Continuous Projection for Fast L_1 Reconstruction

Reinhold Preiner* Oliver Mattausch† Murat Arikan* Renato Pajarola† Michael Wimmer*

*Vienna University of Technology †University of Zurich

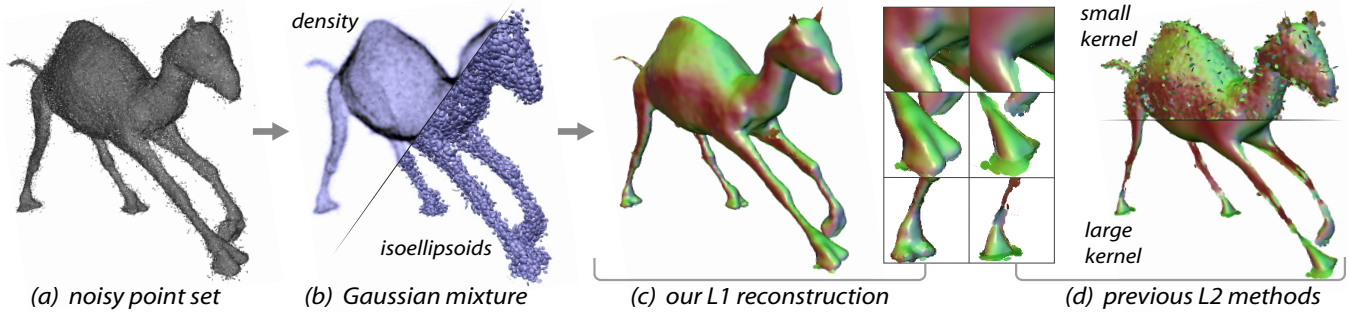


Figure 1: *From a noisy input point set of 87K points (a) we hierarchically build a reduced Gaussian mixture representing its density (b). On this mixture, a* Continuous Projection *operator is applied, which efficiently produces an L_1 reconstruction of 72K point positions (c) at ~ 9 FPS. In contrast, an L_2 reconstruction (d) with small feature-preserving kernel exhibits heavily visible noise (top), while a larger kernel biases and oversmoothes the result (bottom). Our method runs at up to 7 times the speed of a fast GPU implementation of standard WLOP while providing comparable or even better quality, allowing for interactive robust reconstruction of unordered dynamic point sets.*

Abstract

With better and faster acquisition devices comes a demand for fast robust reconstruction algorithms, but no L_1-based technique has been fast enough for online use so far. In this paper, we present a novel continuous formulation of the weighted locally optimal projection (WLOP) operator based on a Gaussian mixture describing the input point density. Our method is up to 7 times faster than an optimized GPU implementation of WLOP, and achieves interactive frame rates for moderately sized point clouds. We give a comprehensive quality analysis showing that our continuous operator achieves a generally higher reconstruction quality than its discrete counterpart. Additionally, we show how to apply our continuous formulation to spherical mixtures of normal directions, to also achieve a fast robust normal reconstruction.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—[Geometric algorithms, languages, and systems]

Keywords: L1 reconstruction, point set, Gaussian mixture, Hierarchical EM, dynamic reconstruction, upsampling

Links: DL PDF

*e-mail:{preiner|marikan|wimmer}@cg.tuwien.ac.at

†e-mail:{mattausch|pajarola}@ifi.uzh.ch

1 Introduction

In recent years, many robust surface reconstruction techniques have been developed that can deal with a variety of acquisition errors like noise, outliers, missing data (holes) or registration artifacts. They are commonly based on a robust L_1-optimization approach and are able to produce high-quality output despite very difficult data. However, current L_1 techniques are typically too expensive to achieve interactive reconstruction times for at least moderately sized point sets, even for parallel implementations. Hence, due to their nature, they are designed for quality rather than performance. The availability of modern online acquisition devices has however created new research challenges in performing instant surface reconstruction of such dynamic point data. A variety of methods have recently been proposed for online L_2 reconstruction of static [Izadi et al. 2011] as well as dynamic scenes [Zhou et al. 2008]. However, a major challenge for a dynamic point-based reconstruction technique is its robust performance in the face of corrupt and noise-contaminated data, as is commonly given in real-world scanning scenarios due to sensor accuracy, occlusions, imperfect registration and other issues.

In this paper, we introduce a highly efficient variant of the *locally optimal projection* (LOP) operator [Lipman et al. 2007]. LOP robustly fits a set of resampling *particles* to a noisy point cloud by iteratively applying a system of attractive forces defined by the input points. Characteristically, LOP requires high computational effort for the iterative evaluation of all mutual forces between the particles and the *discrete* set of points. Our approach reformulates this operator to be applicable to a much more compact, *continuous* representation of the point cloud's attractive potential field. We use a *Gaussian mixture model* (GMM) to describe the point cloud's density in a geometry-preserving manner and show how to compute an *efficient analytic solution* to the integral forces exerted on the particles by each of its continuous components. By operating on a small set of Gaussians instead of the large input point cloud, *Continuous LOP* (CLOP) achieves speedups of up to a factor of 7 over a comparable LOP implementation on the GPU. This makes a robust reconstruction at interactive frame rates for moderately sized dynamic point sets possible (see Figure 1). We also demonstrate that the

same continuous formulation can be directly applied to the spherical domain to efficiently compute *locally robust point normals* as well. Furthermore, our continuous representation allows for robust point-cloud upsampling. Our results show that despite its much faster computation, our continuous algorithm achieves better point regularity and equal or even higher reconstruction accuracy than its discrete counterpart and even high-quality variants like *Weighted LOP* [Huang et al. 2009].

2 Related Work

Surface Reconstruction from point clouds has been a major research topic for 20 years. Implicit surface reconstruction methods range from locally fitting tangent planes [Hoppe et al. 1992], using Radial Basis Functions [Carr et al. 2001], or Poisson Reconstruction [Kazhdan et al. 2006; Kazhdan and Hoppe 2013]. Point set surfaces based on a moving least squares formulation (MLS) have been successfully used to resample and smooth point clouds [Alexa et al. 2003; Amenta and Kil 2004] in the presence of noise, but they have problems handling outliers. L_1-based reconstruction techniques have gained a lot of attention over the last years in many fields, as they are known to be less sensitive to the presence of outliers. For surface reconstruction, methods have been proposed that use recent advances in robust statistics [Fleishman et al. 2005; Oztireli et al. 2009]. These methods are theoretically able to faithfully reconstruct a surface as long as there are less than 50 percent outliers (the breakdown point). Other methods perform a global minimization on the orientation of the input normals, using ideas from compressed sensing and sparse signal recovery [Avron et al. 2010]. However, while these algorithms are of high quality, due to their global nature they are often extremely slow, do not scale well to large data or require special assumptions on the input data (e.g., data with a few planar elements).

LOP and Variants. The Locally Optimal Projection (LOP) operator [Lipman et al. 2007] is particularly attractive for reconstruction since it does not put many constraints on the nature of the input data, i.e., it does not require a well-defined surface parametrization nor a surface which can be locally well approximated with a plane. As the algorithm is related to the Weiszfeld-algorithm for finding the L_1-median [Hartley et al. 2011], it is robust against outliers, but still relatively cheap and highly parallelizable because of its local support. However, the running times of the original algorithm are still infeasible for interactive applications. Weighted LOP (WLOP) [Huang et al. 2009] deals with unevenly sampled point clouds by taking into account a local density measure which relaxes the attractive forces in denser regions and hence reaches more evenly distributed points. We will show that this concept is compatible with our continuous LOP formulation and can be adopted without any additional effort. Kernel LOP (KLOP) [Liao et al. 2013] reduces the computation cost of LOP by *subsampling* the point cloud using a *kernel density estimate* (KDE). While this reduction achieves a decent acceleration, reducing the number of discrete input samples also constrains the number of usable resampling particles [Lipman et al. 2007], thus the general reconstruction quality deteriorates quickly for a small number of kernels. As our approach describes the point set's whole KDE continuously, we are able to reconstruct at high particle rates using only few Gaussian components.

GPU-based Online Reconstruction. With the introduction of real-time 3D acquisition devices, interactive reconstruction has become a highly important topic. The recently developed KinectFusion algorithm [Izadi et al. 2011] builds a complete volumetric model of the environment by integrating range data over time in a 3D grid. The data can then be used for tracking and rendering by raycasting the implicit surface in the grid, but it only handles a limited amount of dynamics in the scene. For dynamic point sets, Zhou et al. [2008] build a GPU kD-tree in each frame to perform range queries for a per-frame computation of the k-nearest neighbors. Auto Splats [Preiner et al. 2012] accelerate the normal estimation process by executing the same range queries entirely in image space, discarding the necessity for a per-frame kD-tree construction.

Gaussian Mixtures. To describe complex data distributions using a simple model, Gaussian mixture models have been used in various scientific fields, like image segmentation [Garcia et al. 2010], object recognition [Vasconcelos 1998] and rendering [Walter et al. 2008; Jakob et al. 2011]. We employ this model to obtain a simple but expressive representation of the point distribution in an input point set. We use a hierarchical version [Vasconcelos 1998] of the Expectation-Maximization (EM) algorithm [Dempster et al. 1977], which quickly computes a maximum likelihood estimate of the mixture component parameters.

Robust Normal Estimation. Normal estimation is a fundamental problem in surface reconstruction and point rendering. Basic approaches use some form of local plane fitting [Hoppe et al. 1992], but noisy point sets with outliers and possible sharp features require more robust normal estimations. Approaches range from inscribing empty balls [Dey and Sun 2006], smoothing and outlier removal [Huang et al. 2009], global L_1 norm optimization [Avron et al. 2010] to randomized Hough transforms [Boulch and Marlet 2012]. Robust statistics-based methods have been shown to achieve superior results in the presence of outliers [Kalogerakis et al. 2007; Li et al. 2010; Zheng et al. 2010; Oztireli et al. 2009]. In this paper, we will show that our proposed continuous LOP operator can be directly adopted to not only perform a fast robust resampling of a point cloud, but also an equally efficient robust reconstruction of point normals.

3 A Review of the LOP Operator

The *Locally Optimal Projection* (LOP) Operator [Lipman et al. 2007] fits a number of points $Q = \{q_i\}_{i\in I}$ (denoted as *particles*) into local medians of a point set $P = \{p_j\}_{j\in J}$, I and J being the respective index sets. The algorithm performs a localized version of Weiszfeld's algorithm for finding the spatial median $q = \underset{\mathbf{x}}{\operatorname{argmin}}\{\Sigma_{j\in J}\|\mathbf{x} - \mathbf{p}_j\|\}$ using a steepest descent on the sum of Euclidean distances from all points p_j. To extend the Weiszfeld algorithm to multiple particles, LOP uses an isotropic, fast decaying localization kernel $\theta(r) = e^{-r^2/(h/4)^2}$ around each particle, which concentrates its influence onto its support radius h. Starting with an arbitrary initial particle set $Q^{(0)}$, LOP computes the target particle positions Q by performing a fixed-point iteration

$$Q^{(k+1)} = \underset{X=\{x_i\}_{i\in I}}{\operatorname{argmin}} \{E_1(X, P, Q^{(k)}) + E_2(X, Q^{(k)})\} \quad (1)$$

where
$E_1(X, P, Q^{(k)}) = \sum_{i\in I}\sum_{j\in J}\|x_i - p_j\|\theta(\|q_i - p_j\|)$,
$E_2(X, Q^{(k)}) = \sum_{i'\in I}\lambda_{i'}\sum_{i\in I\setminus\{i'\}}\eta(\|x_{i'} - q_i\|)\theta(\|q_{i'} - q_i\|)$.

Here, E_1 is an energy term attracting Q towards the local medians of P, while E_2 defines a repulsive energy between the particles that strives for an equal distribution of the q_i over the approximated surface. θ denotes the localization kernel constraining both terms to a finite influence radius, $\{\lambda_i\}_{i\in I}$ are weights balancing the particles' attractive and repulsive forces, and η is a repulsion function determining a distance-based repulsion strength (we use $\eta(r) = -r$ as

suggested by Huang et al. [2009]). Eq. (1) leads to the following formulation for the updates of each particle $q_i^{(k)} \in Q^{(k)}$ in iteration k. The first iteration acts as an L_2 initializer for Q,

$$q_i^{(1)} = \frac{\sum_{j\in J} p_j \theta(\|p_j - q_i^{(0)}\|)}{\sum_{j\in J} \theta(\|p_j - q_i^{(0)}\|)}, \quad i \in I, \tag{2}$$

followed by the fixed-point iteration updates

$$q_i^{(k+1)} = F_1(q_i^{(k)}, P) + \mu\, F_2(q_i^{(k)}, Q_i'^{(k)}) \tag{3}$$

$$F_1(q, P) = \sum_{j\in J} p_j \frac{\alpha_j}{\sum_{j'\in J} \alpha_{j'}} \tag{4}$$

$$F_2(q, Q_i') = \sum_{i'\in I\setminus\{i\}} (q - q_{i'}) \frac{\beta_{i'}}{\sum_{i''\in I\setminus\{i\}} \beta_{i''}} \tag{5}$$

where Q_i' denotes the set of complementary particles $Q\setminus\{q_i\}$. The repulsion parameter $\mu \in [0, 0.5)$ controls the balancing between the attractive forces F_1 of the points p_j and the repulsive forces F_2 from the neighboring particles q_i. Both forces are defined as convex sums over their respective neighbors, with pairwise weights

$$\alpha_j = \frac{\theta(\|p_j - q\|)}{\|p_j - q\|}, \;\; \beta_{i'} = \frac{\theta(\|q - q_{i'}\|)}{\|q - q_{i'}\|} \left|\frac{\partial \eta}{\partial r}(\|q - q_{i'}\|)\right|. \tag{6}$$

Huang et al. [2009] proposed an improved, *weighted* version of LOP, referred to as WLOP, that introduces additional balancing of these weights, allowing for a more uniform distribution of the particles in regions of varying point density. We show in Section 7 that this additional balancing can be natively integrated into our analytic approach, enabling us to actually perform analytic WLOP. However, for the rest of this paper, we will refer to our method as *Continuous LOP* (CLOP) for simplicity.

4 Motivation and Overview

The formulation of Eq. (1) can be interpreted as a particle simulation of a set of repulsive particles Q on an attractive background potential field Π, which is represented by a discrete set of samples P. The computational effort of LOP scales with the number $|P|$ of points and the number $|Q|$ of resampling particles to be processed in order to evaluate all mutual forces in the system. As typically $|Q| \ll |P|$, the majority of the time will be spent on the evaluation of the attractive forces from all points p_j, which can be seen as the carriers of the energy potential of Π. We therefore propose to reduce Π to a more compact, yet still accurate representation, which allows evaluating the attraction term much more efficiently.

In this paper, we use a mixture $\mathcal{M}$ of anisotropic Gaussians to represent the density of the input points, where $|\mathcal{M}| \ll |P|$ (Section 5). We then derive an analytic solution for the continuous attraction forces exerted by each individual Gaussian (Section 6). The mixture is efficiently computed from the input points P by a constrained hierarchical expectation-maximization procedure. This one-time effort easily pays off, considering the reduced amount of density-representing entities to process during the following LOP iterations (typically 10–20). In Section 7, we show how to extend our approach to WLOP without additional cost. In Section 8, we exploit an inherent coherency in the repulsive moments of the particles to also accelerate the evaluation of the repulsion term over all LOP iterations. Finally, we extend our continuous formulation to the robust estimation of normals (Section 9).

5 Gaussian Mixture Density Computation

In this section, we efficiently reduce the set P of unordered input points to a much more compact mixture of Gaussians $\mathcal{M} = \{w_s, \Theta_s\}$ that reflects the density distribution of the points. That is, $\mathcal{M}$ defines a probability density function (pdf) as a weighted sum of $|\mathcal{M}|$ Gaussian components

$$f(\mathbf{x}|\mathcal{M}) = \sum_s w_s g(\mathbf{x}|\Theta_s), \tag{7}$$

where the $\Theta_s = (\mu_s, \Sigma_s)$ are the Gaussian parameters, w_s their corresponding convex weights, and g denotes the d-variate Gaussian pdf with $g(\mathbf{x}|\mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$. We demand $\mathcal{M}$ to be efficiently computable in parallel, and to ideally reflect the density of P while minimizing the smoothing of the signal which LOP tries to reconstruct. We use a constrained variant of *hierarchical expectation maximization* [Vasconcelos 1998], which aims at optimizing $\mathcal{M}$ in the maximum-likelihood sense while trying not to destroy characteristic information about the underlying geometry. Next, we will shortly review the EM and hierarchical EM (HEM) algorithms, and then present a modified, constrained variant of HEM to compute an accurate density estimate of the point cloud.

5.1 Expectation Maximization in GMMs

An ideal density estimate $\mathcal{M}$ of an input point set $\{\mathbf{p}_i\}$ is defined in a way that it maximizes the likelihood $\mathcal{L}(\mathcal{M}) = \prod_i f(\mathbf{p}_i|\mathcal{M})$ of producing the set P under $\mathcal{M}$. Starting with an initial guess $\mathcal{M}^{(0)}$, Expectation Maximization [Dempster et al. 1977] computes such a *maximum likelihood estimate* (MLE) for mixture models by iteratively optimizing an estimator of the parameters of $\mathcal{M}$ until a local maximum of the objective log-likelihood function $\mathcal{L}_{log} = log\ \mathcal{L}(\mathcal{M})$ is found. It thereby uses a discrete distribution of posterior responsibility probabilities r_{is} for a fuzzy assignment of each point to each component, and optimizes them along with the model parameters. This is done in an alternating two-step procedure, which eventually converges to a local maximum of $\mathcal{L}_{log}$:

E-Step: Given the current model parameters $\mathcal{M}$, compute the expected responsibilities

$$r_{is} = \frac{\mathcal{L}(\Theta_s|\mathbf{p}_i) w_s}{\sum_{s'} \mathcal{L}(\Theta_{s'}|\mathbf{p}_i) w_{s'}}, \quad \mathcal{L}(\Theta_s|\mathbf{p}_i) = g(\mathbf{p}_i|\Theta_s) \tag{8}$$

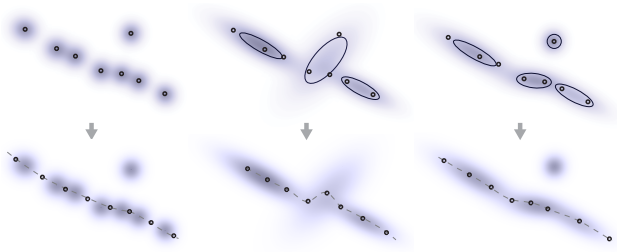
M-Step: Based on the new responsibilities, update the model parameters $\mathcal{M}'$. For Gaussian components, these are the points' weighted means μ_s and weighted covariances Σ_s with convex weights $r_{is}/\sum_{i'} r_{i's}$, and mixture coefficients $w_s = \sum_i r_{is}/|P|$.

5.2 Hierarchical EM

In contrast to classic EM, hierarchical EM performs only one initial EM iteration on the complete input data, and then successively reduces the mixture by hierarchically applying EM on Gaussians instead of points. HEM equips each input point $\mathbf{p}_i$ with an initial low-variance Gaussian $\Theta_i^{(0)}$, which results in an initial mixture $\mathcal{M}^{(0)} = \{w_s^{(0)}, \Theta_s^{(0)}\}$ with initially equal component weights $w_s^{(0)} = 1/|P|$. The component parameters $\Theta_s^{(l+1)}$ of the next level are then estimated based on the current level l by an adapted EM step. Since each Gaussian Θ_i represents $\overline{w}_i = w_i|P|$ points, HEM alters the likelihood function $\mathcal{L}$ employed in the E-Step in Eq. (8) to incorporate $\overline{w}_i$ representative "virtual samples":

$$\mathcal{L}(\Theta_s^{(l+1)}|\Theta_i^{(l)}) = \left[g(\mu_i^{(l)}|\Theta_s^{(l+1)})\, e^{-\frac{1}{2} tr([\Sigma_s^{(l+1)}]^{-1}\Sigma_i^{(l)})}\right]^{\overline{w}_i} \tag{9}$$

Given the responsibilities r_{is} and the mixture $M^{(l)}$ of the current level, the model parameters of the next higher level are again max-



(a) (b) (c)

Figure 2: *Gaussian Mixture on a signal with an outlier (top) and its LOP reconstruction (bottom). Ellipses denote Gaussians' one-σ-isodistances. (a) Initial mixture. (b) Level $\mathcal{M}^{(1)}$ with unconstrained HEM. (c) $\mathcal{M}^{(1)}$ with constrained clustering radius. Standard HEM tends to smooth the signal, while regularized HEM is more feature preserving, at the cost of less component reduction.*

imized by convex sums

$$w_s^{(l+1)} = \sum_i r_{is}\, w_i \qquad \mu_s^{(l+1)} = \sum_i \omega_{is}\mu_i$$

$$\Sigma_s^{(l+1)} = \sum_i \omega_{is}\left(\Sigma_i^{(l)} + (\mu_i^{(l)} - \mu_s^{(l+1)})(\mu_i^{(l)} - \mu_s^{(l+1)})^T\right) \quad (10)$$

with convex weights $\omega_{is} = r_{is}w_i / \sum_{i'} r_{i's}w_{i'}$. To initialize the mixture $\mathcal{M}^{(l+1)}$ of each next higher level before the hierarchical EM-step, we randomly subsample the set $\mathcal{M}^{(l)}$ (usually $\sim 33\%$).

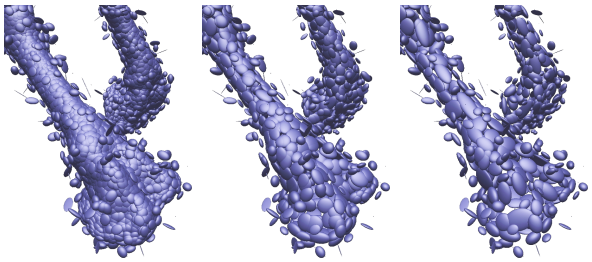
5.3 Geometrically Regularized HEM

Since the maximum likelihood estimate of a Gaussian is a least-squares solution, thus non-robust, an ordinary MLE of a Gaussian Mixture is inherently prone to bias the input signal in a way that obliterates any subsequent robust reconstruction. Thus, without any further consideration, a mere *statistically* optimal fit of $\mathcal{M}$ could place a Gaussian component in a way that blurs the information of outliers against which we want to robustly reconstruct (Figure 2(a) and 2(b)). While there are alternative distributions that provide a robust MLE, like the Laplace distribution, these cannot be expressed in closed form and would thus require an expensive iterative approximation. Instead, we improve the robustness of the Gaussian mixture by adopting a *geometric* regularization to Hierarchical EM, which stems from the idea of agglomerative hierarchical clustering to merge only those clusters which are closest under a given distance measure. Restricting the set of neighboring components that are considered for merging to a *clustering kernel* of finite radius ρ allows merging the energy mass of close-by Gaussians, while leaving more distant clusters untouched (Figure 2(c)). This results in a *regularized hierarchical EM* procedure, which strives for a maximum likelihood estimate under a reinforced similarity constraint.

Dissimilarity Measure To measure the distance between two Gaussians Θ_t and Θ_s in $\mathbb{R}^3$, we use their Kullback-Leibler divergence

$$D_{KL}(\Theta_t \| \Theta_s) = \tfrac{1}{2}\left(d_M(\mu_t, \Theta_s)^2 + tr(\Sigma_s^{-1}\Sigma_t) - 3 - \ln\tfrac{|\Sigma_t|}{|\Sigma_s|}\right) \quad (11)$$

and define ρ to be the maximum distance $D_{KL}(\Theta_t \| \Theta_s)_{max}$ within which Θ_s might merge other components Θ_t. Although D_{KL} is a measure of relative entropy, it has an intuitive geometric interpretation, as it accounts for both the scale-invariant Mahalanobis distance d_M between their centers as well as the deviation of their principal component directions. Thus, D_{KL} lets large anisotropic Gaussians continue clustering in the direction of their largest variance, while smaller Gaussians, possibly representing outlier points, are restricted to a small clustering radius.



(a) $|\mathcal{M}^{(0)}| = 64K$ (b) $|\mathcal{M}^{(2)}| = 13K$ (c) $|\mathcal{M}^{(4)}| = 8400$

Figure 3: *Unit-σ-isosurfaces of the mixture Gaussians at the camel model's front hooves at different hierarchy levels for $\alpha = 2.1$. Note how with successive levels, the main signal components are merged while the Gaussians modeling outlier information stay unchanged.*

Clustering Kernel Size In contrast to previous authors [Jakob et al. 2011; Walter et al. 2008], who choose ρ to be the n-th globally smallest occurring distance between Gaussians, we try to avoid such a global computation, but rather choose ρ to be a good compromise between clustering efficiency (large, relaxed ρ), and geometric accuracy (small, restrictive ρ). To provide an intuitive control over ρ, we suggest a free parameter α, so that $\rho = \alpha^2/2$, which has a simple interpretation: If two Gaussians have equal covariances, thus presumably representing similarly oriented geometry, Eq. (11) reduces α to a simple threshold of their centers' Mahalanobis distance. On the other hand, assuming the Gaussians have coinciding centers, differently oriented covariances suggest a change in orientation of the underlying surface, which α will segregate even more. In our experiments, $\alpha \approx 2$ has proven to give a good balance between clustering efficiency and accuracy.

Mixture Initialization The initial mixture $\mathcal{M}^{(0)}$ needs to be defined in a way that allows α to provide a similar regularization behavior throughout all levels of the hierarchy. Placing an initial Gaussian at each point ($\mu_i^{(0)} = p_i$) creates a simple kernel density estimate of $|P|$, whose kernel bandwidth defines the extent of the covariances $\Sigma_i^{(0)}$ [Vasconcelos 1998]. A too small bandwidth requires a large α to allow any clustering at all, but also diminishes the regularization effect in subsequent levels. On the other hand, a too large bandwidth smooths the signal in advance, thus again increasing the reconstruction bias. To produce a suitable initial density estimate, we first use a conservative kernel radius r (usually $2 \sim 3$ times a point's nearest neighbor distance) to compute for each point an initial covariance matrix Σ_r, whose shape already reflects the local distribution of the n points within the kernel. In a second step, Σ_r is scaled down so that its unit-σ-ellipsoid fits its local nearest neighbor distances. This gives an initial covariance

$$\Sigma^{(0)} = \Sigma_r \frac{r}{\sigma_{max}\sqrt[3]{n}} + c\, I, \quad (12)$$

where σ_{max} is the square root of the largest eigenvalue of Σ_r, I is the identity matrix, and c is a small trace bias giving $\Sigma^{(0)}$ a minimum extent. Figure 3 shows the feature-preserving reduction of the mixture from Figure 1 over different hierarchy levels with regularized HEM.

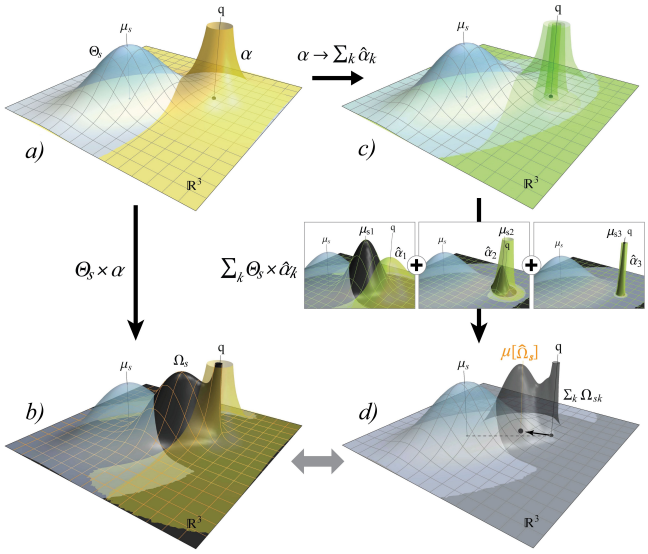
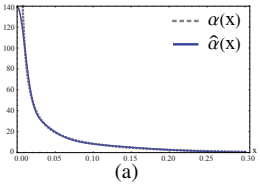


Figure 4: *Continuous attraction of a particle q from a Gaussian Θ_s. (a) Density of Θ_s (blue) and kernel α (yellow) centered in q lead to (b) a product weight Ω_s (black) with infinite integral. (c) Approximation of α by a sum of 3 Gaussians (green) divides this integral into 3 product Gaussians Ω_{sk} (black). The sum of their means, convexly weighted by their integrals, yields (d) the estimated mean of Ω_s, which is the destination point of q intended by the attraction of Θ_s. Note the good approximation quality of $\hat{\Omega}_s$ (d) compared to Ω_s (b), with only 3 Gaussian summands.*



(a)

k	$\hat{w}_k$	$\hat{\sigma}_k$
1	11.453	0.11772
2	29.886	0.03287
3	97.761	0.01010

(b)

Figure 5: *(a) Comparison plot of the original function α (dashed) and its approximation by a sum of 3 Gaussians (solid) using the coefficients listed in (b). Note the finite peak of $\hat{\alpha}$.*

6 Continuous LOP in Gaussian Mixtures

In this section we show how to apply the robust LOP operator to a mixture of Gaussians. By reformulating and evaluating the attractive force F_1, we obtain our accelerated CLOP algorithm.

