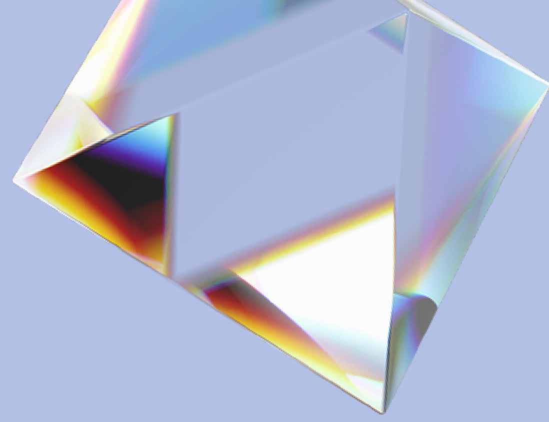
# UNITY 3D POINT CLOUD HAPTIC SIMULATION



- Based on density binning algorithm
- Achieve dual feedback effect
- Adapt to 40,000-particle scenes

Haotian.Zeng
2255702

## Real-Time Surface Feedback

**Project Background**

Adapting density binning from cited paper to enable lightweight 3D point cloud haptics without complex colliders.

**Core Features**

Dual feedback: hover highlights via hoverBrightness, pressure deformation via currentPressure along camera axis.

**Key Code Logic**

Double-buffered positions (original/current) in PointCloudGenerator with GenerateUniformSphere method.

**Performance Benefits**

Optimized for ≤40k particles, maintains 60fps with clean reusable code structure.

## Surface touch selection algorithm

Step 1 — Step 2 — Step 3

**Dual cache location storage**

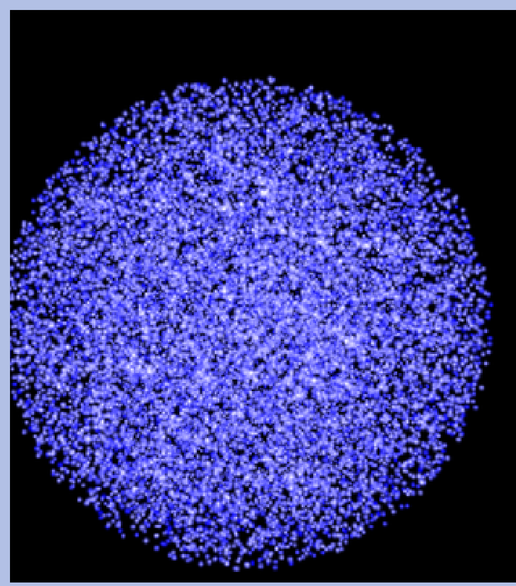
se "originalPositions" to store he initial coordinates, and currentPositions" to record he dynamically deformed ositions. This ensures that the

**Deep binning filtering**

The number of particles in each depth interval is counted using the int[] bins, and the firstSurfaceBin is used to locate the surface particles, thereby optimizing the efficiency of the

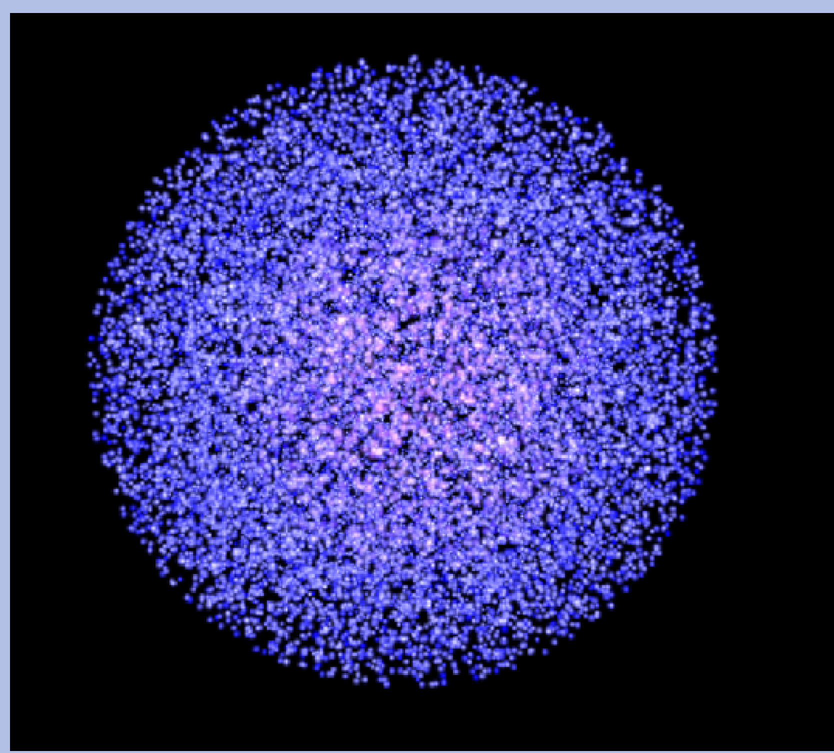
**Dynamic feedback logic**

hoverBrightness controls the hover highlight, targetDisp realizes the pressing deformation, and parameters drive the visualization of particle behavior.



