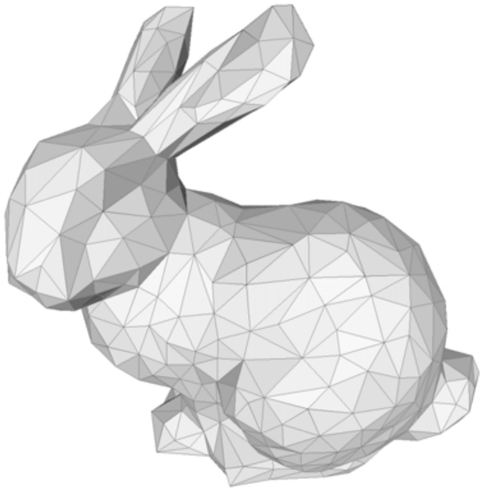
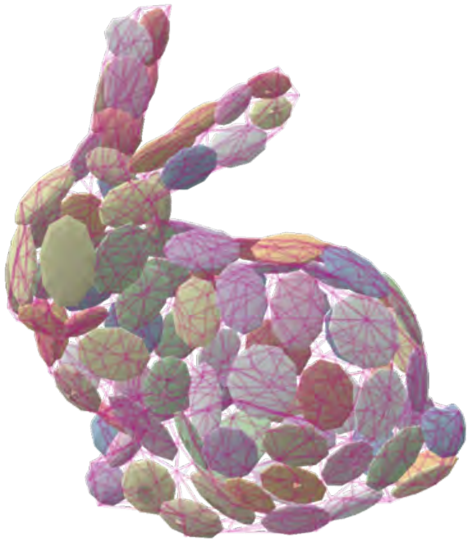
Direct Fitting of Gaussian Mixture Models



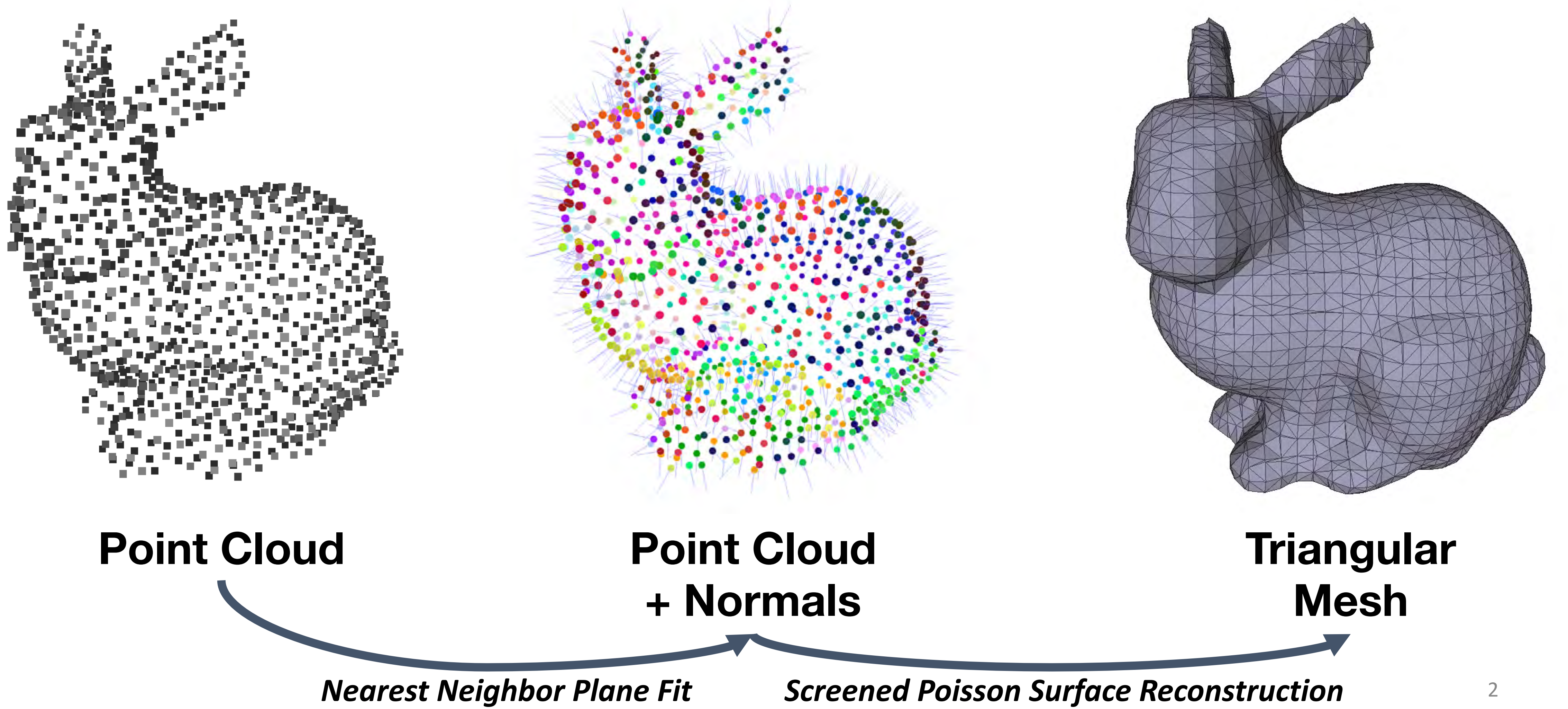
Leonid Keselman, Martial Hebert

Robotics Institute
Carnegie Mellon University
May 29, 2019



https://github.com/leonidk/direct_gmm

Representations of 3D data

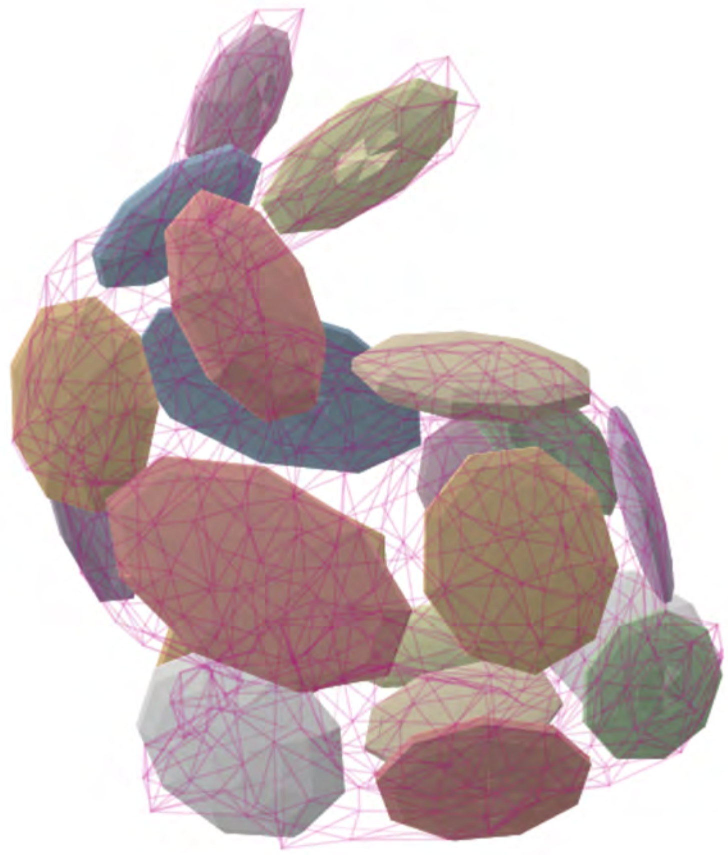


Gaussian Mixture Models for 3D Shapes

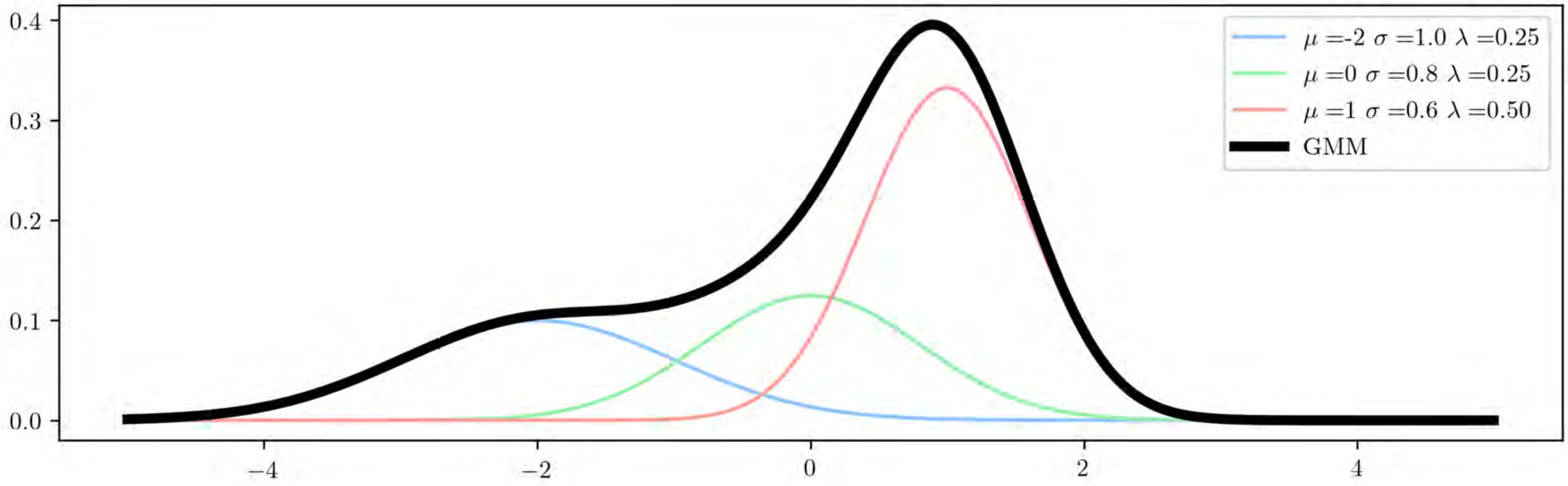
GMM fit to object surface

Benefits

- Closed-form expression
- Can represent contiguous surfaces
- Easy to build from noisy data
- Sparse



Gaussian Mixture Model (GMM)



$$p(x) = \sum_{n=1}^{K} \lambda_i \, \mathcal{N}(x; \mu_i, \Sigma_i)$$

$$\sum_i \lambda_i = 1$$

$$\lambda_i \geq 0$$

Σ_i is symmetric, positive-semidefinite

Gaussian Mixtures as a shape representation

Efficient Representation

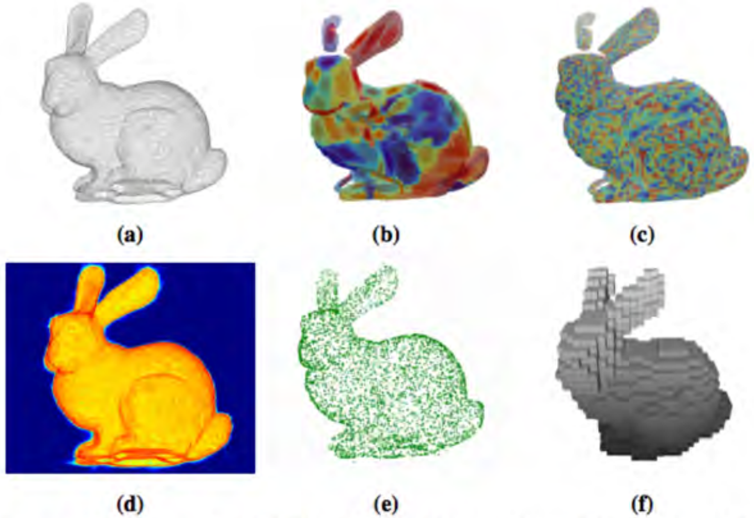
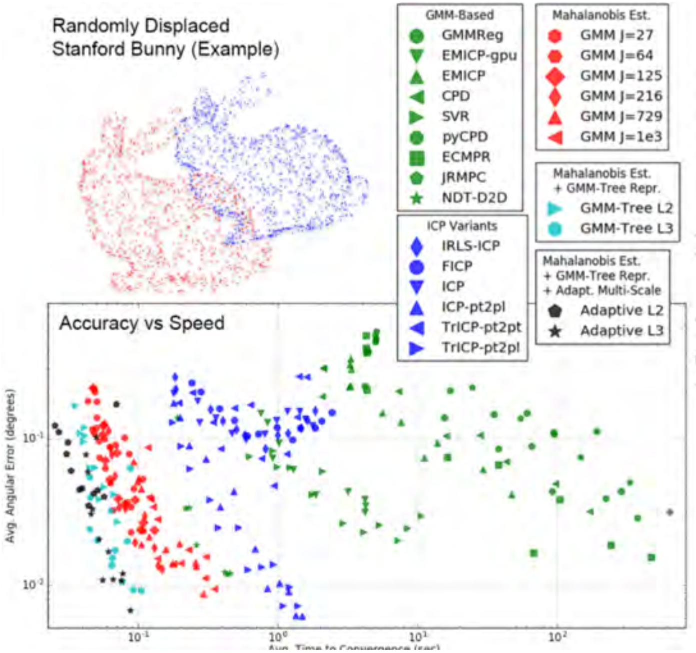


Figure 1. **Processing PCD with a Hierarchy of Gaussian Mixtures:** (a) Raw PCD from Stanford Bunny (*35k vertices*), (b) and (c) Two levels of detail extracted from the proposed model. Each color denotes the area of support of a single Gaussian and the ellipsoids indicate their one σ extent. Finer grained color patches therefore indicate higher statistical fidelity but larger model size, (d) a log-scale heat-map of a PDF from a high fidelity model. (e) stochastically re-sampled PCD from the model (*5k points*), (f) occupancy grid map also derived directly from the model.

