

Optimization-Free Style Transfer for 3D Gaussian Splats - Supplemental Materials

1. Point Filtering

In a practical sense, individual Gaussian splats tend to lie along implicit geometric surfaces, or at the very least tend to cluster rather than occupying isolated points in space. In most situations, any Gaussian splats that are highly isolated are likely to be “noisy” data points or artifacts left over from the training steps when the representation was first generated from image data. To reduce this noise, a filtering mechanism is implemented as preprocessing step to our graph construction pipeline. In addition to being visually unappealing, these noisy data potentially distort or even obscure content and reduce the accuracy of style transfers. With the assumption that most geometric objects tend to be contiguous and will be represented by splats that roughly follow along an implicit surface, we conclude that the more isolated a point is, and the more distant it is from any implicit surface, the more likely it is to be a noisy data point and a candidate for filtering out. Handily, both isolation from neighboring points and distance from an implicit surface can be calculated at the same time in our method.

The filtering method is as follows: the Gaussian splat is converted to a sparse point cloud comprised of the central points of each individual Gaussian splat, which we define as set P . For each $p \in P$, the KNN algorithm is used to determine the neighborhood \mathcal{N} about that point. In that neighborhood, the midpoint p_m is calculated as the mean of all points in the neighborhood. The distance from the neighborhood average is then simply $p - p_m$:

$$d = (p - p_m) \text{ where } p_m = \frac{1}{K} \sum_{j \in \mathcal{N}} p_j$$

Figure 1 provides a visual demonstration of this mathematical operation.

The calculated distance d for isolated points is much greater than the d of points in or on the implicit surface or for points that are in tightly clustered groups. Figure 2 visualizes this phenomenon with a toy example. The filtering process then simply filters out the points with the top distances according to a user defined percentile. Filtering those points reduces the noise in the 3D Gaussian splat and better represents the implicit surface.

This filtering showed a high degree of reliability in segregating between individual Gaussian splats centered on points on and along the surface of objects being represented and those with large offsets from that surface. Some well-trained 3D Gaussian splats from public repositories required no filtering. Other publicly available 3DGSs, and in particular user-generated Gaussian splats, do require filtering. User-generated splats suffered from low training times and low-resolution reference images, and thus contained a high percentage of artifacts and noise.

To study the efficacy of our filtering approach, we examined two different cases. The first was a test case constructed by generating a geometrically simple Gaussian splat containing noisy points. The second case involved a user-generated Gaussian splats.

For the first case, a programmatically-derived Gaussian splat made up of 5000 equal sized Gaussians distributed at equidistant points along the surface of a unit sphere was generated. This was seeded with an additional 500 Gaussian splats centered on points offset from the surface. Offsetting was accomplished by randomly dispersing the points within a 2x2x2 volume centered on the unit sphere. Our filtering approach was applied to remove points not along the surface of the sphere. This obtained a filtering accuracy of 98.87%. Results are shown in Figure 3.

In the second case, the Scaniverse mobile app was used to create a 3D Gaussian splat of a household object, in this case a chair. This file was found to be particularly noisy and so was an excellent candidate for this study. Before and after filtering results are presented in Figure 4. As is shown, isolated points are preferentially filtered, while points along the implicit surface of the object, in this case a chair, are not filtered. One limitation of this approach, however, is the inability to filter points that are noisy but tightly clustered.

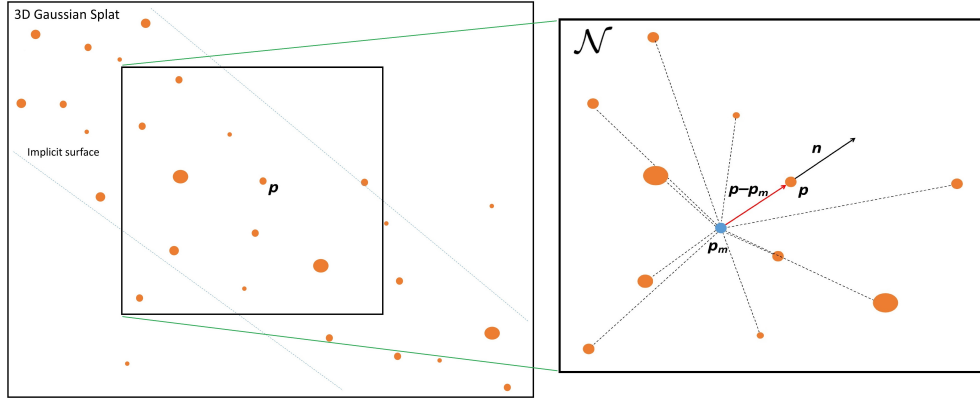


Figure 1. Within the point cloud derived from a 3D Gaussian splat, points in a neighborhood about some selected point p are used to determine the mean point p_m from which the distance d is derived. Insofar as p is on or within the implicit surface, d should be small. The value of d increases when p is at a greater distance from other points in its neighborhood, i.e when p does not lie in or on the implicit surface.

