

УДК 537:53.06]:004.9

ПРОГРАМНИЙ МЕТОД ДЛЯ МОДИФІКАЦІЇ 3D-МОДЕЛЕЙ БЕЗ ЗМІНИ ФІЗИЧНИХ ХАРАКТЕРИСТИК

Левченко Є. В., Кравчук О.О.

Науковий керівник – проф., д.ф.-м.н., зав. каф. МЕЕПП Бондаренко І. М.,
Харківський національний університет радіоелектроніки, каф. МЕЕПП,
м. Харків, Україна

тел. +38(095) 683-29-02, e-mail: yevhenii.levchenko@nure.ua

In the field of 3D modeling, it is often necessary to modify the internal structure of a 3D model without changing its physical characteristics, such as volume or surface area. This can be a challenging task, especially when dealing with complex geometries and meshes. However, with the use of program methods and algorithms, it is possible to achieve such modifications with relative ease.

Однією з поширених технік, які використовуються в цьому методі, є параметричне моделювання. У параметричному моделюванні геометрія моделі визначається набором параметрів, які можна змінювати для досягнення бажаного результату. Наприклад, параметрична модель автомобіля може мати такі параметри, як довжина, ширина і висота автомобіля, розмір коліс і розташування фар. Змінюючи ці параметри, програма може створювати варіанти моделі без зміни її фізичних характеристик.

Приклад реалізації в Python для зміни внутрішньої структури 3D-моделі з твердого тіла на структуру з маленькими сферами всередині наведено на рисунку 1.

У цій реалізації ми спочатку визначаємо функцію `create_sphere`, яка створює новий об'єкт сфери в Blender. Потім ми визначаємо функцію `fill_with_spheres`, яка виконує ітерацію по всіх вершинах у сітці та додає сфери в їхніх місцях за допомогою функції `create_sphere`. Нарешті, ми визначаємо функцію `change_internal_structure_to_spheres`, яка обчислює розмір кожної сфери на основі об'єму сітки та бажаної щільності сфери, а потім викликає `create_spheres_mesh`.

Функція `create_spheres_mesh` використовує вихідну сітку та щільність сфери, а потім викликає `generate_spheres` для створення списку вершин сфери. Потім він створює новий сітчастий об'єкт і додає до нього ці вершини разом із новими гранями, згенерованими на основі близькості сфер. Це генерує сітку зі структурою, яка нагадує оригінальну сітку, але з пористою внутрішньою структурою, створеною доданими сферами.

У функції `create_spheres_mesh` ми спочатку генеруємо вершини сфери за допомогою функції `generate_spheres`. Потім ми створюємо новий об'єкт сітки та встановлюємо його вершини на згенеровані вершини сфери. Далі ми створюємо список нових граней, перевіряючи близькість кожної пари

```
python Copy code

import bpy
import bmesh
from mathutils import Vector

# Create a new sphere object in Blender
def create_sphere(radius, location):
    bpy.ops.mesh.primitive_uv_sphere_add(size=radius, location=location)

# Iterate through all vertices in the mesh and add spheres at their location
def fill_with_spheres(mesh, sphere_radius):
    bm = bmesh.new()
    bm.from_mesh(mesh)

    for vert in bm.verts:
        create_sphere(sphere_radius, vert.co)

    bm.to_mesh(mesh)
    bm.free()

# Convert the internal structure of the mesh from a solid to a structure with
def change_internal_structure_to_spheres(mesh, sphere_radius):
    # Calculate the size of each sphere based on the mesh's volume and the density
    sphere_density = 0.1 # Number of spheres per cubic unit
    mesh_volume = mesh.volume
    sphere_volume = 4/3 * 3.14 * sphere_radius ** 3
    sphere_size = ((sphere_volume * sphere_density) / mesh_volume) ** (1/3)

    # Fill the mesh with spheres
    fill_with_spheres(mesh, sphere_size)
```

Рисунок 1 – Реалізація зміни геометрії

сфер і додаючи нову грань, якщо вони досить близько. Останнім кроком ми встановлюємо грані нової сітки в цей список і повертаємо його. За допомогою такого підходу ми можемо модифікувати внутрішню структуру 3D-моделей для широкого спектру застосувань, включаючи створення пористих структур або зміну внутрішньої структури біологічних моделей. Метод, який ми представили, є простим, але ефективним способом досягти цих змін, зберігаючи фізичні характеристики вихідної 3D-моделі. Цей метод можна реалізувати в різних мовах програмування, таких як C++ і Python, і можна адаптувати для різних випадків використання. Використовуючи програмні методи та алгоритми, ми можемо ефективно змінювати 3D-моделі та досягати бажаних результатів без шкоди для їхніх фізичних характеристик.

Список використаних джерел:

1. Badenhorst Wessel. Practical Python Design Patterns [Електронний ресурс] : Pythonic Solutions to Common Problems / Wessel Badenhorst ; by Wessel Badenhorst. – 1st ed. 2017. – Berkeley, CA: Apress, 2017. – XVII, 350 p. 1 illus. in color. – ISBN 9781484226803.