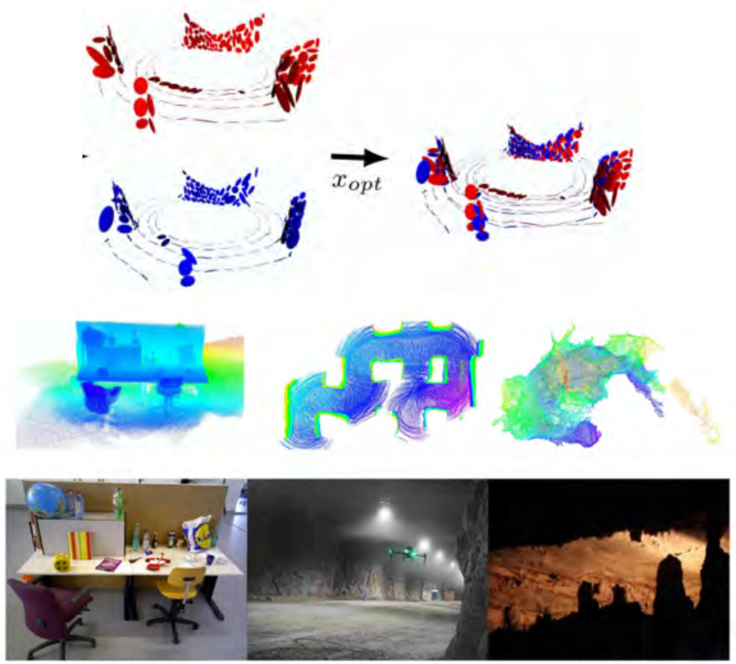
B. Eckart, K. Kim, A. Troccoli, A. Kelly, J. Kautz.
CVPR (2016)

Mesh Registration



B. Eckart, K. Kim, J. Kautz.
ECCV (2018)

Frame Registration

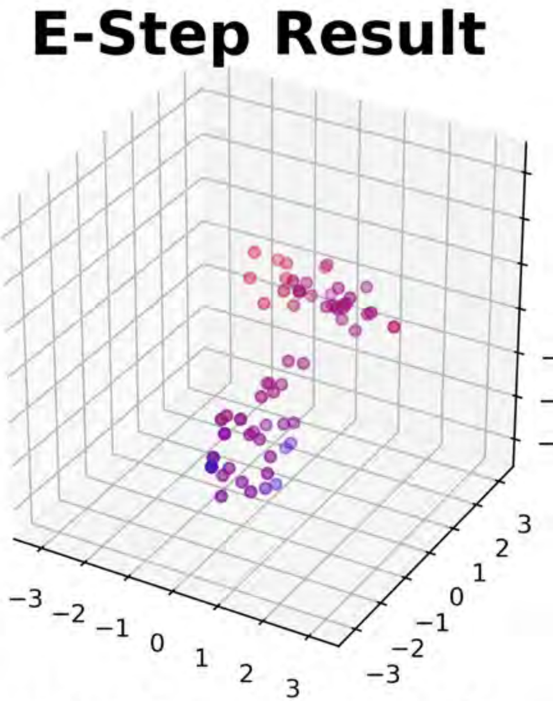


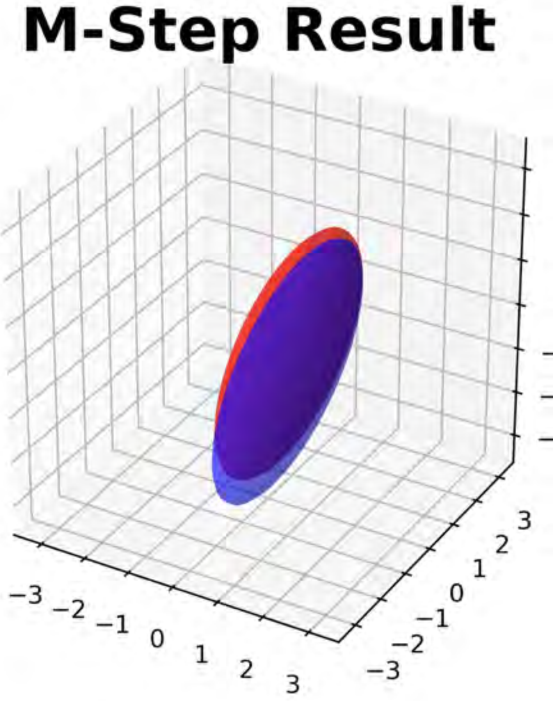
(a) TUM (b) Mine (c) Cave

W. Tabib, C. O'Meadhra, N. Michael
IEEE *R-AL* (2018)

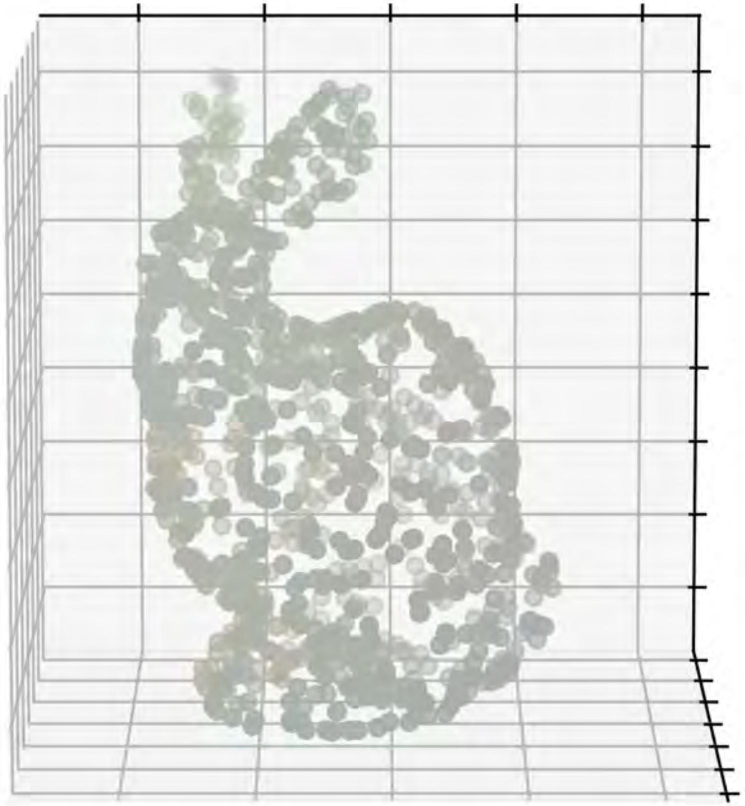
Fitting a Gaussian Mixture Model

1. Obtain 3D Point Cloud
2. Select Initial Parameters
3. Iterate Expectation & Maximization
   i. E-Step: Each point gets a likelihood
   ii. M-Step: Each mixture gets parameters

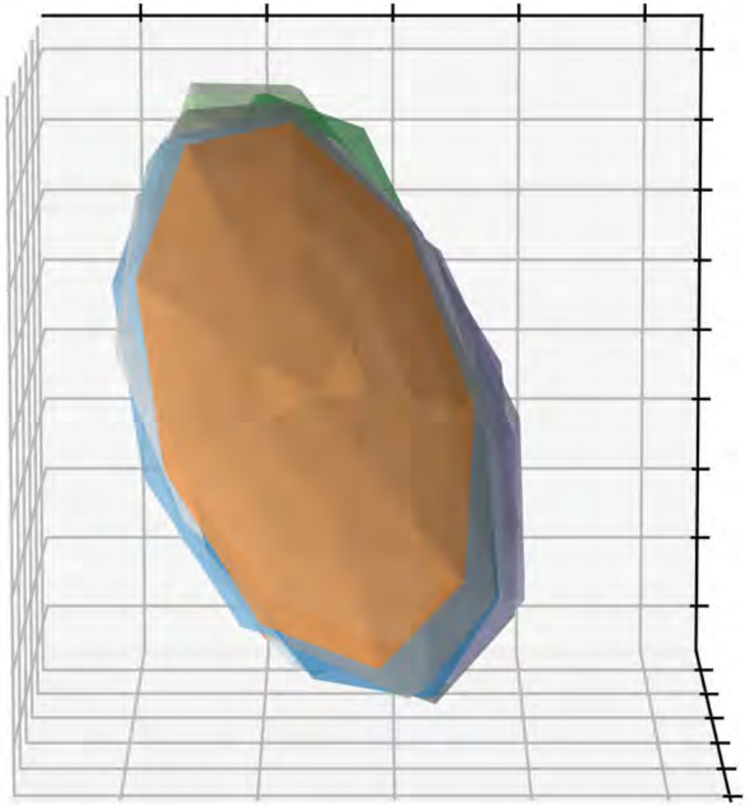




E-Step Result



M-Step Result



The E-Step (Given GMM parameters)

$$\eta_{ij} = \frac{1}{C_j} \lambda_i \, \mathcal{N}(x_j; \mu_i, \Sigma_i)$$

Affiliation between point j & mixture i

$$C_j = \sum_k \lambda_k \, \mathcal{N}(x_j; \mu_k, \Sigma_k)$$

Normalization constant for point j

The M-Step (Given point-mixture weights)

points **mixtures**

$$LB = \sum_{j=1}^{M} \sum_{i=1}^{K} \eta_{ij} \log\left(\lambda_i \, \mathcal{N}(x_j; \mu_i, \Sigma_i)\right)$$

lower-bound loss

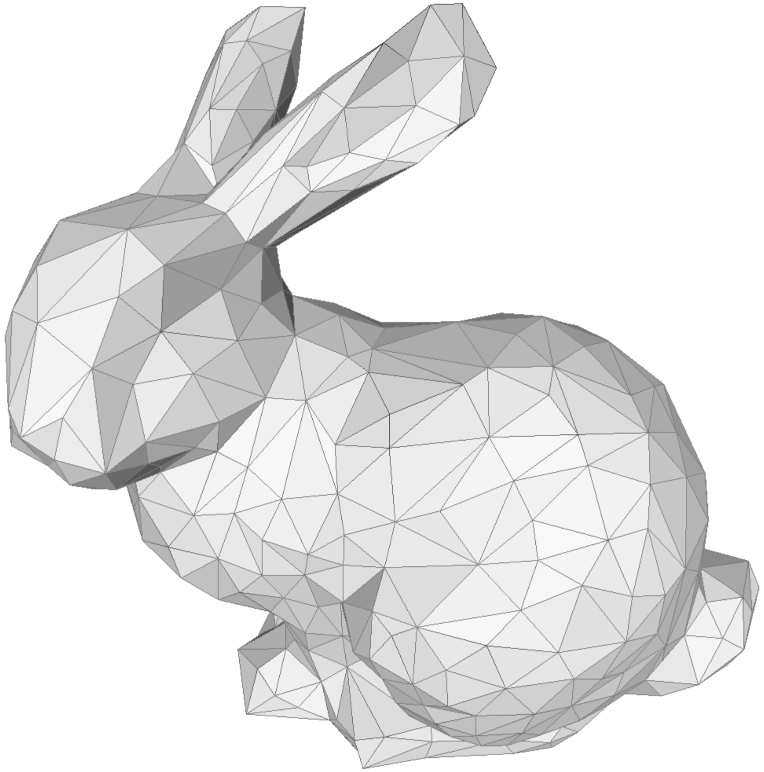
To get new parameters: takes derivatives, set equal to zero, and solve

$$w_{ij} = \eta_{ij}$$

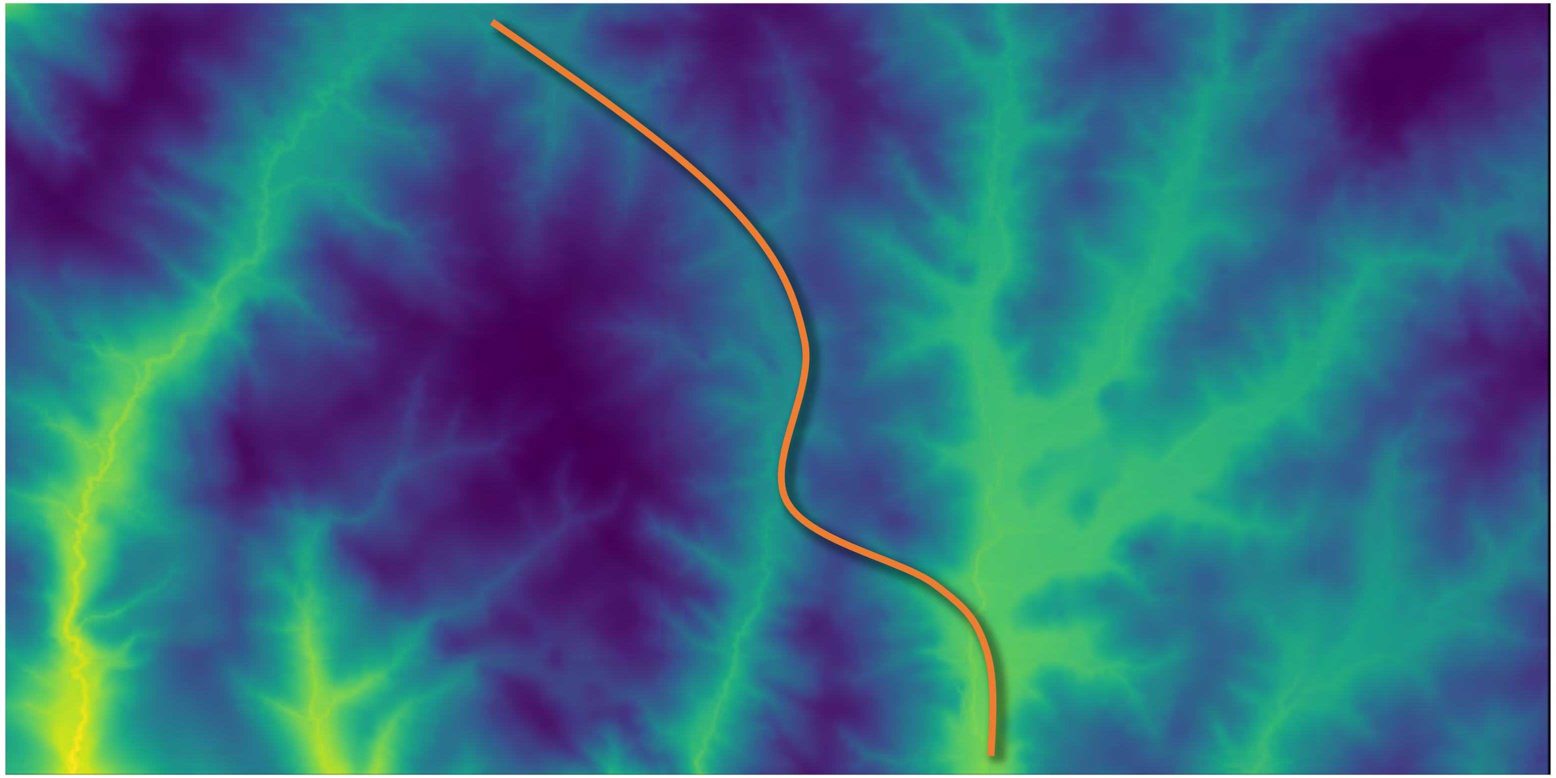
$$\frac{\partial LB}{\partial \lambda_i} = 0 \qquad \frac{\partial LB}{\partial \mu_i} = 0 \qquad \frac{\partial LB}{\partial \Sigma_i} = 0$$

$$W_i = \sum_j w_{ij}$$

$$\lambda_i = \frac{W_i}{M} \qquad \mu_i = \frac{1}{W_i} \sum_j w_{ij} x_j \qquad \Sigma_i = \frac{1}{W_i} \sum_j w_{ij} (x_j - \mu_i)(x_j - \mu_i)^T$$



