Motivation

- A Gaussian Mixture Model (GMM) is a probabilistic model that is described by a weighted sum of normal distributions (Gaussians). GMMs can be fitted to 3D point clouds to describe their density.
- A novel deep learning method is being developed at TU Wien, which needs ways to construct and visualize GMMs.
- No tool for GMM visualization is available that matches our requirements.
- We want to quantify differences of existing GMM construction algorithms.

Contribution

- Development of a 3D-GMM visualization tool enabling both isoellipsoid and density visualization with different configurations.
- Description of the closed-form solution for the integral of a GMM along a ray, which was necessary for the visualization tool. We also correct a mistake in a previous related paper.
- Definition of suited metrics for evaluating a GMM’s quality.
- Implementations, adaptations, and quantitative comparison of several GMM-construction algorithms: EM, Top-Down HEM, and Geometrically regularized Bottom-Up HEM.

Construction

Evaluation Strategy:
- 200 Models from ModelNet40
- Each algorithm is tested in different configurations
- Metrics:
  - Reconstruction Error: Based on Chamfer Distance between point clouds from mesh and GMM
  - Irregularity: Variation of nearest-neighbor-distances from GMM point cloud to mesh point cloud

EM [1]
- E-step: Calculates point-Gaussian-responsibilities
- M-step: Updates GMM according to responsibilities
- Repeat steps until convergence
- Implementation in Python/PyTorch

Top-Down HEM [2]
- Fits a small GMM using EM
- Repeatedly replaces Gaussians with new sub-GMMs
- Implementation in Python/PyTorch

Geometrically reg. Bottom-Up HEM [3]
- Starts with many Gaussians
- Repeatedly merges several Gaussians into one
- Adaption of open-source C++ implementation

Visualization

Graphical User Interface

- Colored by weights
- Black to white color map, Regular
- White to black color map, Logarithmic

Isoellipsoids
One Ellipsoid per Gaussian

Density
Accumulated density along viewing ray through pixel

References: