Advanced Computer Graphics: H01 - Gaussian splatting report

Advanced computer graphics 2023/24

Tjaša Mlakar, 63210505

1 Introduction

In this homework, we got familiar with the basic principles of Gaussian splatting. Gaussian splatting is a method for rendering volumetric data, especially point clouds, where each point is associated with a Gaussian function.

2 The tasks

2.1 Basic transformations and point rendering

The first tasks after reading and parsing the input data was to implement basic transformations and point rendering. We constructed a view transformation matrix that transformed each of the splat centers into camera space and a perspective transformation matrix, that transformed each splat from camera space to clip space. After that, we projected the splats on the screen and colored the underlying pixels with the colors of the corresponding splats, as seen in Figure 1.

2.2 Perspective-correct scaling

When projecting the splats in Task 1 we ignored perspective projection. In Task 2 we drew splats that are nearer to the camera larger and those further away smaller. To achieve this we used a scaling parameter s that was manually inputted when running the code. The final scale is s/z, where z is the depth of the splat (z coordinate of the splat) in view space. The splats were then rendered as view-aligned squares with side length 2s/z pixels, as seen in Figures 2 and 3.

2.3 Order-correct blending

By this point, the order we were drawing the splats in was random. When multiple splat centers were projected onto the same pixel, the color of the splat that was drawn last overwrote the previous colors. To achieve correct rendering in Task 3, we sorted the splats according to the screen-space depth before rendering. We also incorporated the alpha component of the splat colors and used back-to-front alpha compositing to blend the colors of different splats together, as seen in Figures 4 and 5.

2.4 Gaussian falloff

The last task, Task 4, was to update the renderer by varying the alpha channel according to the distance from the center of the splat. The intensity of the color of the splat is highest in the center of the splat and it radially decreases the further away we are from the center. After we computed the Gaussian falloff value for each pixel in the splat ”environment”, we multiplyed the alpha channel with it before the last step, back-to-front alpha compositing. This can be seen in Figures 6 and 7.

3 Performance measurements

Presented are the performance measurements for the input file ”nike.splat” (see Table 1), varying the scaling parameter s. The bigger the scaling parameter s, the bigger the area of pixels that is covered by a single splat and therefore more time needed for code to perform the Gaussian falloff calculation and connected tasks.

Task	Param. s	Running time
Basic transformations	20	2.82 secs
Perspective-correct scaling	20	3.37 secs
Order-correct blending	20	4.91 secs
Gaussian falloff	20	143.70 secs
Basic transformations	50	2.84 secs
Perspective-correct scaling	50	3.59 secs
Order-correct blending	50	5.21 secs
Gaussian falloff	50	796.14 secs

Table 1: Performance measurements for input file ”nike.splat”, where number of splats is 270.491 and viewport size is 512 by 512 pixels.

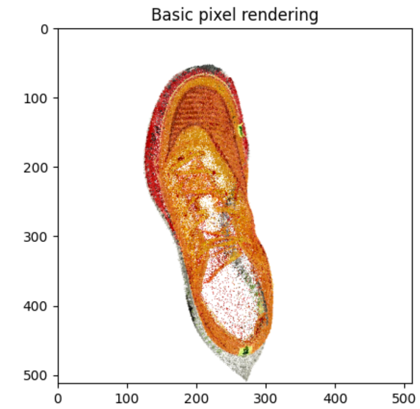

Figure 1: Basic transformations and point rendering result for input file ”nike.splat”.

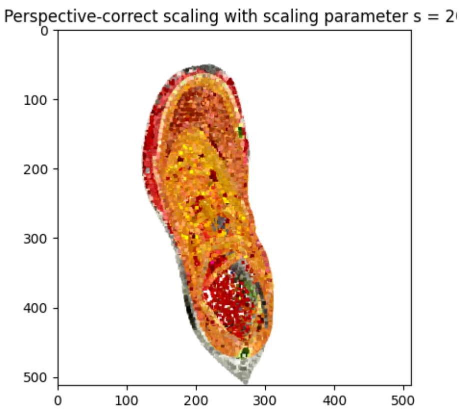

Figure 2: Perspective-correct scaling for input file ”nike.splat” with scaling parameter s=20.