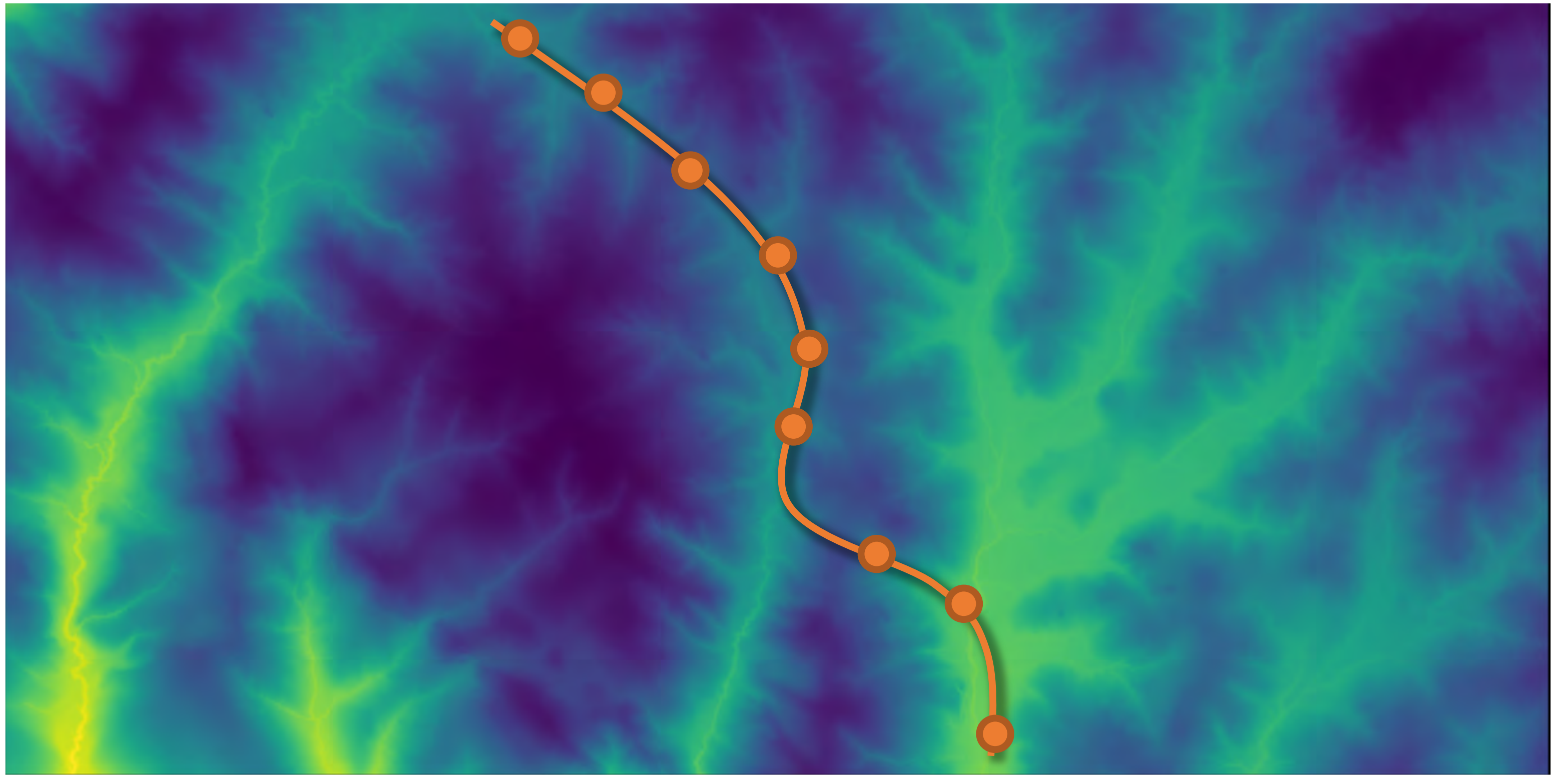
Geometric Objects in a Probability Distribution



Known curve in a given 2D probability distribution

Geometric Objects in a Probability Distribution

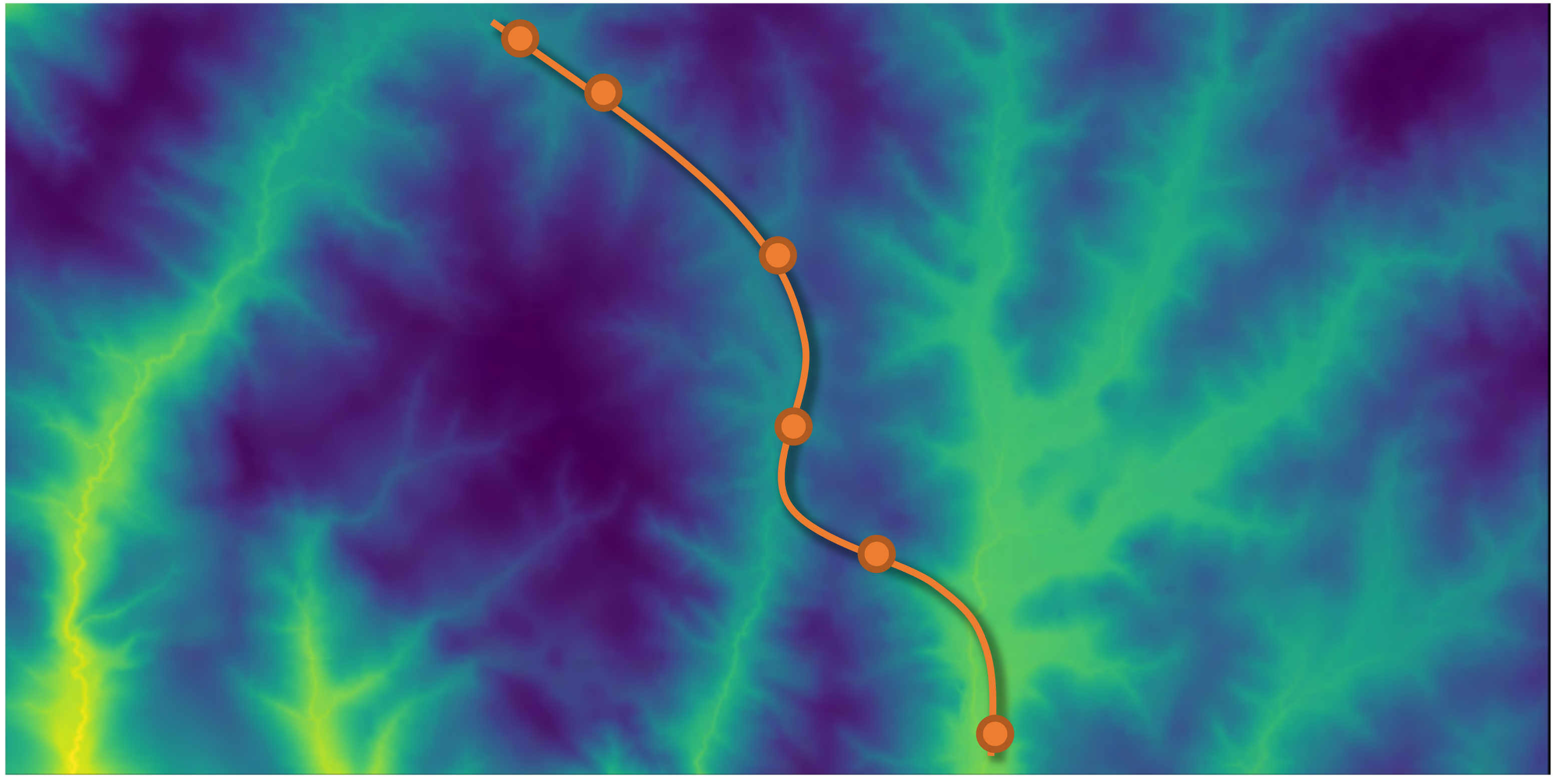
$$\ell(\text{curve}) \cong \prod_{i=1}^{N} p(x_i)$$



Consider sampling N points from this curve

Geometric Objects in a Probability Distribution

$$\ell\ (\text{curve}) \cong \left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}$$



Take a geometric mean to account for sample number

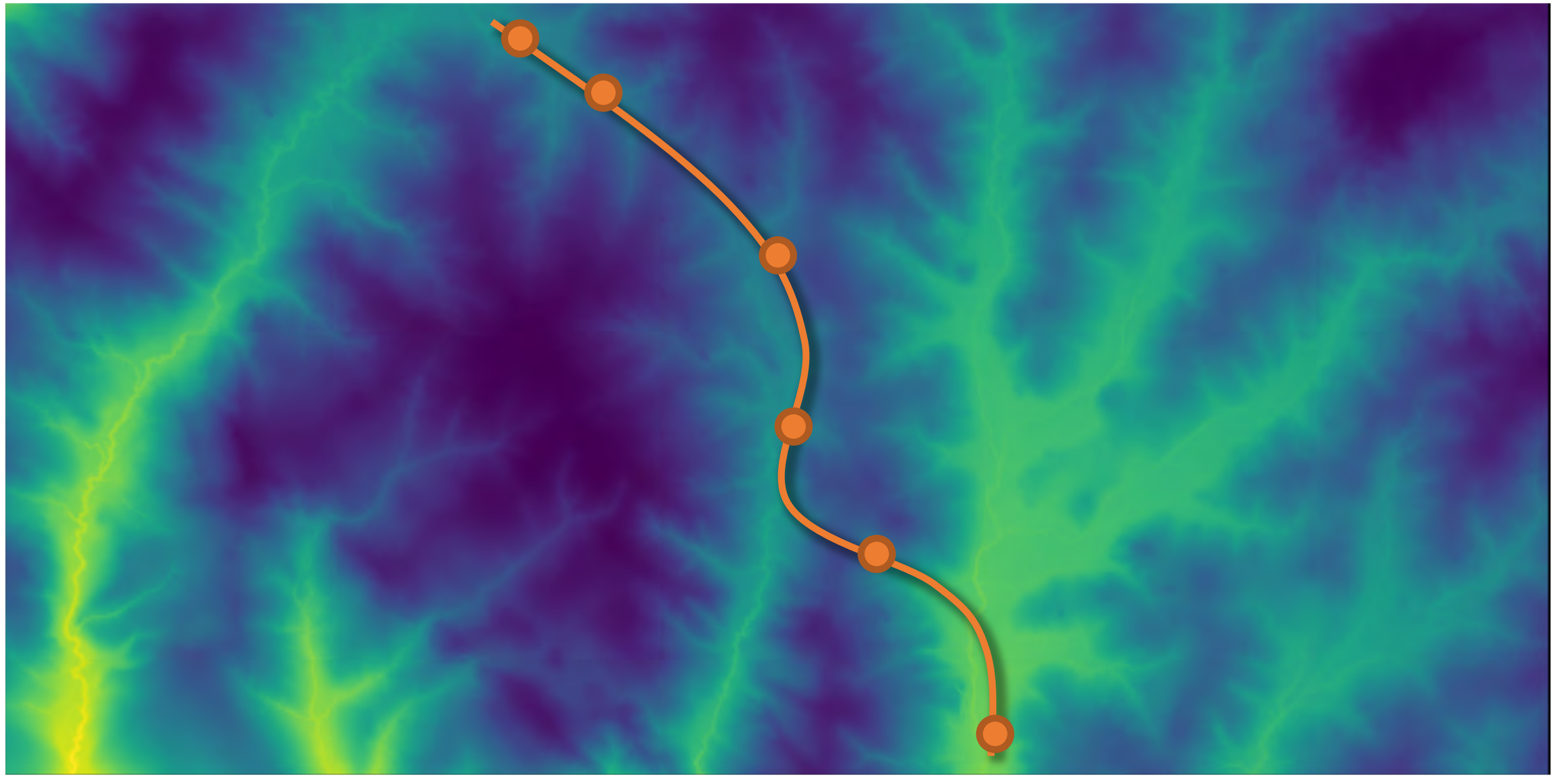
Geometric Objects in a Probability Distribution

$$\ell(\text{curve}) \cong \left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}$$

$$\ell(\text{curve}) = \lim_{N\to\infty} \left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}$$

$$= \lim_{N\to\infty} \exp\left(\log\left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}\right)$$

