)
y = (R + r * cos(u)) * sin(v)
z = r * sin(u)

where:

- (x, y, z) are the coordinates of a point on the surface of the torus
- R is the major radius of the torus (distance from the center of the torus to the center of the tube)
- r is the minor radius of the torus (radius of the tube)
- u is the angle around the tube (poloidal angle)
- v is the angle around the torus (toroidal angle)

Using Algorithm in Python Programming (AI Generated) [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Define the parameters of the torus
R1 = 3  # major radius
R2 = 1  # minor radius
# Create a meshgrid of the parameters
u = np.linspace(0, 2 * np.pi, 100)
v = np.linspace(0, 2 * np.pi, 100)
u, v = np.meshgrid(u, v)
# Calculate the x, y, and z coordinates of the torus
x = (R1 + R2 * np.cos(v)) * np.cos(u)
y = (R1 + R2 * np.cos(v)) * np.sin(u)
z = R2 * np.sin(v)
# Create a 3D plot
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
# Plot the surface of the torus
ax.plot_surface(x, y, z, color='b', alpha=0.5)
# Set the plot's title and labels
ax.set_title('3D Torus')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
# Display the plot
plt.show()
```

3. Ellipsoid:

An ellipsoid is a three-dimensional object that is symmetrical about its three axes. The mathematical equation for an ellipsoid is:

x = a * sin(u) * cos(v)
y = b * sin(u) * sin(v)
z = c * cos(u)

where:

- (x, y, z) are the coordinates of a point on the surface of the ellipsoid
- a, b, and c are the lengths of the three axes of the ellipsoid
- u is the angle between the positive z-axis and the line segment connecting the origin to the point (polar angle)
- v is the angle between the positive x-axis and the projection of the line segment onto the xy-plane (azimuthal angle)

Using Algorithm in Python Programming (AI Generated) [2]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Define the parameters of the ellipsoid
u = np.linspace(0, 2 * np.pi, 100)
v = np.linspace(0, np.pi, 100)
# Create a meshgrid of the parameters
x = 2 * np.outer(np.cos(u), np.sin(v))
y = 1.5 * np.outer(np.sin(u), np.sin(v))
z = 1 * np.outer(np.ones(np.size(u)), np.cos(v))
# Create a 3D plot
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
# Plot the surface of the ellipsoid
ax.plot_surface(x, y, z, color='b')
# Set the plot's title and labels
ax.set_title('3D Ellipsoid')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
# Display the plot
plt.show()
```

4. Cylinder:

A cylinder is a three-dimensional object with two parallel and circular bases connected by a curved surface. The mathematical equation for a cylinder is:

x = r * cos(v)
y = r * sin(v)
z = u

where:

- (x, y, z) are the coordinates of a point on the surface of the cylinder
- r is the radius of the cylinder
- u is the height of the point above the xy-plane
- v is the angle around the cylinder (azimuthal angle)

Using Algorithm in Python Programming (AI Generated) [4]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Define the parameters of the cylinder
h = 5  # height of the cylinder
r = 2  # radius of the cylinder
# Create a meshgrid of the parameters
z = np.linspace(0, h, 100)
theta = np.linspace(0, 2 * np.pi, 100)
z, theta = np.meshgrid(z, theta)
# Calculate the x, y, and z coordinates of the cylinder
x = r * np.cos(theta)
y = r * np.sin(theta)
# Create a 3D plot
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
# Plot the surface of the cylinder
ax.plot_surface(x, y, z, color='b', alpha=0.5)
# Set the plot's title and labels
ax.set_title('3D Cylinder')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
# Display the plot
plt.show()
```

Conclusion:

AI-powered 3D generation is a rapidly developing area that can lead to a revolution in the generation of 3D models. The latest developments in AI-powered 3D generation have allowed the creation of a powerful generative model that can produce realistic 3D models. Even though some difficulties are associated with the use of AI-powered 3D generation, the advantages are many. We are of the view that AI-powered 3D generation will play a crucial role in determining the future of 3D modelling and animation.

References:

1. https://ch.mathworks.com/matlabcentral/answers/2034534-how-can-i-generate-a-3d-torus-taking-a-function-as-the-torus-shape?s_tid=prof_contriblnk
2. https://matplotlib.org/stable/users/explain/toolkits/mplot3d.html
3. https://ch.mathworks.com/matlabcentral/answers/62894-trying-to-plot-a-3d-closed-cylinder?s_tid=ta_ans_results
4. Superintelligence: Paths, Dangers, Strategies by Nick Bostrom, published in 2014. This book explores the potential risks and benefits of advanced artificial intelligence.
5. Life 3.0: Being Human in the Age of Artificial Intelligence by Max Tegmark, published in 2017. This book discusses the possibilities and implications of artificial intelligence on human society.
6. AI Superpowers: China, Silicon Valley, and the New World Order by Kai-Fu Lee, published in 2018. This book provides insights into the current state of AI research and development in China and Silicon Valley.
7. The Alignment Problem: Machine Learning and Human Values by Brian Christian, published in 2020. This book examines the challenges of aligning AI systems with human values and ethics.
8. Co-Intelligence: The Definitive, Bestselling Guide to Living and Working with AI by an unknown author, but highly recommended, published in 2024. This book offers guidance on how to work effectively with AI systems