6.1 Reformulation of the Attraction Force

Eq. (4) concentrates the attractive energies of the potential field Π in singular points P, and defines F_1 as a convex weighted sum over all points p_j with corresponding weights α_j. As $\mathcal{M}$ now continuously distributes these energies according to its density function (7), we define a corresponding *continuous* force $\mathcal{F}_1$ by the convex sum over the integral attraction of each single Gaussian, with convex weights w_s accounting for the Gaussians' relative point mass:

$$\mathcal{F}_1(q,\mathcal{M}) = \sum_s w_s \int_{\mathbb{R}^3} \frac{\mathbf{x}\, g(\mathbf{x}|\Theta_s)\alpha(\mathbf{x})}{\sum_{s'} w_{s'} \int_{\mathbb{R}^3} g(\mathbf{x}'|\Theta_{s'})\alpha(\mathbf{x}')\,\mathrm{d}\mathbf{x}'}\,\mathrm{d}\mathbf{x} \tag{13}$$

where similar to Eq. (6), we define the weight $\alpha(\mathbf{x}) = \theta(\delta)/\delta$, with $\delta = \|\mathbf{x} - q\|$. Additionally, each point $\mathbf{x} \in \mathbb{R}^3$ is now weighted by the Gaussian density g of the corresponding component Θ_s. As before, the integral over all weighted contributions is normalized by the integral over all weights. Figure 4(a) illustrates the spatial weights induced on the domain $\mathbb{R}^3$ by an anisotropic Gaussian Θ_s and a radial kernel α centered at a particle q. Multiplying all occurring weights into a combined weight function

$$\Omega_s(\mathbf{x}) = w_s g(\mathbf{x}|\Theta_s)\alpha(\mathbf{x}) \tag{14}$$

defines the attraction step $\mathcal{F}_1$ as

$$\mathcal{F}_1(q,\mathcal{M}) = \frac{\sum_s \int_{\mathbb{R}^3} \mathbf{x}\,\Omega_s(\mathbf{x})\,\mathrm{d}\mathbf{x}}{\sum_s \int_{\mathbb{R}^3} \Omega_s(\mathbf{x})\,\mathrm{d}\mathbf{x}}. \tag{15}$$

Figure 4(b) shows the form of the combined weight function Ω_s for the generating Gaussian Θ_s and kernel α. Note that the integral in Eq. (15) is not finite due to a singularity at $\delta = 0$ produced by its factor δ^{-1}. However, in the basic Weiszfeld's algorithm as well as in LOP, a particle spatially coinciding with an input point (i.e., $\delta = 0$) represents a singularity anyway, which is typically accounted for by removing the point in question, or biasing the denominator δ^{-1} to some $(\epsilon + \delta)^{-1}$ to clamp the point's energy at $\delta = 0$ to a finite peak. Following the same reasoning, we circumvent the infinite integral by approximating the weight function α with a sum of K Gaussians

$$\hat{\alpha}(\mathbf{x}) = \sum_{k=1}^{K} \hat{\alpha}_k(\mathbf{x}) = \sum_{k=1}^{K} \hat{w}_k \hat{c}_k\, g(\mathbf{x}|q,\hat{\Sigma}_k) \tag{16}$$

(Figure 4(c)), which provides an integrable finite peak at $\delta = 0$ while still exhibiting the characteristic weight falloff of α. Eq. (16) gives the general d-dimensional formulation for $\hat{\alpha}_k$, where $\hat{\Sigma}_k = \hat{\sigma}_k^2 h^2 I$ denotes its covariance, $\hat{w}_k$ the (dimension-invariant) weight, and $\hat{c}_k = |2\pi\hat{\Sigma}_k|^{\frac{1}{2}}$ compensates for the dimension-dependent normalization factor of the pdf g. Since the Gaussian kernel is normalized by the LOP support radius h, the coefficients $\hat{w}_k$ and $\hat{\sigma}_k$ of the model function (16) can be fitted by setting $h = 1$ and considering only the normalized range $\delta \in [0,1]$. In our experiments, we observed that a sum of $K = 3$ Gaussians provides a sufficient approximation (see Figure 5(a)). The coefficients obtained using Levenberg-Marquardt optimization are listed in Figure 5(b). Replacing α by $\hat{\alpha}$ gives an estimator $\hat{\Omega}$ approximating the combined weight function (14) by

$$\hat{\Omega}_s(\mathbf{x}) = w_s g(\mathbf{x}|\Theta_s) \sum_{k=1}^{K} \hat{\alpha}_k(\mathbf{x}) = \sum_{k=1}^{K} \hat{\Omega}_{sk}(\mathbf{x}). \tag{17}$$

This comes with a convenient property: Since the product of two Gaussians is again a Gaussian, Eq. (17) reduces the complete weight function $\hat{\Omega}_s$ to a sum of K product Gaussians $\hat{\Omega}_{sk}$, which we can again interpret as weighted Gaussian pdfs, with weights ω_{sk} and means μ_{sk}. Therefore, Equation (15) can now be expressed in **closed form** by

$$\mathcal{F}_1(q,\mathcal{M}) = \frac{\sum_s \sum_k \int_{\mathbb{R}^3} \mathbf{x}\,\hat{\Omega}_{sk}(\mathbf{x})\,\mathrm{d}\mathbf{x}}{\sum_s \sum_k \int_{\mathbb{R}^3} \hat{\Omega}_{sk}(\mathbf{x})\,\mathrm{d}\mathbf{x}} = \frac{\sum_{s,k} \omega_{sk}\mu_{sk}}{\sum_{s,k} \omega_{sk}}, \tag{18}$$

which in the same way that Eq. (4) is a convex sum of 3D points p_j, now becomes a convex combination of the product Gaussians' means μ_{sk} with weights ω_{sk}. By applying the identities for the integral and the expectation of a Gaussian product [Petersen and Pedersen 2012], we derive these quantities as follows.

Weight ω_{sk} Using Eq. (17), we obtain ω_{sk} for the general d-dimensional case as

$$\begin{aligned}\omega_{sk} &= \int_{\mathbb{R}^d} \hat{\Omega}_{sk}(\mathbf{x})\mathrm{d}\mathbf{x} = w_s\hat{w}_k\hat{c}_k \int_{\mathbb{R}^d} g(\mathbf{x}|\Theta_s)\; g(\mathbf{x}|\hat{\Theta}_k)\mathrm{d}\mathbf{x} \\ &= w_s\hat{w}_k\hat{c}_k\; g(\mu_s|q,\Lambda_{sk}) \\ &= w_s\hat{w}_k\hat{\sigma}_k^d h^d |\Lambda_{sk}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mu_s-q)^T\Lambda_{sk}^{-1}(\mu_s-q)} \end{aligned} \tag{19}$$

where we have introduced the covariance sum $\Lambda_{sk} = \Sigma_s + \hat{\Sigma}_k$.

Mean μ_{sk} Evaluating the weighted mean in the numerator of Eq. (18) gives

$$\begin{aligned}\omega_{sk}\mu_{sk} &= \int_{\mathbb{R}^d} \mathbf{x}\,\hat{\Omega}_{sk}(\mathbf{x})\mathrm{d}\mathbf{x} = w_s\hat{w}_k\hat{c}_k \int_{\mathbb{R}^d} \mathbf{x}\; g(\mathbf{x}|\Theta_s)\; g(\mathbf{x}|\hat{\Theta}_k)\mathrm{d}\mathbf{x} \\ &= w_s\hat{w}_k\hat{c}_k g(\mu_s|q,\Lambda_{sk})(\Sigma_s^{-1}+\hat{\Sigma}_k^{-1})^{-1}(\Sigma_s^{-1}\mu_s + \hat{\Sigma}_k^{-1}q) \\ &= \omega_{sk}(\Sigma_s^{-1}+\hat{\Sigma}_k^{-1})^{-1}(\Sigma_s^{-1}\mu_s + \hat{\Sigma}_k^{-1}q). \end{aligned} \tag{20}$$

Due to the expensive inversions in this formulation, we centralize the coordinate frame in q to further simplify the mean

$$\begin{aligned}\mu_{sk} &= (\Sigma_s^{-1}+\hat{\Sigma}_k^{-1})^{-1}(\Sigma_s^{-1}(\mu_s-q) + \hat{\Sigma}_k^{-1}(q-q)) + q \\ &= \hat{\Sigma}_k(\Sigma_s+\hat{\Sigma}_k)^{-1}(\mu_s-q)+q \\ &= \hat{\sigma}_k^2 h^2 \Lambda_{sk}^{-1}(\mu_s-q)+q. \end{aligned} \tag{21}$$

This way, the evaluation of both quantities requires only one matrix inversion of Λ_{sk}, which already produces the term $|\Lambda_{sk}|^{-1}$ required in ω_{sk} as side product. The final complete continuous attraction step is thus given by

$$\mathcal{F}_1(q,\mathcal{M}) = q + \sum_{s,k} \hat{\sigma}_k^2 h^2 \Lambda_{sk}^{-1}(\mu_s - q)\frac{\omega_{sk}}{\sum_{s',k'}\omega_{s'k'}}. \tag{22}$$

6.2 Initial Iteration

As shown in Eq. (2), LOP initializes its particle positions with the weighted mean of the input points using the weight kernel θ. Its continuous variant

$$\mathcal{F}_1^{(1)}(q,\mathcal{M}) = \frac{\sum_s \int_{\mathbb{R}^3} \mathbf{x}\, w_s g(\mathbf{x}|\Theta_s)\theta(\delta)\,\mathrm{d}\mathbf{x}}{\sum_s \int_{\mathbb{R}^3} w_s g(\mathbf{x}|\Theta_s)\theta(\delta)\,\mathrm{d}\mathbf{x}} = \frac{\sum_s \omega_s^{(0)}\mu_s^{(0)}}{\sum_s \omega_s^{(0)}} \tag{23}$$

is similar to Eq. (13), except that it omits the term δ^{-1} and can thus be evaluated only by the weight θ instead of K summands $\hat{\alpha}_k$. Expressing $\theta(\delta)$ by a scaled Gaussian pdf $c_\theta g(\mathbf{x}|q,\Sigma_\theta)$, with $\Sigma_\theta = (h^2/32)I$ and $c_\theta = |2\pi\Sigma_\theta|^{-\frac{1}{2}}$, gives the quantities for the initial weight and mean as

$$\omega_s^{(0)} = w_s c_\theta g(\mu_s|q,\Lambda_{s\theta}), \quad \mu_s^{(0)} = \tfrac{1}{32}h^2\Lambda_{s\theta}^{-1}(\mu_s-q)+q, \tag{24}$$

with covariance sum $\Lambda_{s\theta} = \Sigma_s + \Sigma_\theta$.

7 Weighted CLOP

As the original LOP operator is very sensitive to regions of varying point densities, Huang et al. [2009] proposed a weighted LOP operator (WLOP), which normalizes the attractive force (4) over differently dense regions by adding for each point p_j a density-dependent weight $v_j = 1 + \sum_{j'\in J\setminus\{j\}} \theta(\|p_j - p_{j'}\|)$, so that

$$F_1(q,P) = \sum_{j\in J} p_j \frac{\alpha_j/v_j}{\sum_{j'\in J}\alpha_{j'}/v_{j'}}. \tag{25}$$

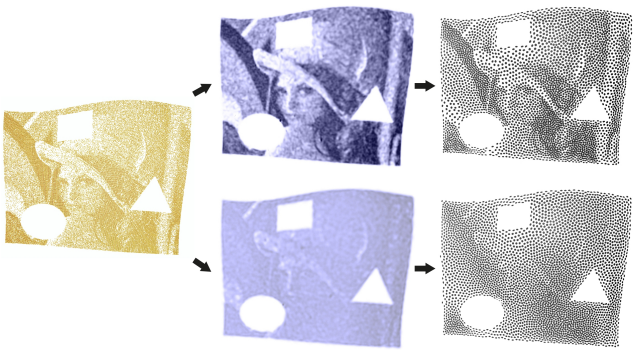


Figure 6: *Point sampling of Lena with point density inverse proportional to image intensity (74K points, left), its corresponding mixture $\mathcal{M}^{(4)}$ (5K Gaussians, middle) and CLOP resampling (3700 particles, right). The top row shows the unweighted mixture, resulting in an unevenly distributed resampling resembling Lena's portrait, while the bottom mixture shows the desired, balanced particle distribution built on the initially weighted Gaussians.*

This additional weighting can be easily adopted in CLOP, without even changing its integral formulations in Section 6. Since a Gaussian's attractive potential is defined by its weight w_s, we can encode the balancing weights v_j directly in the Gaussians representing the p_j in the initial mixture $\mathcal{M}^{(0)}$, by altering their initial weights to

$$w_j^{(0)} = (v_j|P|)^{-1}. \tag{26}$$

This way, the attraction-determining weights w_j of Gaussians that cluster points in dense regions (large v_j) will be relaxed more strongly than weights in regions of lower density. Applying CLOP to such a *weighted* mixture thus results in a continuous equivalent of the weighted attraction in WLOP. As we can directly accumulate the sum (25) along with the points' covariances in the initial kernel pass (Section 5.3), this weighting can be achieved in CLOP without any additional effort. Figure 6 recreates the Lena demo from Huang et al. [2009], demonstrating the improved performance of CLOP when using such weighted mixtures.

8 Accelerating Repulsion

The reformulation and continuous evaluation of F_1 shown in the previous sections accelerates the major part of the computational load in a LOP iteration (1). As a result, when using a larger number of particles, the discrete computation of the repulsion forces becomes the bottleneck. In this section, we address this problem by two approaches.

Kernel Cutoff. A simple way to accelerate repulsion is to skip the evaluation of F_2 (5) between particles where the relative repulsive influence is very low. For higher particle counts, this applies to all particles q with distance $\gtrsim h/2$ to a repulsing particle q', due to the Gaussian kernel falloff weighting this repulsion exponentially lower than those from particles closer to q. We have observed that simply cutting off the repulsion kernel at about half its radius reduces the repulsion computation effort by $\sim 75\%$, while having a negligible effect on the regularity of the final particle distribution. Note, however, that such a cutoff is not applicable to the attraction force, as there it is crucial for the kernel to bridge the gap between an outlier and the surface it should be projected to.

Repulsion Coherence. Another optimization exploits a coherence in the particles' repulsive moments $F_2(q,Q')$ in Eq. (5), which

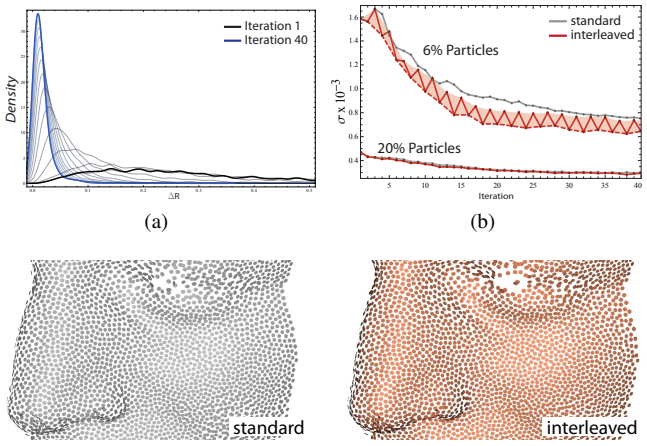


Figure 7: *(a) Development of the distribution of $\Delta\dot{R}$ for the Face model, from first (black) to the last CLOP iteration (blue). (b) Convergence of nearest-neighbor variances σ for unoptimized (gray) and interleaved repulsion (red). The closeup visually compares the resulting particle distributions.*

we will here denote as $\dot{R}$. We have observed that although the moment $\dot{R}$ of each individual particle q is highly dynamic in both direction and magnitude, the relative change in the overall system is generally low. To measure a particle's coherence of $\dot{R}$ between two iterations, we examine its relative change of magnitude $\Delta\dot{R}^{(k)} = \frac{\|\dot{R}^{(k)} - \dot{R}^{(k-1)}\|}{\|\dot{R}^{(k-1)}\|}$, which can also be thought of as a scale-invariant error measure when using $\dot{R}^{(k-1)}$ to approximate $\dot{R}^{(k)}$. Figure 7(a) shows the distribution of $\Delta\dot{R}$ for the face data set as it develops over different numbers of CLOP iterations. The graphs show that the overall error $\Delta\dot{R}$ is bounded and progressively reduces as CLOP converges. After the first iteration, the repulsive moment of most particles do not deviate more than $\sim 40\%$, and with successive iterations, hardly more than 10%. This observation suggests that, under acceptance of the discussed error, a given $\dot{R}^{(k)}$ can be used as an estimator for $\dot{R}^{(k+1)}$ in the next iteration. Therefore, we are able to reduce the computation effort by another 50% alone by reusing the repulsion vectors every second frame. We propose to still perform an actual repulsion computation in the final iteration in case of an odd number of iterations. Figure 7(b) plots the variance σ of nearest neighbor distances, measuring the regularity of the point distribution [Huang et al. 2009], for both unoptimized and interleaved repulsion and different amounts of resampling particles. We observe that with a larger relative number of particles (20%), an interleaved repulsion update hardly affects the convergence behavior of σ. On the other hand, a lower number of particles (6%) allows them to move more freely, leading to an oscillation of σ when correcting the repulsive moment only each second iteration. However, the band in which it oscillates generally appears to drop faster, which shows that in addition to a performance improvement, interleaved repulsion actually leads to a potentially faster convergence in point regularity.

9 Robust Normal Computation in Mixtures

Having derived a robust projection operator for spatial data to accelerate the L_1 point reconstruction, we are interested in a similar speedup for locally robust normal reconstruction methods [Oztireli et al. 2009; Zheng et al. 2010]. In this section we show that our derivation of the continuous attraction in Section 6 can be directly applied to the domain of unit normals to quickly compute L_1-aligned, unoriented normals for the particles obtained by CLOP.

9.1 Spherical Weiszfeld for Normal Axes

The basic idea of our local L_1-based normal alignment is to find the robust median within a set of unoriented normals *(normal axes)* $\mathbf{m}_j^* \in \mathbb{S}^2$ of spatially neighboring points $\mathbf{p}_j$, which can be roughly estimated using standard PCA. Similar to how Weiszfeld's algorithm iteratively approximates the spatial median of noisy points in $\mathbb{R}^3$, we can use a spherical equivalent to find a *spherical median* $\mathbf{n}_{opt} = \underset{\mathbf{n}\in\mathbb{S}^2}{\operatorname{argmin}}\{\sum_{j\in J} d_g(\mathbf{m}_j, \mathbf{n})\}$ of these noisy estimated point normal axes, which minimizes the sum of geodesic distances $d_g(\mathbf{m}_j, \mathbf{n}) = cos^{-1}\langle \mathbf{m}_j, \mathbf{n}\rangle_{max}$ [Banerjee et al. 2005]. Here, $\mathbf{m}_j$ represents the unit vector parallel to the (bipolar) axis $\mathbf{m}_j^*$ that minimizes the geodesic distance $d_g(\mathbf{m}_j, \mathbf{n})$. Based on the above definition, we can define a spherical Weiszfeld iteration that moves an initial estimator of a particle normal $\mathbf{n}$ towards the median of neighboring point normal axes $\mathbf{m}_j^*$ by

$$\mathbf{n}^{(k+1)} = \frac{\sum_{j\in J} \mathbf{m}_j \alpha_j}{\sum_{j\in J} \alpha_j}, \quad \alpha_j = \frac{\theta(\|p_j - q\|)}{d_g(\mathbf{m}_j, \hat{\mathbf{n}}^{(k)})} \tag{27}$$

where $\hat{\mathbf{n}}^{(k)}$ denotes the normalized result from the previous iteration, and θ localizes the median projection to point neighbors within a compact range as before. Projecting the normal of each particle into the median of a set of point normals $\mathbf{m}_j$ produces the same computational effort as the LOP attraction term in Eq. (4). We will therefore now introduce a fast continuous variant of the spherical Weiszfeld algorithm, which corresponds to CLOP and operates on a spherical mixture distribution of the unoriented point normals $\mathbf{m}_j^*$.