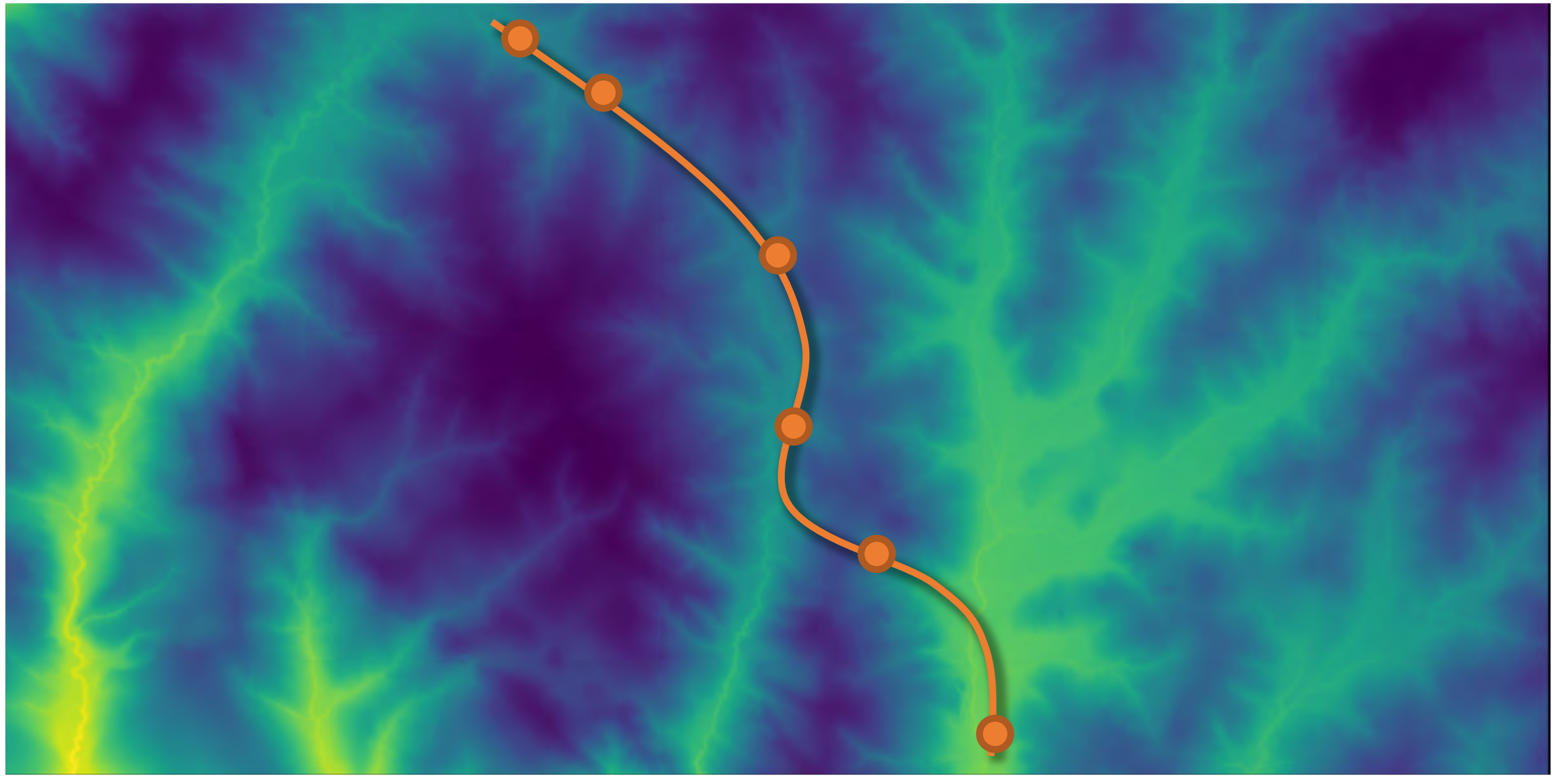
$$= \lim_{N\to\infty} \exp\left(\frac{1}{N} \sum_{i=1}^{N} \log(p(x_i))\right)$$



The curve will be the value in the limit

Geometric Objects in a Probability Distribution

$$\ell(\text{curve}) \cong \left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}$$



$$\ell(\text{curve}) = \lim_{N\to\infty} \left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}$$

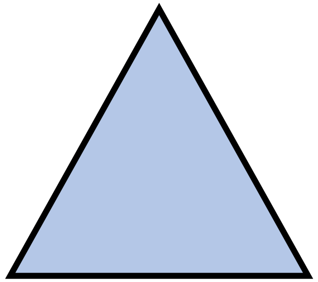
$$= \lim_{N\to\infty} \exp\left(\log\left(\prod_{i=1}^{N} p(x_i)\right)^{\frac{1}{N}}\right)$$

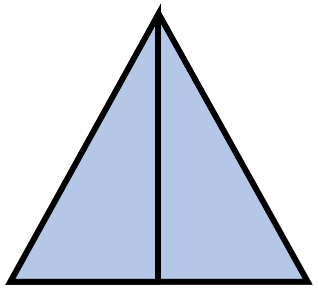
$$= \lim_{N\to\infty} \exp\left(\frac{1}{N} \sum_{i=1}^{N} \log(p(x_i))\right) = \exp\left(\int \log(p(x))\, dx\right)$$

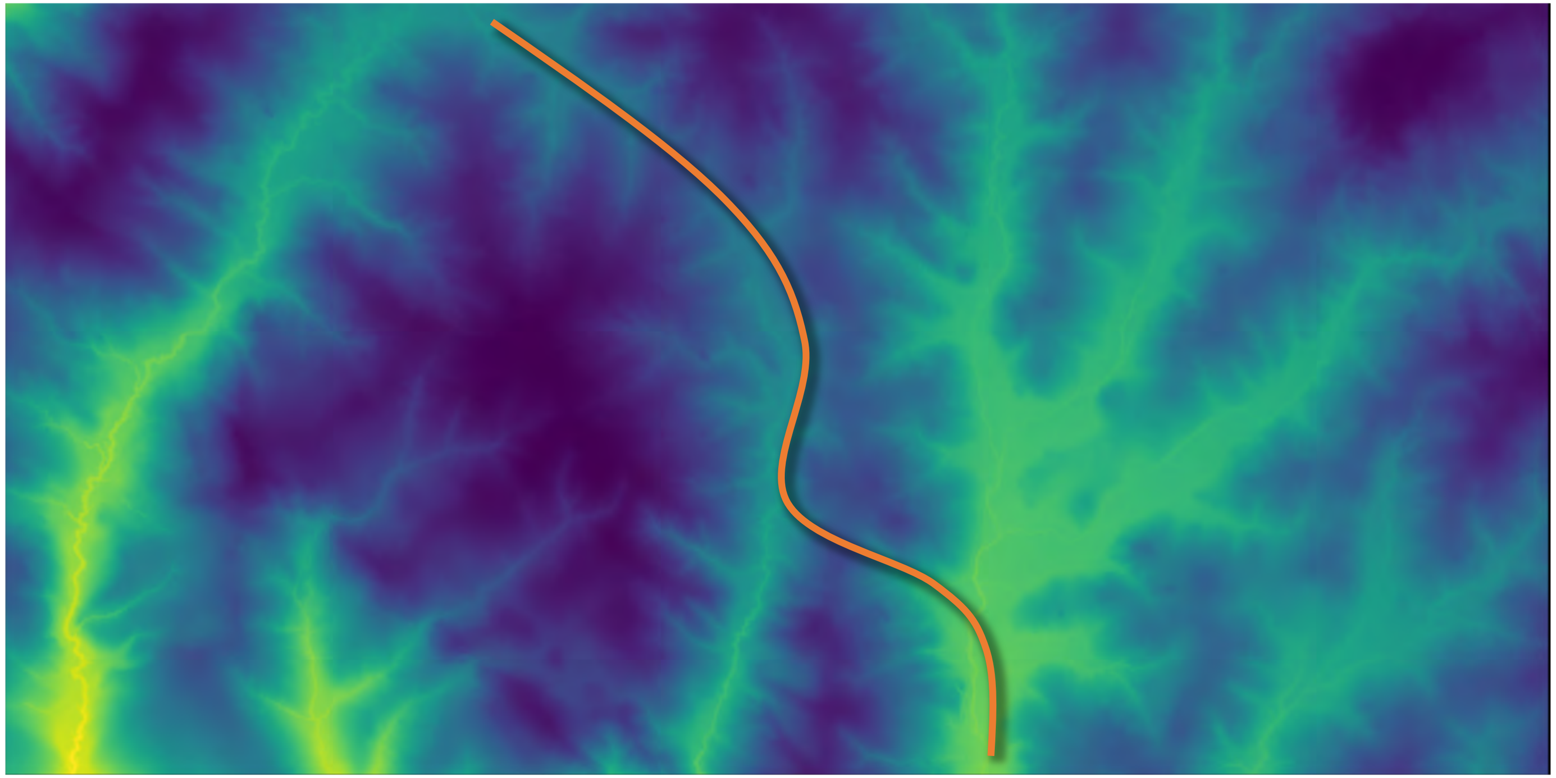
Geometric Objects in a Probability Distribution

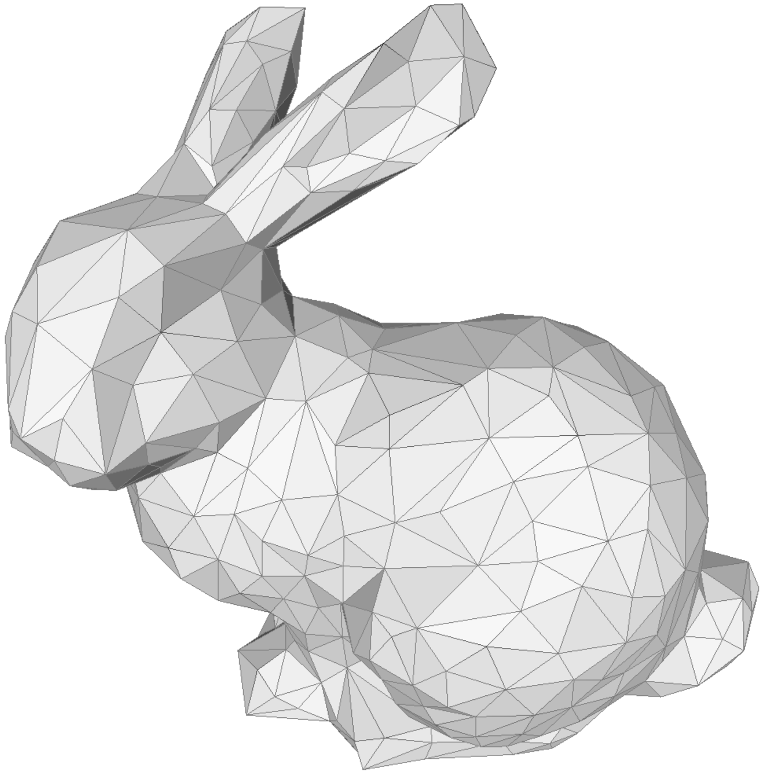
$$L = \exp\left(\int \log(p(x))\, dx\right)$$

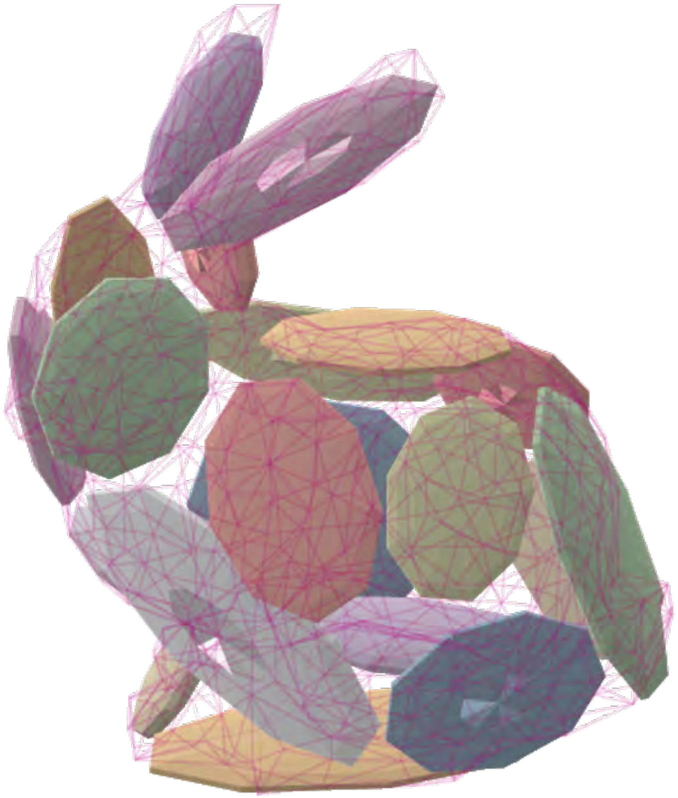
1. If $p(x) = 0$ on curve, then L$= 0$
2. Invariant to reparameterization











α_j Area of each triangle
μ_j Centroid of each triangle
A_j, B_j, C_j Triangle vertices

The E-Step (Given GMM parameters)

$$\eta_{ij} = \frac{1}{C_j} \lambda_i \mathcal{N}(x_j; \mu_i, \Sigma_i)$$

Affiliation between point j & mixture i

$$C_j = \sum_k \lambda_k \mathcal{N}(x_j; \mu_k, \Sigma_k)$$

Normalization constant for point j

α_j Area of each triangle
μ_j Centroid of each triangle
A_j, B_j, C_j Triangle vertices

The New E-Step (Given GMM parameters)

$$\eta_{ij} = \frac{1}{C_j} \lambda_i \alpha_j \mathcal{N}(\mu_j; \mu_i, \Sigma_i)$$

Taylor Approximation (2 terms)

Affiliation between object j & mixture i

$$C_j = \sum_k \lambda_k \alpha_k \mathcal{N}(\mu_j; \mu_k, \Sigma_k)$$

Normalization constant for object j

α_j Area of each triangle
μ_j Centroid of each triangle
A_j, B_j, C_j Triangle vertices

The M-Step (Given point-mixture weights)

$$\lambda_i = \frac{W_i}{M}$$

$$\mu_i = \frac{1}{W_i} \sum_j w_{ij} x_j$$

$$\Sigma_i = \frac{1}{W_i} \sum_j w_{ij} (x_j - \mu_i)(x_j - \mu_i)^T$$

$$w_{ij} = \eta_{ij}$$

$$W_i = \sum_j w_{ij}$$

The New M-Step (Given point-mixture weights)

$$\lambda_i = \frac{W_i}{M}$$

$$w_{ij} = \alpha_j \eta_{ij}$$

$$W_i = \sum_j w_{ij}$$

$$\mu_i = \frac{1}{W_i} \sum_j w_{ij} x_j$$

$$\Sigma_i = \frac{1}{W_i} \sum_j w_{ij} \left[(x_j - \mu_i)(x_j - \mu_i)^T + \Sigma_j \right]$$

α_j Area of each triangle

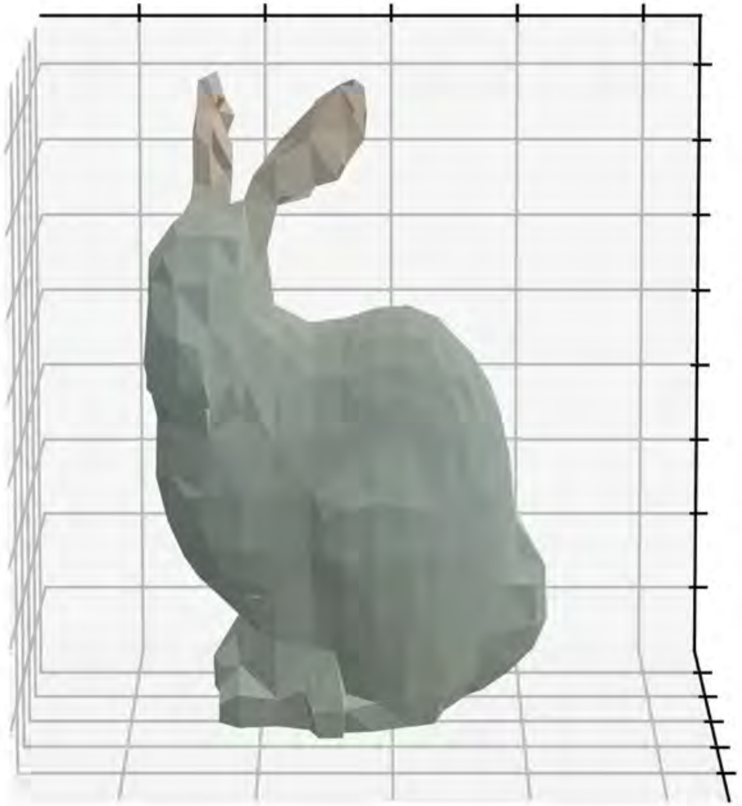
μ_j Centroid of each triangle

A_j, B_j, C_j Triangle vertices

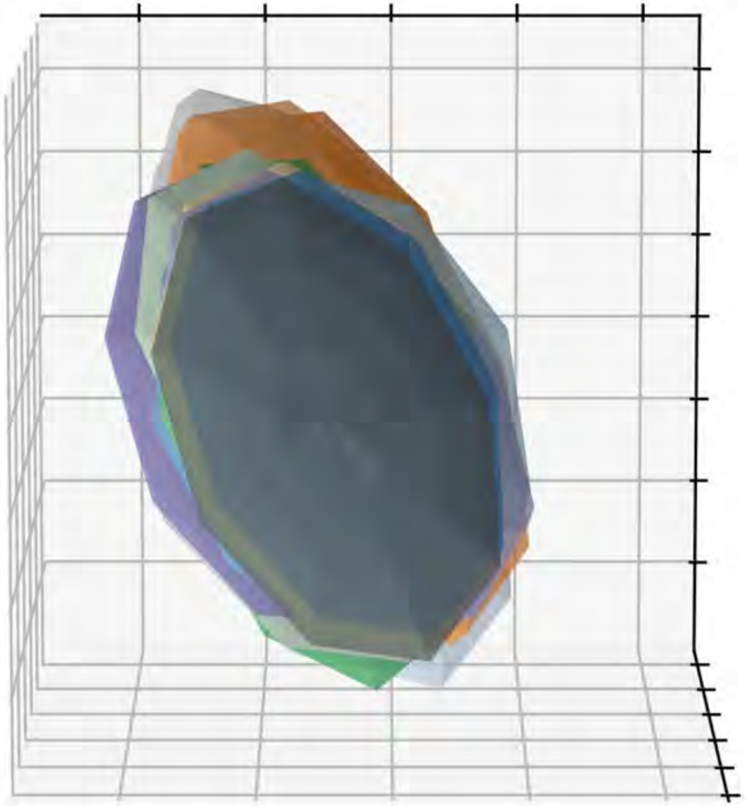
$$\Sigma_j = \frac{1}{12} \left(A_j A_j^T + B_j B_j^T + C_j C_j^T - 3\, \mu_j \mu_j^T \right)$$

What is Σ_j ?

E-Step Result

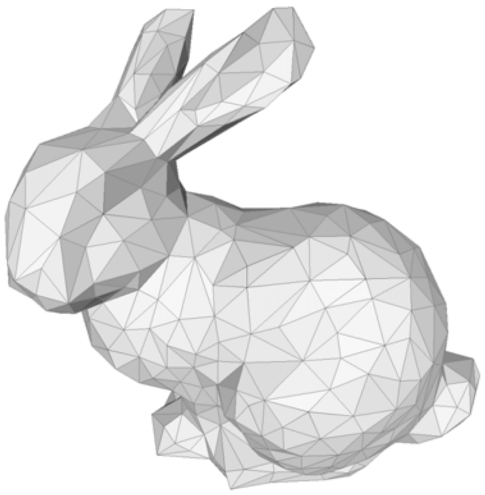


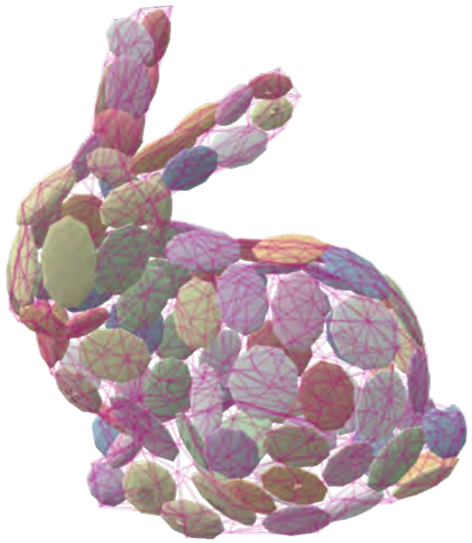
M-Step Result



Results

Did all that math actually help us fit better/faster GMMs?



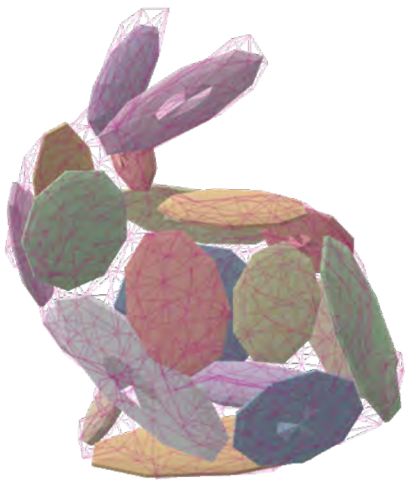


Using different **inputs**

classic algorithm
- Vertices of the mesh
- Triangle centroids

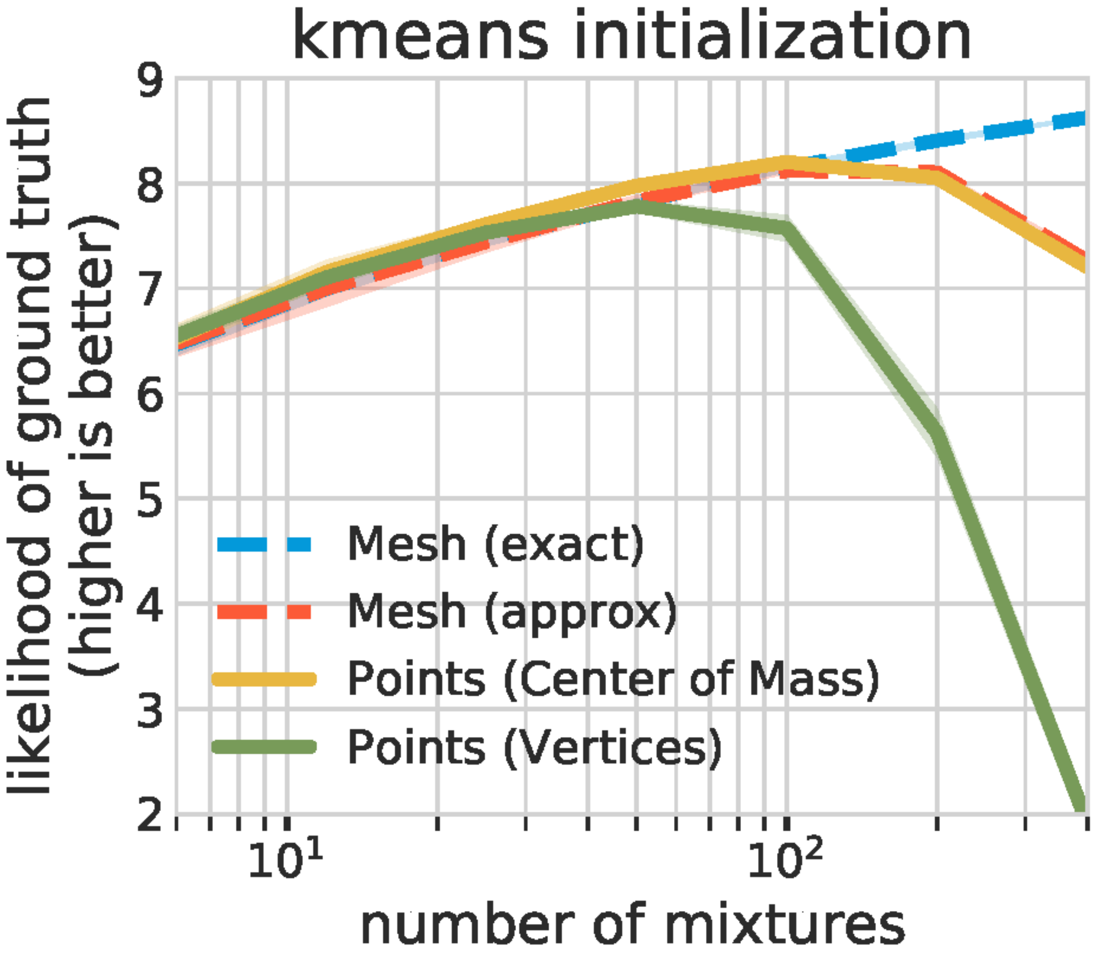
our method
- Approximate (E only)
- Exact (E + M steps)

Evaluate across a wide range of mixtures (6 to 300)

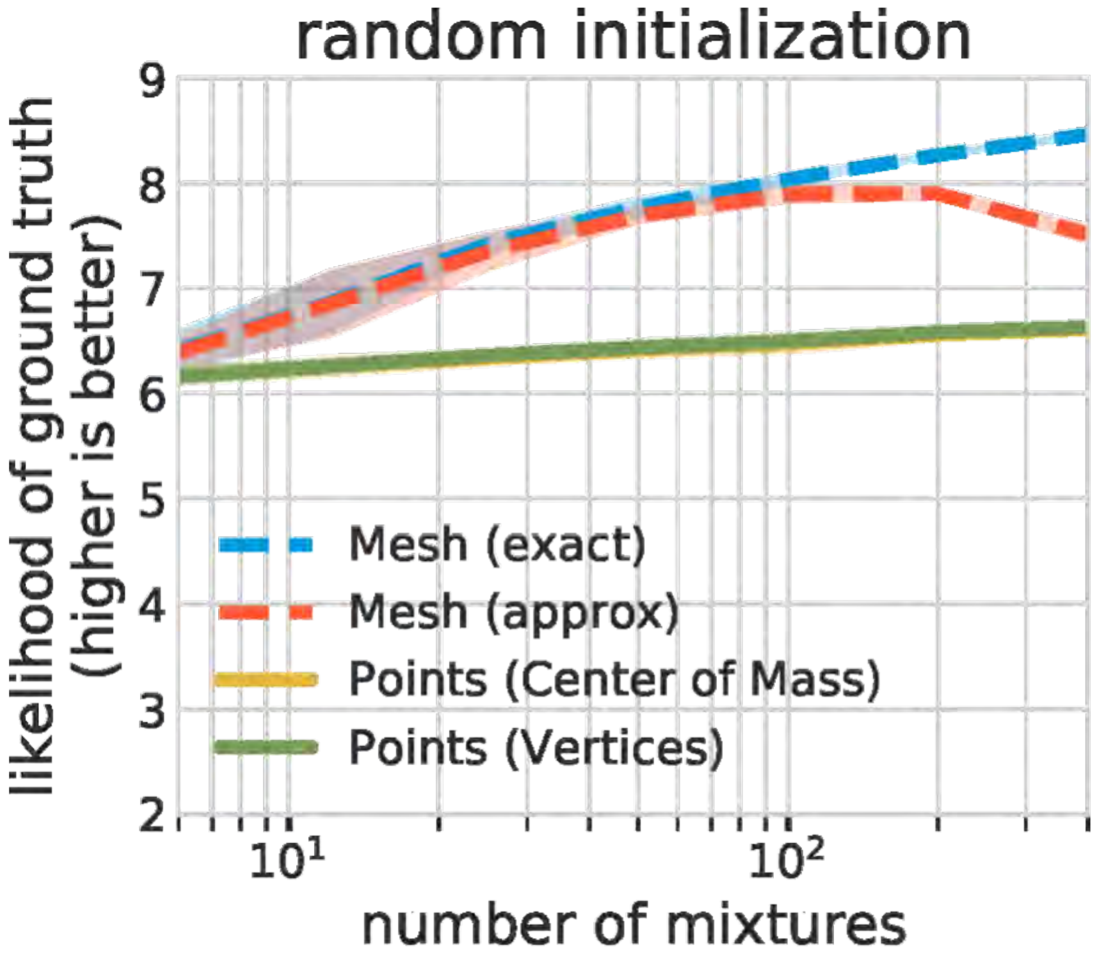


Measure the likelihood of a high-density point cloud (higher is better)

Full E+M method works in all cases



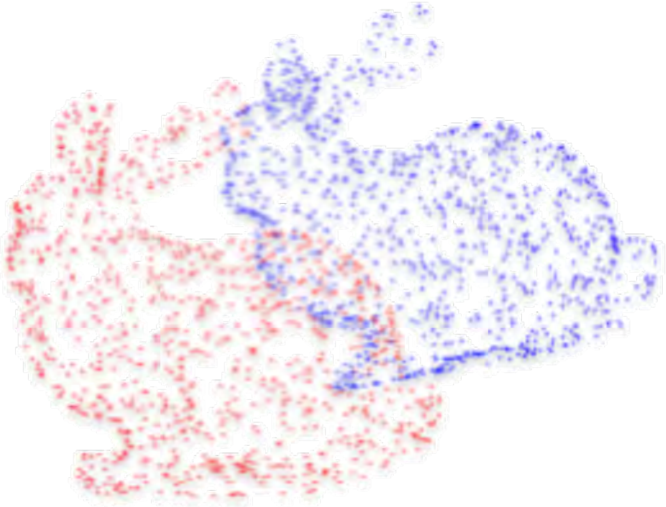
Stable under even random initialization!



Applications

Are these models actually more useful?

Mesh Registration (P2D)

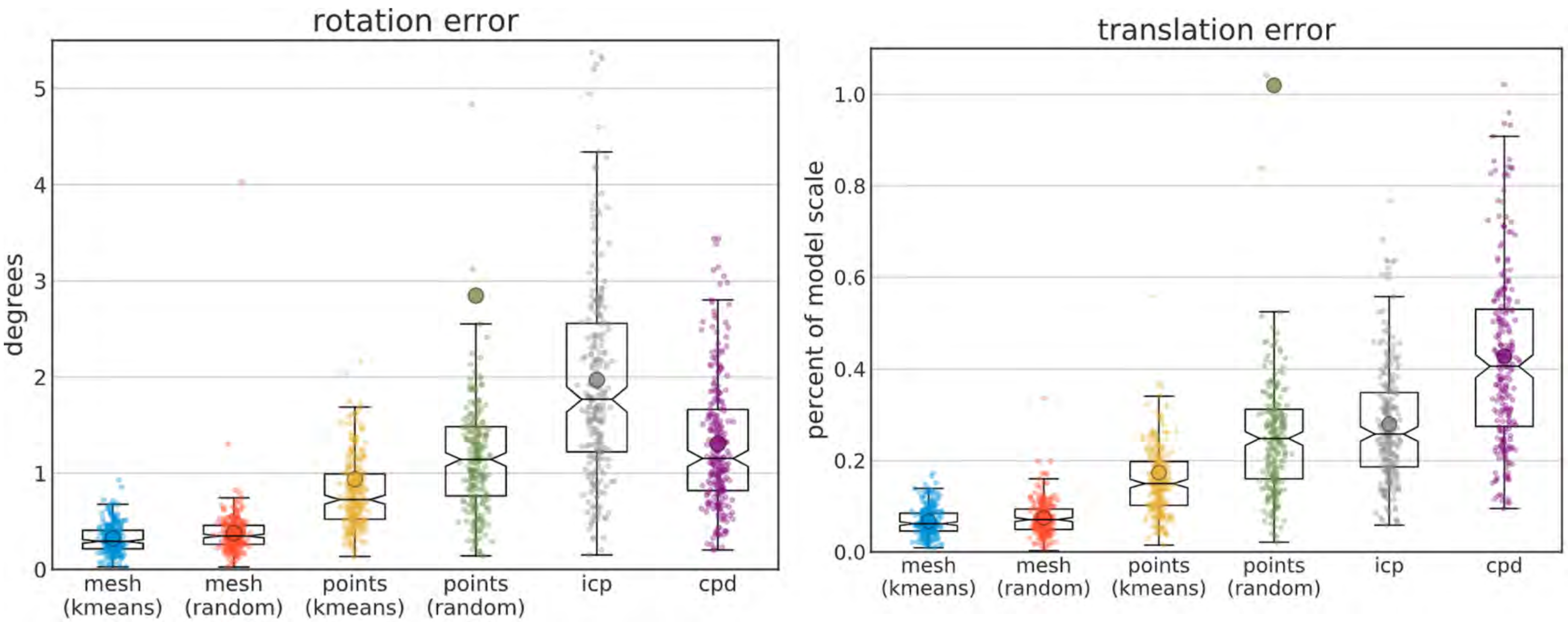


Method

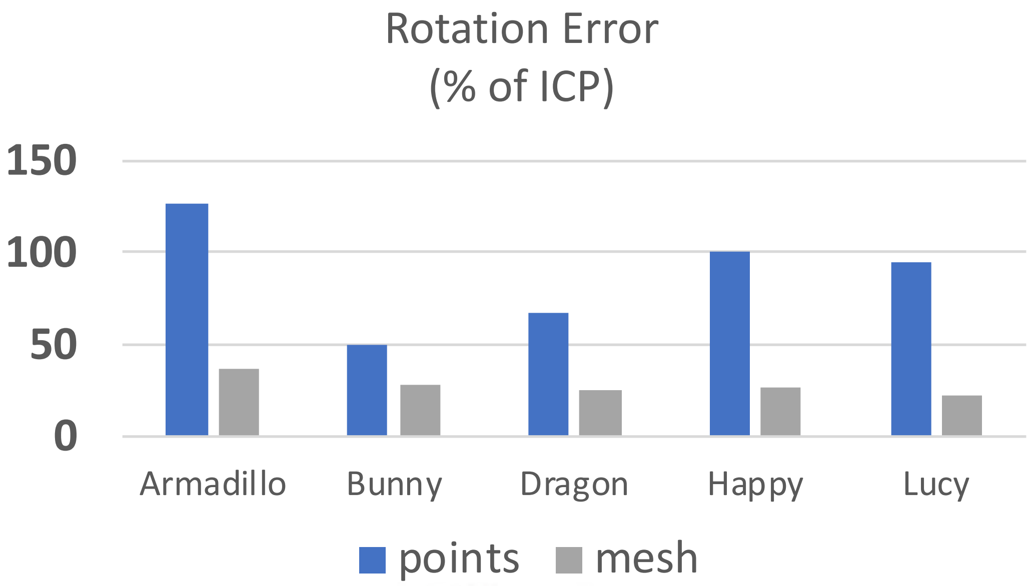
1. Apply a random rotation + translation to the point cloud
2. Find transformation to maximize the likelihood of the points
   - Perform P2D with GMMs fit to
     i. mesh vertices
     ii. mesh triangles

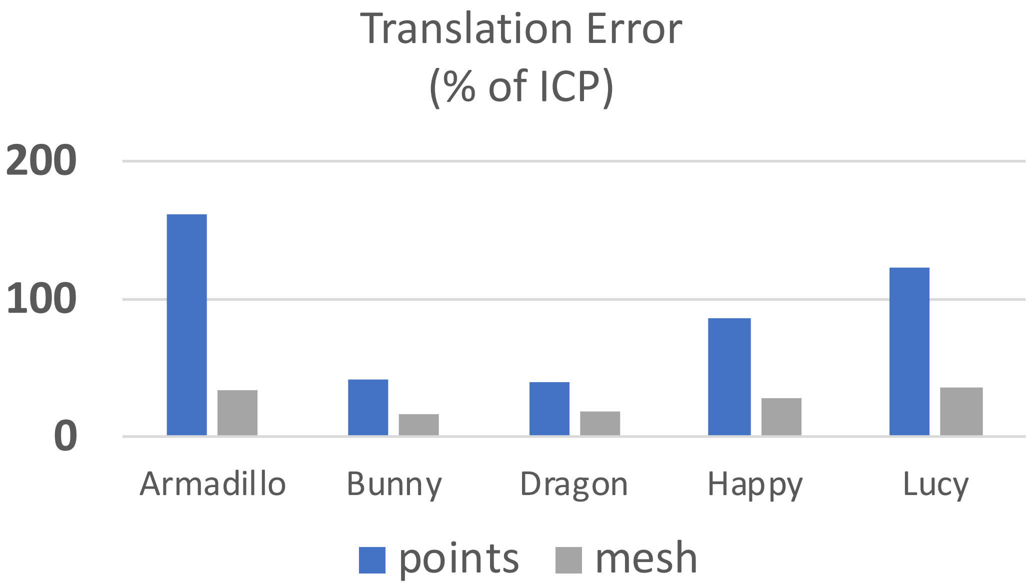
Eckart, Kim, Kautz.
"HGMR: Hierarchical Gaussian Mixtures for Adaptive 3D Registration."
ECCV (2018)

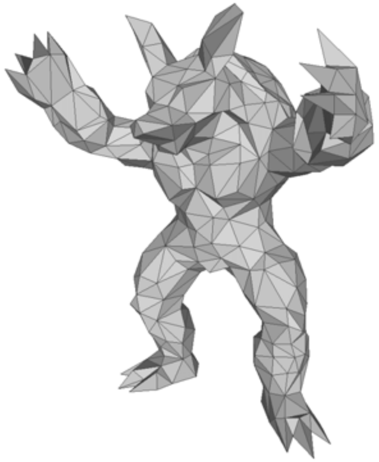
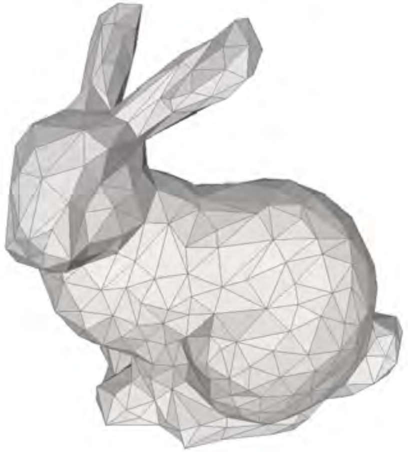
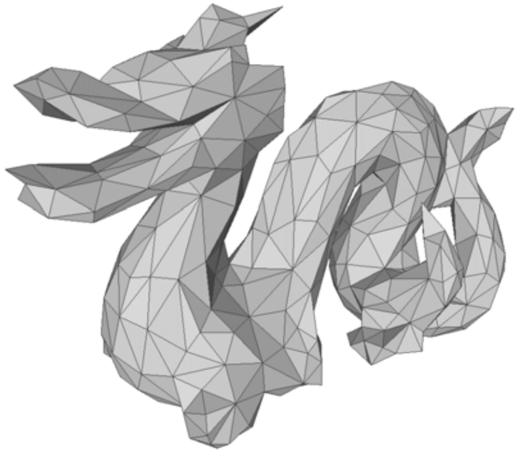
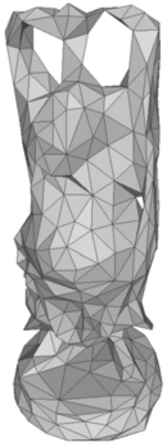
Mesh-based GMMs are more accurate



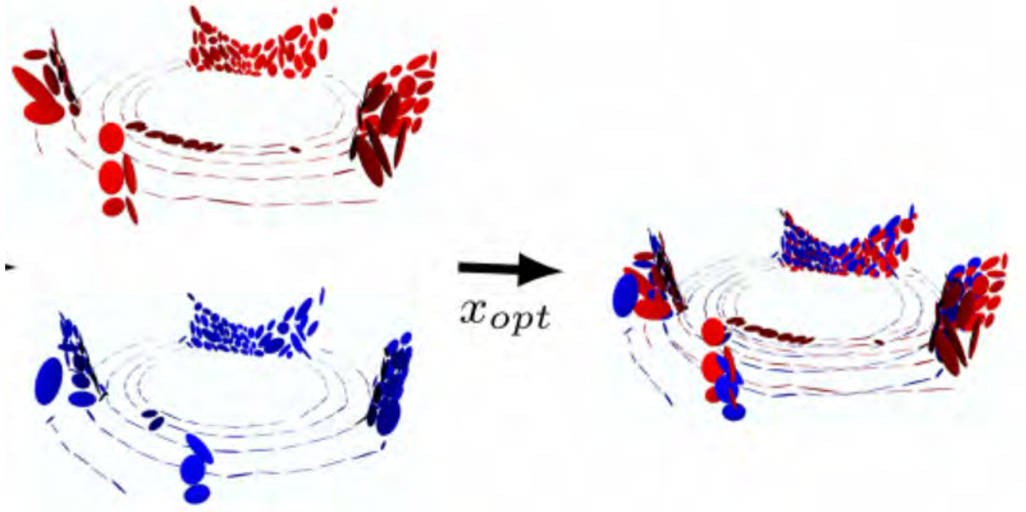
Across multiple models






Frame Registration (D2D)

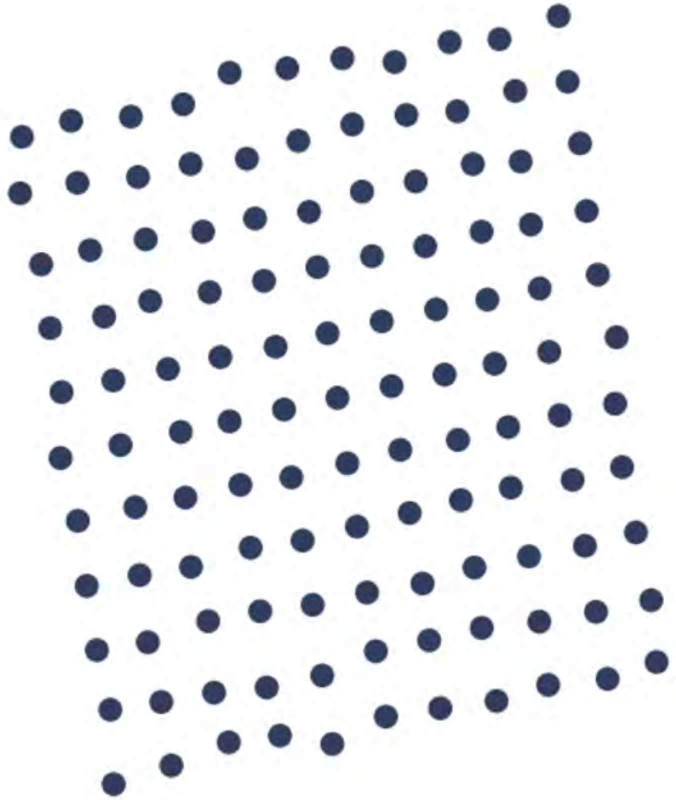


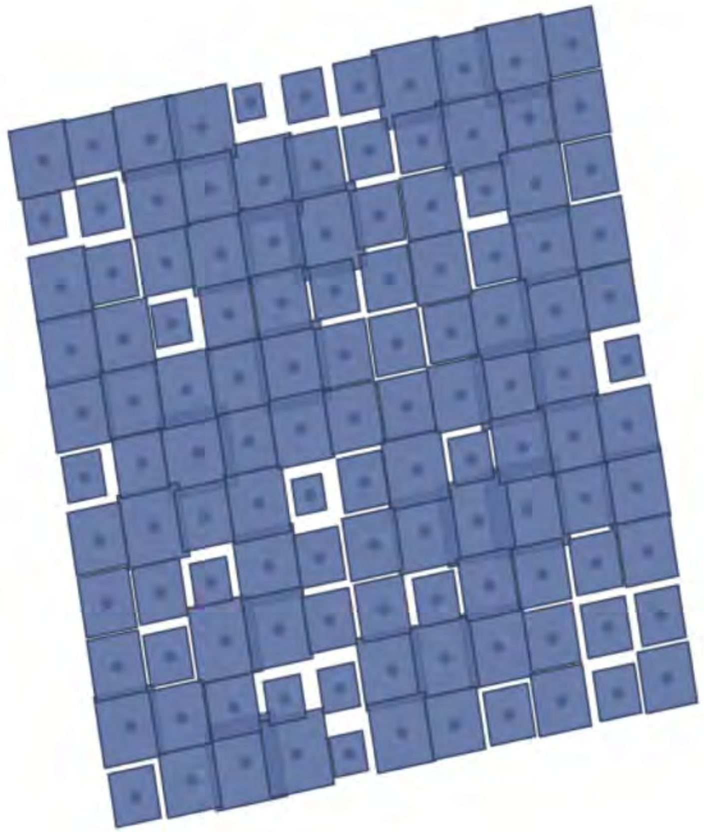
Method

1. Use a sequence from an RGBD Sensor
   - 2,500 frame TUM sequence from a Microsoft Kinect
2. Pairwise registration between t & t-1 frames
   - Optimize the D2D L2 distance
   - Build GMMs using square pixels as the geometric object

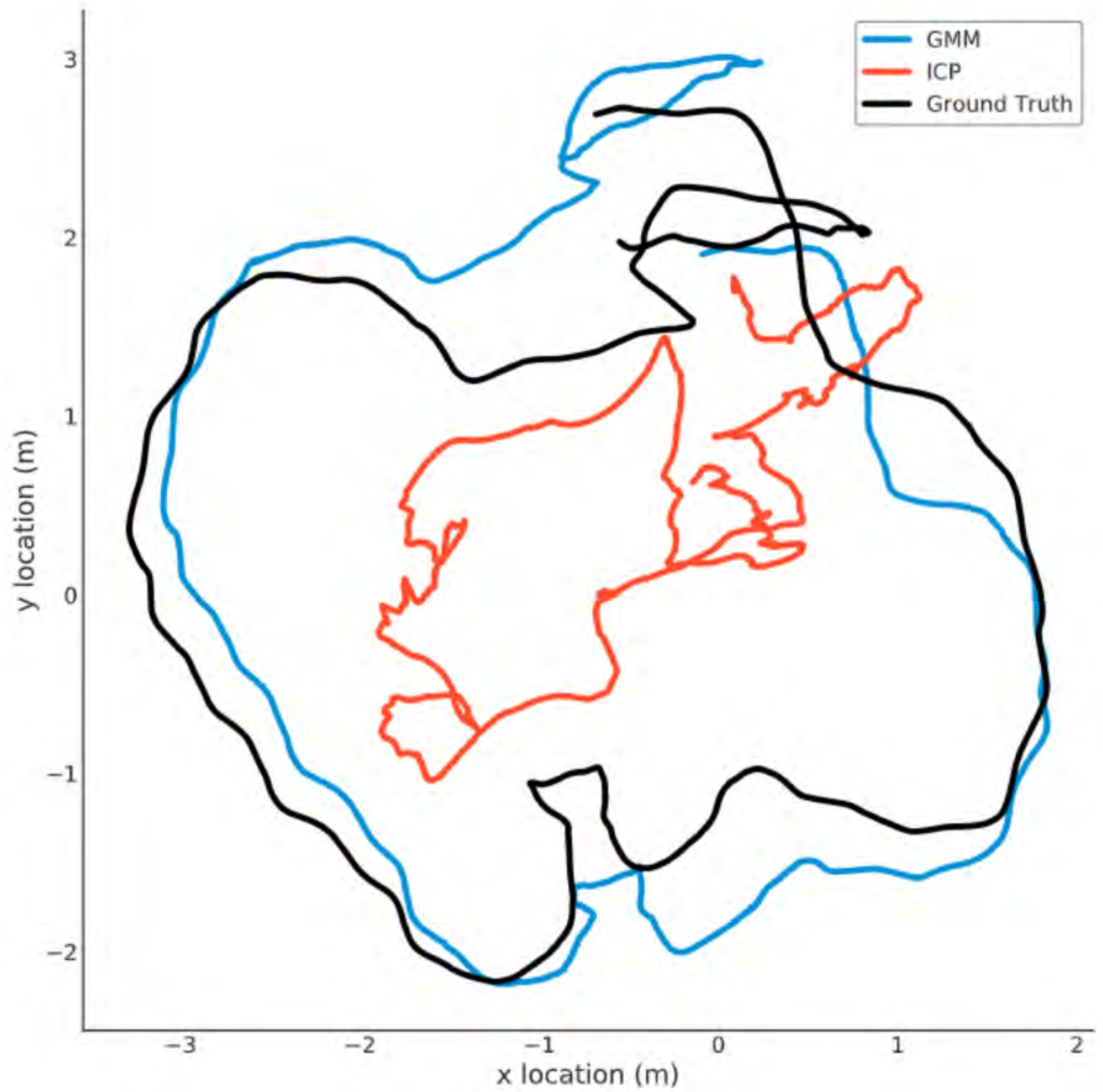
W. Tabib, C. O'Meadhra, N. Michael.
"On-Manifold GMM Registration"
IEEE R-AL (2018)

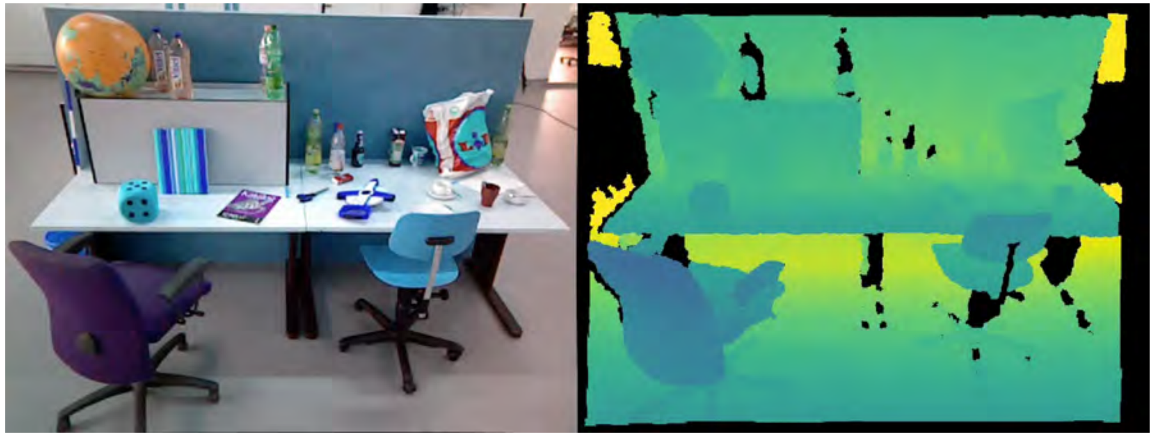
Representing points using pixel squares





D2D Registration Results

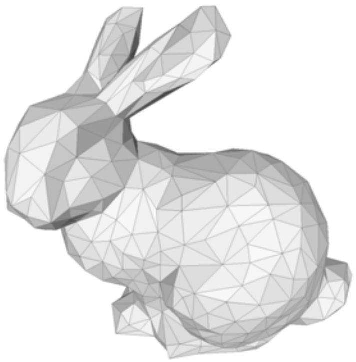




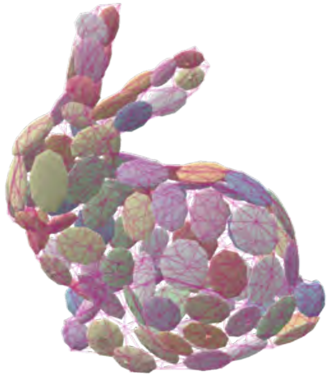
Compared to standard GMM

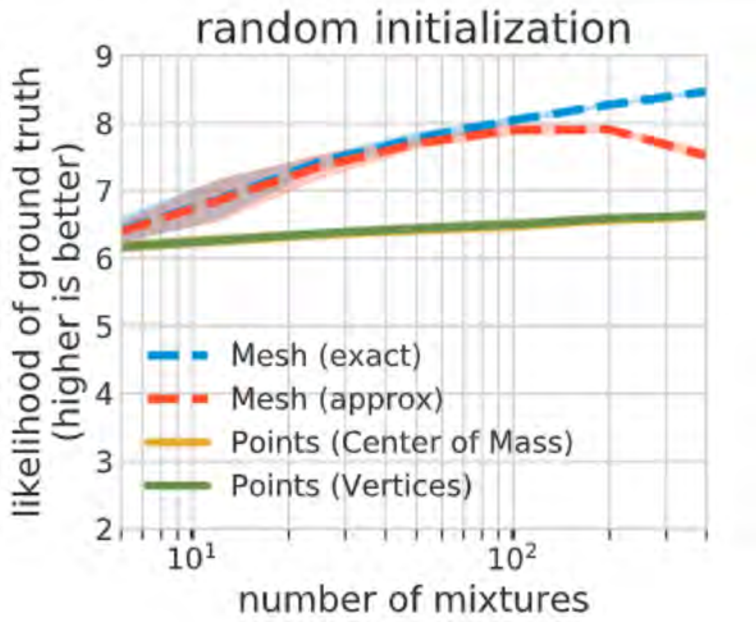
- 2.4% improvement in RMSE
- 22% faster D2D convergence

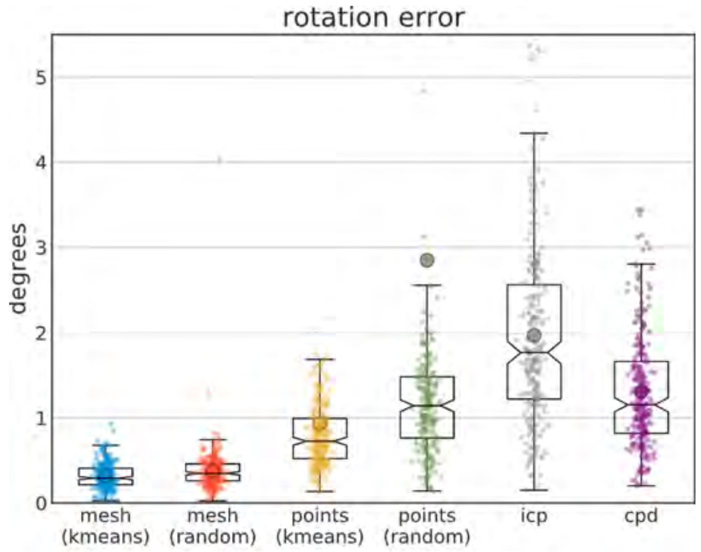
Questions?