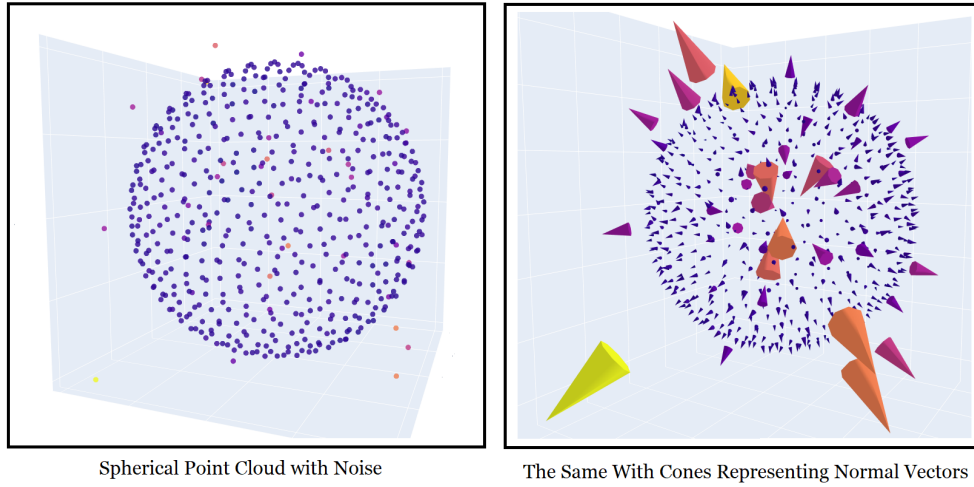


Figure 2. A spherical point cloud with a large number of points added at random distances from the surface to simulate noise (Left) and the same point cloud with cones representing the vectors between p_m and p (Right) demonstrating how the distance between p_m and p strongly segregates points close to the surface from those far from the surface.

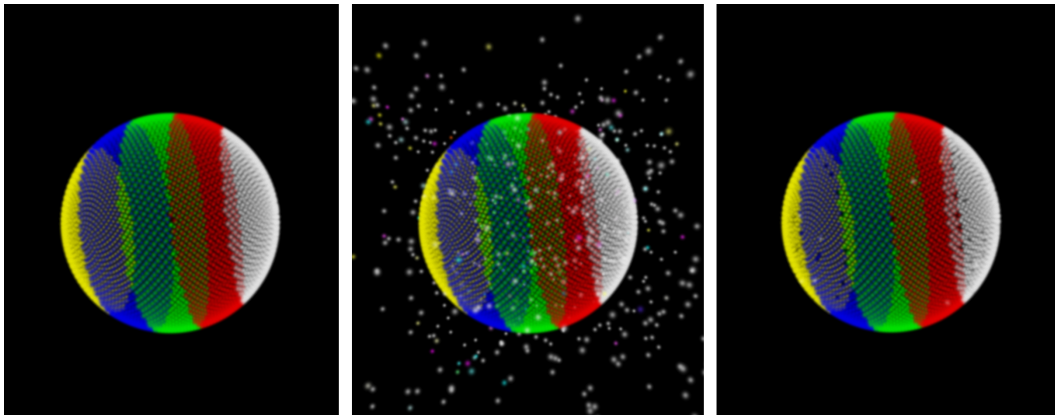


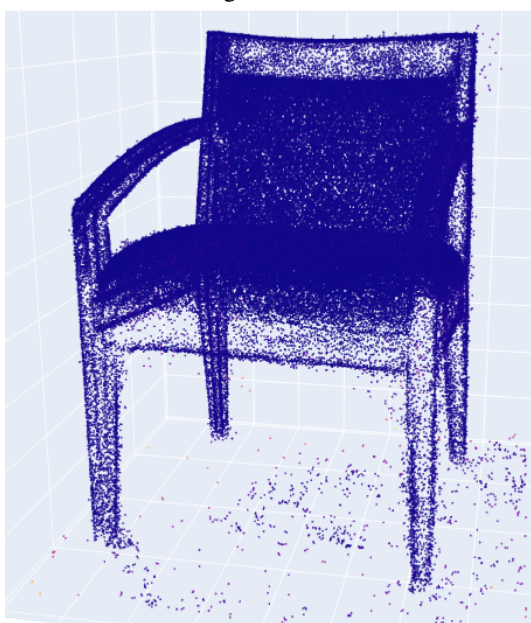
Figure 3. A 3D Gaussian of a sphere (Left) with a large number of points added at random distances from the surface to simulate noise (Middle) and the same after our filtering approach (Right)



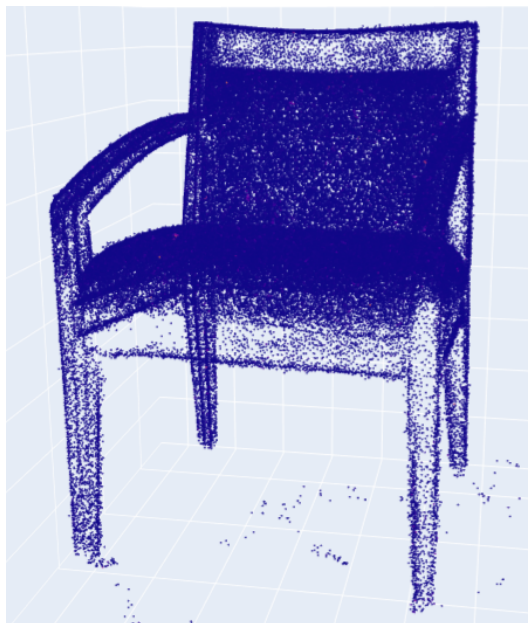
Original 3DGS



Filtered 3DGS



Original 3DGS as a Point Cloud



Filtered 3DGS as a Point Cloud

Figure 4. A 3D Gaussian splat that contains a large number of artifacts (Top) and noise both before (Left) and after (Right) being filtered. Simplified representations of the 3DGS formed by creating a point cloud with the central point of each individual Gaussian splat (Bottom) demonstrate how isolated points and points well-outside the implicit surface are filtered.