VoxelForge

Release 0.1

Andrew Garcia

Aug 15, 2024

CONTENTS

VoxelForge

VoxelForge is a high-performance Python package designed for creating and manipulating voxel models, with advanced spatial data structures for ML and deep learning integration.

Contents:

CHAPTER

ONE

INSTALLATION

VoxelForge can be easily installed using pip. Ensure you have Python and pip installed on your system, then run the following command:

```
pip install VoxelForge
```

This command will download and install VoxelForge and all required dependencies.

CHAPTER TWO

USAGE

2.1 Basic Voxel Operations

To start using **VoxelForge**, import the package and create a *VoxelGrid* instance:

```python
import voxelforge as vf

# Create a VoxelGrid and add voxels
grid = vf.VoxelGrid()
grid.addVoxel(1, 2, 3)
grid.addVoxel(4, 5, 6, "String identity")
grid.addVoxel(7, 8, 9, {"key": "value"})

voxels = grid.getVoxels()
for voxel in voxels:
    print(f'Voxel at ({voxel.x}, {voxel.y}, {voxel.z}) with data {voxel.data}')
```

2.2 Advanced Graph Features

VoxelForge also supports transforming voxel data into graph structures, useful for graph-based machine learning models:

```python
graph_data = grid.toTorchGraph(xDim=10, yDim=10, zDim=10, neighboring_radius=1.0)
print("Node Features:", graph_data['x'])
print("Edge Index:", graph_data['edge_index'])
```

CHAPTER

THREE

API REFERENCE

The API reference details all classes, methods, and their functions:

3.1 Voxel

3.2 VoxelGrid

CHAPTER

FOUR

EXAMPLES

4.1 Example 1: Basic Voxel Manipulation

This example demonstrates how to create a VoxelGrid and add voxels with various identities:

```
# Import the VoxelForge package
import voxelforge as vf

# Create a new VoxelGrid instance
grid = vf.VoxelGrid()

# Add several voxels with default and custom identities
grid.addVoxel(1, 2, 3)  # Default identity (integer)
grid.addVoxel(4, 5, 6, "Colorful Voxel")  # String identity
grid.addVoxel(7, 8, 9, {"key": "value"})  # Dictionary identity
grid.addVoxel(10, 11, 12, 3.14159)  # Floating-point identity

# Print out the voxel information
voxels = grid.getVoxels()
for voxel in voxels:
    print(f'Voxel at ({voxel.x}, {voxel.y}, {voxel.z}) with data: {voxel.data}')
```

4.2 Example 2: Using Octrees for Spatial Indexing

Here we show how to initialize an Octree, insert points, and locate nodes:

```
# Import the VoxelForge package and numpy for handling coordinates
import voxelforge as vf
import numpy as np

# Initialize an Octree with a specific origin, size, and maximum depth
origin = np.array([0.0, 0.0, 0.0])
size = 50.0
max_depth = 4
octree = vf.Octree(origin, size, max_depth)

# Insert points into the Octree
points = [
    np.array([5.0, 5.0, 5.0]),
```

(continues on next page)

(continued from previous page)

```
    np.array([15.0, 15.0, 15.0]),
    np.array([35.0, 35.0, 35.0]),
    np.array([45.0, 45.0, 45.0])
]

for point in points:
    octree.insert_point(point)

# Locate a specific point and print its information
target_point = np.array([5.0, 5.0, 5.0])
leaf_node = octree.locate_leaf_node(target_point)
if leaf_node:
    print(f"Leaf node found at {leaf_node.get_point()}")
else:
    print("No leaf node found at the specified location.")
```

CHAPTER FIVE

FREQUENTLY ASKED QUESTIONS

1. **What is VoxelForge?**

   VoxelForge is a Python package for efficient voxel and mesh model creation with a focus on integration in machine learning workflows.

2. **How can I contribute to VoxelForge?**

   Contributions can be made via our GitHub repository, whether as feature suggestions, bug reports, or pull requests.

CHAPTER SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX