Hardware-accelerated Rendering of Web-based 3D Scatter Plots with Projected Density Fields and Embedded Controls

Wagner, Limberger, Scheibel, Döllner
Hasso Plattner Institute, University of Potsdam

27th International Conference on 3D Web Technology
Web3D 2022
Introduction

3D Scatter Plot Advantages

Comprehensible Mapping of data points to 3D points is easily graspable, leveraging inherent human 3D understanding

Versatile Allow arbitrary mapping of data to visual variables

Performant Hardware-accelerated rendering combined with instancing is fast
Occlusion

Main Issue: Occlusion

- Data points may be occluded by other data points based on view
- May even be surrounded on all sides, becoming inaccessible
- Well-known issue in 3D rendering
Occlusion

Trivial Solution: Hide Occluding Geometry

- Cutting planes allow filtering of visible data points
- Have to be placed manually
A View Inside: Density Maps

- X-ray-like 2D view
- Projected onto grid planes to provide reference
Projected Density Maps

References Between 3D and 2D Plots

References lines from point to grids facilitate orientation

References lines placed perpendicular on grids allow analyzing clusters
Projected Density Maps

Demo
Projected Density Maps

Building on Scatter Plot Matrices

Graphic by RIDC NeuroMat, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60578568

Hardware-accelerated Rendering of Web-based 3D Scatter Plots with Projected Density Fields and Embedded Controls

Wagner, Limberger, Scheibel, Döllner
Overplotting

2D Projection Introduces Overplotting

Overplotting: Data points overlap, becoming indistinguishable

Solution: Visualize density

Density Maps

Rendering the Density Maps

- Project points to 2D
- Calculate distance to point center per output bin
  - Choose adequate distance function (e.g. linear)
- Add up distance per bin to calculate density
Density Maps

Storing the Density Maps

- Count using the stencil buffer
  - Only 8 bit precision in WebGL 2\(^1\)
- Add up using integer textures
  - No blending of integer textures (not even additive) in WebGL 2\(^2\)
- Add up using floating point textures
  - No filtering of 32-bit floating point textures in WebGL 2\(^3\)
  - Greatly reduced precision due to 16-bit textures’ 10-bit mantissa

\(^1\)OpenGL ES 3 specification, table 3.14
\(^2\)OpenGL ES 3 specification, chapter 4.1.7
\(^3\)OpenGL ES 3 specification, table 3.13
Density Maps

**On Density Map Precision**

- IEEE 754 half floats (16 bit) store 1 sign bit, 5 exponent bits and 10 mantissa bits
- 11 effective bits of precision (due to implicit leading 1)
- Only $2^{11} = 2048$ distinct values
- Example: Assuming normal distribution, 0.04 % of points affect the center bin
- Target precision of 0.1 allows for 500 thousand points in plot

---

4 Resolution 512 px, 5 px point size. Using $\mu = 255$, $\sigma = 100$, approximately 2 % of points fall into $[253, 258]$, affecting the center pixel. Extending this to 2D results in 0.04 % of points. See onlinestatbook.com/2/calculators/normal_dist.html
Density Maps

Improving Density Map Precision

• Use 32-bit floating point textures, allowing for 4 billion points to be handled.
• Distribute points onto multiple buffers, which are combined during an additional post-processing pass.

23 bit mantissa, 24 bit precision, \(2^{24} = 16,777,216\) distinct values. Assuming max. 0.04 % of points affecting a single bin and target precision of 0.1, \(2^{24} \times 0.1/0.02^2 = 4,194,304,000\) points can be handled.
Density Maps

Storing the Density Maps

- Stored in RGB texture
- One density map per channel
Projected Density Maps

Assembling the Scene

Data Points

Clusters

Reference lines

Grids with density maps

Labels

Cutting plane handles

15  Hardware-accelerated Rendering of Web-based 3D Scatter Plots with Projected Density Fields and Embedded Controls  Wagner, Limberger, Scheibel, Döllner
Projected Density Maps

Performace Impact of This Technique

- Decreased rendering resolution compared to main rendering pass (e.g., $256 \times 256$ instead of $1920 \times 1080$, factor $31.6$)
- Simpler shaders compared to main rendering pass
- Density maps can be cached as long as data remains unchanged
- In total: no significant impact on rendering performance
Hardware-accelerated Rendering of Web-based 3D Scatter Plots with Projected Density Fields and Embedded Controls

Wagner, Limberger, Scheibel, Döllner

Contact: lukas.wagner@hpi.de

Hasso Plattner Institute, University of Potsdam

Try it out: demo.varg.dev/vidi/web3d-2022/
Source Code: github.com/lukaswagner/vidi