## Generation of Uniform Spheres

The GenerateUniformSphere method generates a uniformly distributed point cloud sphere using the Fibonacci spiral algorithm.



### Hover highlight logic

The particle brightness is controlled based on the hoverBrightness parameter, and the surface particle highlight intensity is automatically calculated when the mouse hovers over it.

### Dual-cache mechanism

Dual-cache mechanism

## 01

## Lightweight 3D Point Cloud Interaction

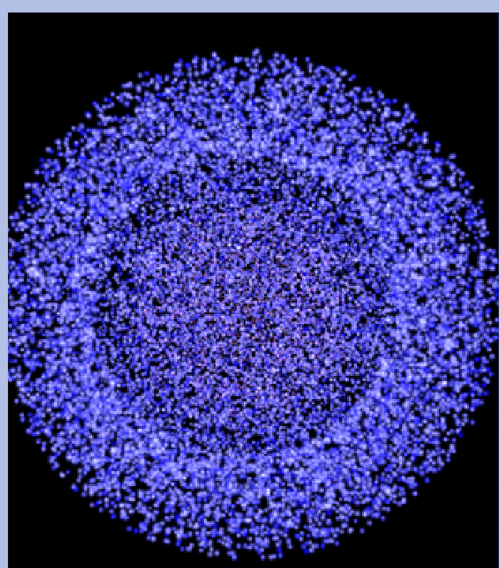
Implementing hover/press dual-feedback using density binning from referenced paper, enabling surface particle selection without complex colliders.

**1 Core Logic**

Adopting depth-based particle counting from the paper to identify surface layer without complex colliders.

**2 Optimization**

Implementing lightweight position storage (dual buffers) for efficient surface particle selection.



## Performance optimization

By conducting deep binning to filter out surface particles, the computational complexity is reduced, ensuring smooth operation of a 40,000-particle scene.

## Core code for tactile feedback

**01. Highlight effect realization**

The brightness of the particles is controlled by the "visualLight" variable, and the "hoverBrightness" parameter is used to adjust the intensity of the highlight when the mouse hovers over.

**02. Deformation displacement calculation**

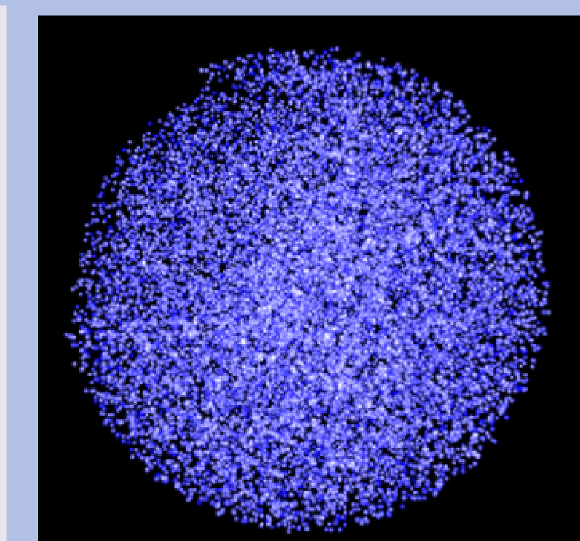
The particle displacement is calculated based on the targetDisp vector, and the currentPressure parameter controls the degree of deformation during pressing.

**03. Dual-cache mechanism**

The "originalPositions" store the initial positions, while "currentPositions" record the real-time positions. This ensures that the deformation is reversible.

**04. Surface layer screening**

The array "bins" is used to count the number of deep particles, and "firstSurfaceBin" determines the outermost particle as the interaction surface.



## Principle of Stress Deformation

Based on the "currentPressure" parameter, the particles are controlled to move in the direction of the camera, achieving the physical deformation feedback effect during pressing.

## Future Work

In the future, we will deeply draw on the more sophisticated density binning and structure-aware algorithms proposed in the paper "Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations" to iterate and optimize the existing Unity 3D point cloud haptic simulation solution. On one hand, we will introduce the hierarchical point cloud density detection logic from the paper to replace the current simplified depth binning scheme, which will enhance the accuracy and efficiency of surface touch detection for large-scale point clouds (over 100,000 particles), reduce invalid computations of particles in non-target areas, and thus optimize overall performance. On the other hand, we will expand the system's scalability based on the multi-dimensional interaction framework in the paper, adding support for multi-touch and custom touch shapes (e.g., irregular Lasso selection), and interface with haptic feedback hardware to achieve a complete closed-loop from visual feedback to physical haptic feedback, adapting to more diverse 3D point cloud interaction scenarios.