Temporal Hierarchical Gaussian Mixture Models for Real-Time Point Cloud Streaming

Roland Fischer
Tobias Gels
rfischer@cs.uni-bremen.de
s_rs5s7r@uni-bremen.de
University of Bremen
Bremen, Germany

Rene Weller
Gabriel Zachmann
weller@cs.uni-bremen.de
zach@cs.uni-bremen.de
University of Bremen
Bremen, Germany

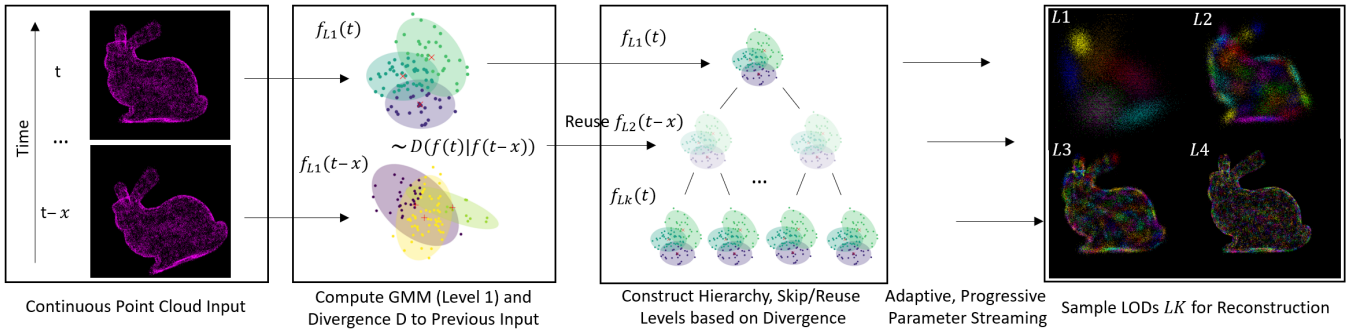

Figure 1: Pipeline of our approach: Cont. input (left), temporal GMM hierarchies (center), progressive LODs (right).

CCS CONCEPTS

• **Computing methodologies** → **Shape modeling**; **Massively parallel algorithms**.

KEYWORDS

Point Cloud, Streaming, Gaussian Mixture Model, Generative Model

ACM Reference Format:
Roland Fischer, Tobias Gels, Rene Weller, and Gabriel Zachmann. 2024. Temporal Hierarchical Gaussian Mixture Models for Real-Time Point Cloud Streaming. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH Posters '24), July 27 - August 01, 2024.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641234.3671086

1 INTRODUCTION

Point clouds play an important role in robotics, autonomous driving, and telepresence applications [Yu et al. 2021] with typical tasks such as SLAM and scene/avatar reconstruction. However, noisy sensor data, huge data loads, and inhomogeneous densities make efficient processing and accurate representation challenging, especially for real-time and streaming-based applications. Spatial data structures can speed up the processing and reduce the size, i.e., octrees are commonly used for compression. Voxelization and occupancy-based methods are popular for point cloud representation but tend to suffer from discretization artifacts and high memory consumption. Generative probabilistic models avoid these issues by providing a continuous parametric representation of the data. These models can model complex distributions while efficiently handling uncertainty in the data. Gaussian Mixture Models (GMMs) have been shown to allow for compact representations as well as high reconstruction fidelity and have been used for tasks such as registration, compression, and incremental mapping [Goel and Tabib 2023]. To reduce the high computational cost and allow for levels of detail (LODs), hierarchical forms and parallel computation can be employed [Eckart et al. 2016].

We propose a novel GMM-based approach, specifically designed for point cloud streaming. Our model consists of a hierarchy of GMMs and a top-down level-wise construction methodology, which enables a compact footprint as well as dynamic and progressive transmission and rendering of LODs. A key feature is that we achieve real-time speed with high-fidelity reconstructions by exploiting temporal coherence between consecutive input and a highly parallelized and optimized CUDA implementation. Our work can be easily extended to consider color and use Gaussian splatting.

2 OUR APPROACH

In our approach, we use a hierarchy of GMMs to represent the point cloud as a number of overlapping probabilistic mixtures, specifically, 3D anisotropic Gaussians. If the color should be encoded, too, the Gaussians' dimensionality increases, i.e., 4D when encoding the hue value. Starting from the top, each successive level subdivides the parent Gaussian mixtures into sets of smaller ones, further partitioning the data, thus, increasing the overall fidelity. See Fig. 1

SIGGRAPH Posters '24, July 27 - August 01, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0516-8/24/07
https://doi.org/10.1145/3641234.3671086

for on overview. We employ an octree and, to reduce the eventual size, dynamically stop the subdivision of a GMM when it contains too few points. We compute the hierarchical GMM by employing and solving the Expectation Maximization (EM) algorithm for each GMM in the hierarchy, resulting in a overall complexity of $O(N \log M)$ with N points and M mixtures. By propagating the mixing weights down the hierarchy, we ensure a valid global GMM. We allow the mixtures to overlap each other and, when subdividing, allow points to be part of multiple mixtures by weighting their contribution to the parent mixtures and eventual normalization. This should only occur rarely on the borders and thus not degrade the efficiency.