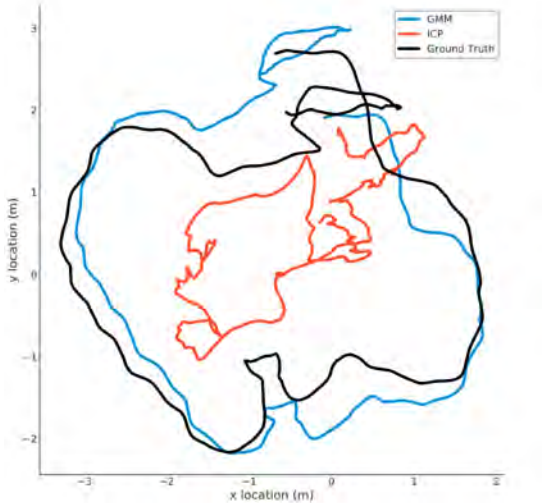


$$P = \exp\left(\int \log(p(x))\, dx\right)$$

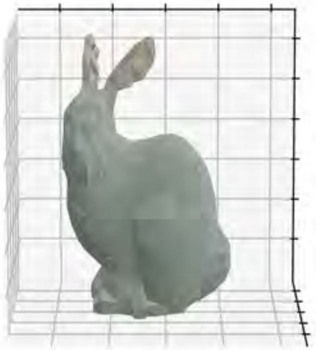




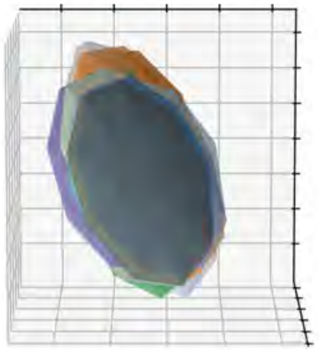




E-Step Result



M-Step Result

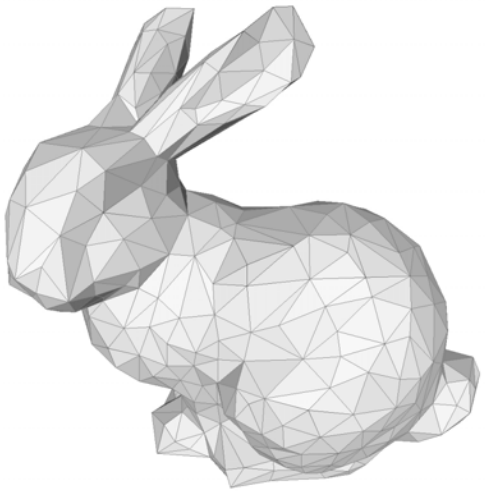


The End!

Extra Slides

How to fit a Gaussian Mixture Model?

1. Obtain any collection of objects
2. Perform Expectation + Maximization
   i. E-Step: Each point gets a likelihood
   ii. M-Step: Each mixture gets new parameters



Extension to arbitrary primitives

$$\mu_i = \frac{1}{W_i} \sum_p^P w_{ip} \mu_p$$

$$\Sigma_i = \frac{1}{W_i} \sum_p^P w_{ip} \left[(\mu_p - \mu_i)(\mu_p - \mu_i)^T + \Sigma_p \right]$$

Vasconcelos, Lippman. "Learning mixture hierarchies."
Advances in Neural Information Processing Systems (1999)

Approximation

$$L \approx L_S = \prod_{j=1}^{M} \left(\sum_{i=1}^{K} \pi_i \, \mathcal{N}(\mu_j; \mu_i, \Sigma_i) \right)^{\frac{\alpha_j}{\sum_k \alpha_k}}$$

area-weighted geometric mean using the primitive's centroids

Product Integral Formulation

- Product integrals provide a resampling-invariant loss function
- Given S samples, of M primitives, with N mixture components

$$L = \prod_{j=1}^{M} \prod_{k=1}^{S} \sum_{i=1}^{K} \pi_i \ \mathcal{N}(x_{jk}; \mu_i, \Sigma_i)$$

- This can be evaluated in the limit of samples (with a geometric mean)

$$\begin{aligned} L &= \prod_{j=1}^{M} \lim_{S \to \infty} \left[\prod_{k=1}^{S} \left(\sum_{i=1}^{K} \pi_i \ \mathcal{N}(x_{jk}; \mu_i, \Sigma_i) \right)^{\frac{1}{S}} \right] \\ &= \prod_{j=1}^{M} \lim_{S \to \infty} \left[\exp \left(\log \prod_{k=1}^{S} \left(\sum_{i=1}^{K} \pi_i \ \mathcal{N}(x_{jk}; \mu_i, \Sigma_i) \right)^{\frac{1}{S}} \right) \right] \\ &= \prod_{j=1}^{M} \lim_{S \to \infty} \left[\exp \left(\sum_{k=1}^{S} \frac{1}{S} \log \left(\sum_{i=1}^{K} \pi_i \ \mathcal{N}(x_{jk}; \mu_i, \Sigma_i) \right) \right) \right] \\ &= \prod_{j=1}^{M} \exp \left(\int_{\triangle} \log \left(\sum_{i=1}^{K} \pi_i \ \mathcal{N}(x; \mu_i, \Sigma_i) \right) dx \right) \end{aligned}$$

$$
\begin{aligned}
Q(\theta) &= \log \prod_{j=1}^{M} \sum_{i=1}^{N} p(x_j, z_i|\theta_i) \\
&= \sum_{j=1}^{M} \log \sum_{i=1}^{N} p(x_j, z_i|\theta_i) \\
&= \sum_{j=1}^{M} \log \sum_{i=1}^{N} \eta_{ij} \frac{p(x_j, z_i|\theta_i)}{\eta_{ij}} \\
&= \sum_{j=1}^{M} \log \mathbb{E}_{z|x,\theta} \left[\frac{p(x_j, z_i|\theta_i)}{\eta_{ij}} \right] \\
&\geq \sum_{j=1}^{M} \mathbb{E}_{z|x,\theta} \left[\log \frac{p(x_j, z_i|\theta_i)}{\eta_{ij}} \right] \\
&\geq \sum_{j=1}^{M} \sum_{i=1}^{N} \eta_{ij} \log \frac{p(x_j, z_i|\theta_i)}{\eta_{ij}} \\
&\geq \sum_{j=1}^{M} \sum_{i=1}^{N} \eta_{ij} \left(\log p(x_j|z_i, \theta_i) - \log \eta_{ij} \right) \\
&= \sum_{j=1}^{M} \sum_{i=1}^{N} \eta_{ij} \left(\log p(x_j|z_i, \theta_i) - \log \eta_{ij} \right)
\end{aligned}
$$

$$
\theta \leftarrow \operatorname{argmax} \sum_{j=1}^{M} \sum_{i=1}^{N} \eta_{ij} \log(\pi_i \mathcal{N}_i(x_{jk}; \mu_i, \Sigma_i))
$$

$$
\begin{aligned}
\phi_{\triangle}(h(x)) &= ||T_u \times T_v|| \int_0^1 \int_0^{1-v} f(T(u,v))\ dudv \\
&= ||T_u \times T_v|| \int_0^1 \int_0^{1-v} \mathcal{N}(M;\mu,\Sigma)(1-(T(u,v)-M)^T K_1 + (T(u,v)-M)^T K_2 (T(u,v)-M))dudv \\
&= ||T_u \times T_v|| \mathcal{N}(M;\mu,\Sigma) \int_0^1 \int_0^{1-v} (1-(T(u,v)-M)^T K_1) + (T(u,v)-M)^T K_2 (T(u,v)-M))dudv \\
&= ||T_u \times T_v|| \mathcal{N}(M;\mu,\Sigma)(\frac{1}{2} + \int_0^1 \int_0^{1-v} (-(T(u,v)-M)^T K_1 + (T(u,v)-M)^T K_2 (T(u,v)-M))dudv) \\
&= ||T_u \times T_v|| \mathcal{N}(M;\mu,\Sigma)(\frac{1}{2} - 0 + K_2 \int_0^1 \int_0^{1-v} (T(u,v)-M)^2 dudv) \\
&= ||T_u \times T_v|| \mathcal{N}(M;\mu,\Sigma)(\frac{1}{2} - 0 + K_2 \int_0^1 \int_0^{1-v} (A+(B-A)u+(C-A)v-M)^2 dudv) \\
&= ||T_u \times T_v|| \mathcal{N}(M;\mu,\Sigma)(\frac{1}{2} - 0 + \frac{K_2}{36}(A \circ (1-(B+C)) + B \circ (1-C) + C \circ C)) \\
&\approx \frac{||T_u \times T_v||}{2} \mathcal{N}(M;\mu,\Sigma)
\end{aligned}
$$

$$\frac{\partial LB}{\partial \Sigma_i^{-1}} = \frac{1}{2}\sum_{j=1}^{M}\int_{\triangle_j}\left[\eta_{ij}\left(\Sigma_i - (x_j-\mu_i)(x_j-\mu_i)^T\right)\right]d\triangle_j$$

$$= \frac{1}{2}\sum_{j=1}^{M}\left(R_j\eta_{ij}\Sigma_i - \eta_{ij}\int_{\triangle_j}\left[(x_j-\mu_i)(x_j-\mu_i)^T\right]d\triangle_j\right) \tag{25}$$

$$\left[\int_{\triangle}\left[(x-\mu)(x-\mu)^T\right]d\triangle\right]_{01} = \int_{\triangle}(x_0-\mu_0)(x_1-\mu_1)d\triangle$$

$$= \left[\frac{2R}{24}\left(A_0(2A_1+B_1+C_1)+B_0(A_1+2B_1+C_1)+C_0(A_1+B_1+2C_1)\right)+\frac{2R}{2}\left(-M_1\mu_0-M_0\mu_1+\mu_0\mu_1\right)\right]$$

$$= \left[\frac{2R}{24}\left(3M_0 3M_1+A_0A_1+B_0B_1+C_0C_1\right)+\frac{2R}{2}\left(-M_1\mu_0-M_0\mu_1+\mu_0\mu_1\right)\right]$$

$$= \left[\frac{2R}{24}\left(A_0A_1+B_0B_1+C_0C_1-3M_0M_1\right)+\frac{2R}{2}\left(M_0M_1-M_1\mu_0-M_0\mu_1+\mu_0\mu_1\right)\right]$$

$$\frac{\partial LB}{\partial \Sigma_i^{-1}} = \frac{1}{2}\sum_{j=1}^{M}\left(R_j\eta_{ij}\Sigma_i - \eta_{ij}R_j\left[(M_j-\mu_i)(M_j-\mu_i)^T+\frac{1}{12}(A_jA_j^T+B_jB_j^T+C_jC_j^T-3M_jM_j^T)\right]\right) \tag{26}$$

Setting this derivative to zero and solving gives us the following expression for the new covariance

$$\Sigma_i = \sum_{j=1}^{M}\frac{\eta_{ij}R_j\left[(M_j-\mu_i)(M_j-\mu_i)^T+\frac{1}{12}(A_jA_j^T+B_jB_j^T+C_jC_j^T-3M_jM_j^T)\right]}{\sum_{j=1}^{M}R_j\eta_{ij}}$$

$$= \sum_{j=1}^{M}\frac{\eta_{ij}R_j\left[(M_j-\mu_i)(M_j-\mu_i)^T\right]}{\sum_{j=1}^{M}R_j\eta_{ij}}+\frac{1}{12}\frac{\eta_{ij}R_j\left[(A_jA_j^T+B_jB_j^T+C_jC_j^T-3M_jM_j^T)\right]}{\sum_{j=1}^{M}R_j\eta_{ij}} \tag{27}$$

$$= \sum_{j=1}^{M}\frac{\eta_{ij}R_j}{\sum_{j=1}^{M}R_j\eta_{ij}}\left[\underbrace{(M_j-\mu_i)(M_j-\mu_i)^T}_{cov(M_j,\mu_i)}+\underbrace{\frac{1}{12}(A_jA_j^T+B_jB_j^T+C_jC_j^T-3M_jM_j^T)}_{cov(\triangle_j)}\right]$$

For GMMs we will use the lower bound

$$L = \exp\left(\sum_{j=1}^{M}\int_{\triangle}\log\left(\sum_{i=1}^{K}\pi_i\,\mathcal{N}(x;\mu_i,\Sigma_i)\right)dx\right)$$

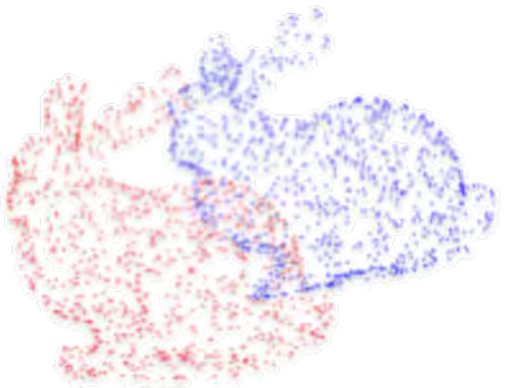
$$\log(L) = \sum_{j=1}^{M}\int_{\triangle}\log\left(\sum_{i=1}^{K}\pi_i\,\mathcal{N}(x;\mu_i,\Sigma_i)\right)dx$$

$$\geq \sum_{j=1}^{M}\sum_{i=1}^{K}\int_{\triangle}\log\left(\pi_i\,\mathcal{N}(x;\mu_i,\Sigma_i)\right)dx$$

P2D Registration Results

Model	Rotation Error (% of ICP)		Translation Error (% of ICP)	
	points	mesh	points	mesh
Armadillo	127	**37**	161	**33**
Bunny	50	**28**	41	**17**
Dragon	68	**25**	40	**19**
Happy	101	**27**	85	**27**
Lucy	95	**23**	122	**35**

Mesh Registration with P2D



Method

1. Apply a random rotation + translation to the point cloud
2. Point-to-Distribution (P2D) registration of point cloud to GMM
   - Perform tests with GMMs fit to
     i. mesh vertices
     ii. mesh triangles
   - Optimize the GMM likelihood with rigid body transformation (q & t)
   - BFGS Optimization using numerical gradients, starting from identity

Eckart, Kim, Kautz.
"HGMR: Hierarchical Gaussian Mixtures for Adaptive 3D Registration."
ECCV (2018)

Acknowledgements

- Martial Hebert
- Robotics Institute
- Reviewers
- Funding?

Representing points using pixel squares