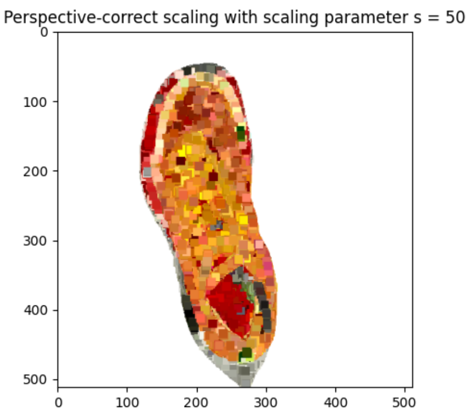

Figure 3: Perspective-correct scaling for input file ”nike.splat” with scaling parameter s=50.

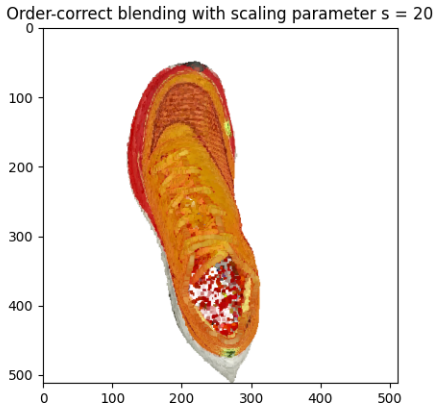

Figure 4: Order-correct blending for input file ”nike.splat” with scaling parameter s=20.

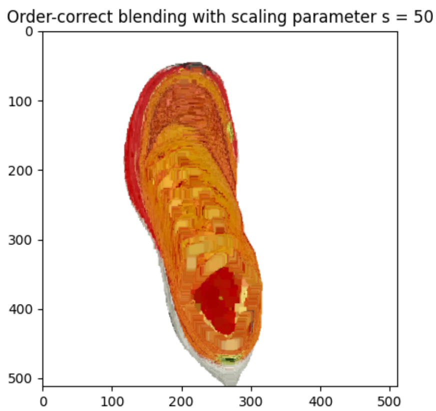

Figure 5: Order-correct blending for input file ”nike.splat” with scaling parameter s=50.

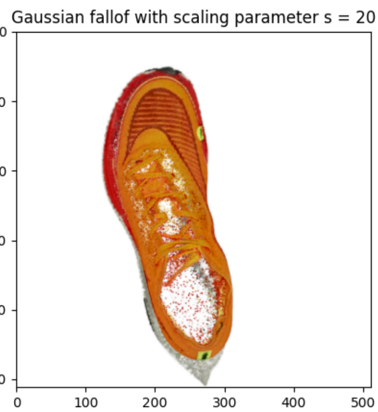

Figure 6: Gaussian falloff for input file ”nike.splat” with scaling parameter s=20.

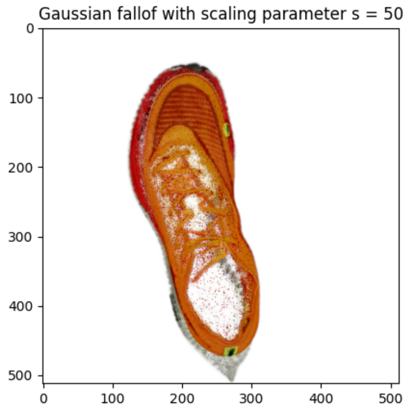

Figure 7: Gaussian falloff for input file ”nike.splat” with scaling parameter s=50.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Trans. Graph., 42(4), jul 2023.

[2] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. EWA volume splatting. In Proceedings of the Conference on Visualization ’01, VIS ’01, page 29–36, USA, 2001. IEEE Computer Society.