The work closest to ours is [Eckart et al. 2016], which also constructs a GMM hierarchy for point cloud processing. In contrast to them, we construct the hierarchy level by level rather than recursively, which enables us to progressively stream and render the computed LOD while the next finer one is being computed, see Fig. 1 (right). The GMM hierarchy also allows us to do bandwidth- and computing power-adaptive transmission and visualization. In the case of sequential point cloud data, e.g., live-captured sensor data, consecutive point clouds will often be highly similar. Therefore, we propose to exploit the temporal coherency to maximize efficiency and performance. Specifically, we utilize the level-wise model construction and compare the computed first level with the corresponding one from the previous input. Depending on the calculated similarity, we skip the computation of a proportional number of levels and reuse the respective ones from the previous input, see Fig. 1 (center). Subsequently, only the most detailed levels are computed by reusing the parent partitioning of the previous input and recomputing the Gaussians with the current points. This works only with structured point clouds, though. For unstructured point clouds, an alternative would be to transform the detail-level mixtures based on the differences in the upper levels. As a similarity metric between GMMs, we employ the variational approximation of the Kullback Leibler divergence by [Hershey and Olsen 2007], as it is more efficient than Monte Carlo sampling and still sufficiently accurate. Additionally, we initialize the top-level GMM with the parameters from the previous input for faster convergence. To reconstruct the point cloud (after transmission), we sample the encoded distribution using ancestral sampling. To achieve even better visualization, the Gaussian splatting rendering technique can easily be applied, thanks to the 3D GMM representation.

To achieve real-time performance, we parallelized our method using the GPU. In particular, we take advantage of dynamic parallelism and compute each GMM itself in parallel as well as all GMMs at the same level. To do this, we implemented a custom synchronization mechanism. Our implementation is highly optimized, i.e. by using sum reductions and maximizing shared memory usage. Our source code will be available soon at https://cgvr.cs.uni-bremen.de/research/pointclouds.

3 RESULTS

To evaluate our method in terms of speed, accuracy, and compactness, we measured the construction time as well as the PSNR and size required to represent the model. All tests were performed on a RTX4090 GPU using a sequential test scene, the common Stanford Bunny (continuously being transformed) with 36 k points. Fig. 2 shows that our method is fast and accurate. Also, with our temporal approach, we save a lot of time at higher levels, without a noticeable loss in accuracy. Naturally, the balance between speed and accuracy can be tuned by adjusting the divergence thresholds, e.g. for an even faster computation. By exploiting temporal coherence, we achieve compression factors of 4 for level 4 and 26.7 for level 3, which are higher than the factors reported by [Eckart et al. 2016].

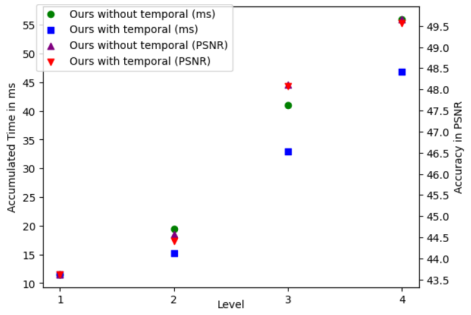

Figure 2: Results: Our approach, which exploits temporal coherency, significantly accelerates construction without compromising accuracy.

4 CONCLUSIONS AND FUTURE WORK

We presented a novel approach for compact point cloud representation and real-time streaming using a temporal hierarchical GMM-based generative model. Our level-based construction scheme allows us to dynamically adjust the maximum LOD and progressively transmit and render more detailed levels. We minimize the construction cost by exploiting the temporal coherence between consecutive frames. Combined with our highly parallelized and optimized CUDA implementation, we achieve real-time speeds with high-fidelity reconstructions. Our results show that we achieve significantly higher compression factors than previous work with similar accuracy, and that the temporal approach saves 20-36 % construction time in our test scene. In the future, we plan on integrating Gaussian splatting and generalizing our approach to unstructured point clouds.

REFERENCES

Ben Eckart, Kihwan Kim, Alejandro Troccoli, Alonzo Kelly, and Jan Kautz. 2016. Accelerated Generative Models for 3D Point Cloud Data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5497–5505.

Kshitij Goel and Wennie Tabib. 2023. Incremental Multimodal Surface Mapping via Self-Organizing Gaussian Mixture Models. *IEEE Robotics and Automation Letters* PP (12 2023), 1–8.

John R. Hershey and Peder A. Olsen. 2007. Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Vol. 4. IV–317–IV–320.

Kevin Yu, Gleb Gorbachev, Ulrich Eck, Frieder Pankratz, Nassir Navab, and Daniel Roth. 2021. Avatars for Teleconsultation: Effects of Avatar Embodiment Techniques on User Perception in 3D Asymmetric Telepresence. *IEEE Transactions on Visualization and Computer Graphics* PP (08 2021), 1–1.