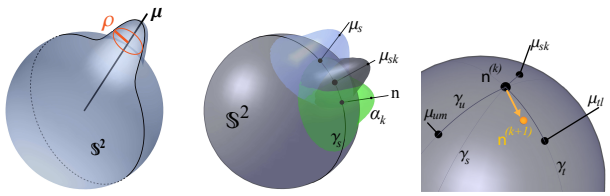
9.2 Spherical Mixture Distribution

Similar to how HEM reduces the input points to a mixture of continuous spatial distributions, we reduce the set of estimated point normals $\mathbf{m}_j$ to a set of *wrapped normal* (WN) distributions, which can be thought of as normal distributions endlessly wrapped around the unit circle [Mardia and Jupp 2009]. They give an approximate description of the Mises-Fisher (vMF) distribution, which is a well established and exact model for a random variate on $\mathbb{S}^2$. Nevertheless, considering a hierarchical clustering and doing calculus to obtain a continuous formulation of a spherical Weiszfeld iteration, vMFs are hard to handle directly. However, we can sufficiently simulate a vMF by rotating a one-dimensional wrapped normal distribution $\Phi = (\mu, \rho)$ around its mean μ on the unit sphere (Figure 8(a)). The concentration parameter ρ defines the dispersion of the distribution and measures the *mean resultant length* $\rho = \|\sum_j \mathbf{m}_j\|/n$ of a set of n unit vectors $\mathbf{m}_j$, increasing in value as the dispersion decreases. ρ also relates to the variance of the standard normal distribution by $\sigma^2 = -2\ log\ \rho$. (In the following, we will use both ρ and σ^2 in our derivations). The pdf of Φ is given by

$$g_w(x|\Phi) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} e^{-\frac{1}{2}(\frac{x-\mu+2\pi k}{\sigma})^2}. \tag{28}$$

For a reasonably concentrated distribution (variance bounded by 2π), the sum representing the infinite wrapping of the distribution can be sufficiently approximated by the term $k = 0$ ([Mardia and Jupp 2009] p. 50), which gives a standard normal distribution. Note that in Eq. (28), μ and x represent angles on a great circle.

An ordinary HEM-like maximum-likelihood estimate of the set of Φ_s on the spherical domain [Banerjee et al. 2005] is not sufficient for our needs, as it neglects the association of normals $\mathbf{m}_j$ to



(a) (b) (c)

Figure 8: *(a) Univariate WN distribution on the $\mathbb{S}^2$, creating a spherical normal distribution by rotation. (b) Since both α_k and μ_s are isotropic, their product μ_{sk} lies on their common geodesic γ_s. (c) Weiszfeld step for $\mathbf{n}_j$ defined by the weighted mean of all μ_{sk}.*

points $\mathbf{p}_j$ required for the spatial localization kernel θ in Eq. (27). Therefore, instead of computing a spherical mixture independent from the Gaussian point distributions $\mathcal{M}$, we assign a distinct WN Φ_s to each Gaussian $\Theta_s \in \mathcal{M}$, and cluster them along with the Gaussians during HEM in Section 5, i.e., the normals do not influence the computed responsibilities which determine the clustering. This leads to a simple extension of the HEM clustering algorithm:

1. For each point p_j, extend the initial mixture $\mathcal{M}^{(0)}$ by an initial WN distribution $\Phi_j^{(0)} = (\mathbf{m}_j, \rho^{(0)})$, with $\rho^{(0)} = 1$
2. In the M-Step, update the MLE of Φ_s using the same spatial weights r_{is} and $\bar{w}_s$ that cluster Θ_s.

We define the MLE updates for a next level WN $\Phi_s^{(l+1)}$ by

$$\mu_s^{(l+1)} = \frac{\sum_i \mu_i^{(l)}\, r_{is}\overline{w}_i}{\|\sum_i \mu_i^{(l)}\, r_{is}\overline{w}_i\|} \tag{29}$$

$$[\sigma_s^2]^{(l+1)} = \frac{\sum_i \sigma_i^2\, r_{is}\overline{w}_i}{\sum_i r_{is}\overline{w}_i} - 2\, log\left(\left\|\frac{\sum_i \mu_i^{(l)}\, r_{is}\overline{w}_i}{\sum_i r_{is}\overline{w}_i}\right\|\right), \tag{30}$$

which is the spherically wrapped isotropic equivalent to the clustering of Gaussians in Eq. (10). Since the *log*-argument in Eq. (30) is the mean resultant length ρ of WN means $\mu_i^{(l)}$, the complete second term gives the variance of these $\mu_i^{(l)}$. Thus, according to Eq. (10), Eq. (30) defines the MLE of the variance $[\sigma_s^2]^{(l+1)}$ by the weighted sum of level-l variances and the variance of the level-l means.

9.3 Continuous Spherical Weiszfeld in Mixtures

We will now show that the results for the continuous attraction $\mathcal{F}_1$ in Section 6 can be directly applied to formulate a continuous spherical Weiszfeld (CSW) step. CLOP defines the target position of an iteration step by a convex combination of expectations $E[\hat{\Omega}_{sk}]$, where $\hat{\Omega}_{sk}$ are Gaussian weights defined in Eq. (17). On the unit sphere, those weights are now accordingly defined by

$$\hat{\Omega}_{sk} = w_s\, g_w(x|\Phi_s)\, \hat{\alpha}_k(x). \tag{31}$$

where w_s are the mixture coefficients of $\mathcal{M}$ as before, g_w is the (one-dimensional) pdf of the WN distribution, and $\hat{\alpha}_k(x)$ is defined as in Eq. (16). The sought quantities μ_{sk} and ω_{sk} for computing $E[\hat{\Omega}_{sk}]$ can be obtained by wrapping the Euclidean arrangement of the involved weights (Figure 4) onto the unit sphere (Figure 8). Although we operate on the 2-dimensional domain $\mathbb{S}^2$, it is sufficient to evaluate these expectations only along 1-dimensional geodesics, on which the wrapped normal distributions Φ_s are defined. Due to the isotropic symmetry of the weighted Gaussian components g_w and $\hat{\alpha}_k$ in Eq. (31), the sought mean μ_{sk} always lies on the geodesic γ_s through the particle normal $\mathbf{n}$ and the mean normal μ_s of the s-th mixture component (Figure 8(b)). This allows us to evaluate μ_{sk} and ω_{sk} using an angular parametrization on γ_s. This way, the derived formulations for the mean and weight in Section 6 can be directly applied to the 2-dimensional unit sphere setting $d = 2$:

Weight ω_{sk} The covariance sum Λ_{sk} defined in Eq. (19) now simplifies to an isotropic bivariate matrix $\Lambda_{sk} = \lambda_{sk} I_2$ with diagonal entries $\lambda_{sk} = \sigma^2 + \hat{\sigma}_k^2 h^2$, and ω_{sk} becomes

$$\omega_{sk} = w_s \hat{w}_k \hat{\sigma}_k^2 h^2 \lambda_{sk}^{-1} e^{-\frac{1}{2}\frac{d_g(\mu_s,\mathbf{n})^2}{\lambda_{sk}}}. \tag{32}$$

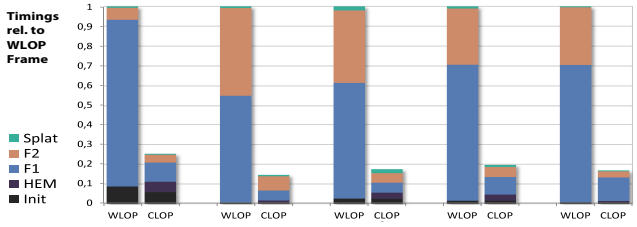
Mean μ_{sk} Similar to the Euclidean case (21), we centralize the angular parametrization in the particle normal $\mathbf{n}$. Then the relative mean of the product function $\hat{\Omega}_{sk}$ on γ_s is defined by

$$d_g(\mu_{sk}, \mathbf{n}) = \hat{\sigma}_k^2 h^2 \lambda_{sk}^{-1} d_g(\mu_s, \mathbf{n}). \tag{33}$$

Here we can use the same coefficients $\hat{w}_k$ and $\hat{\sigma}_k$ as in CLOP (Figure 5(b)). Since it is not necessary to localize the median seeking of $\mathbf{n}$ on the unit sphere like LOP does in Euclidean space, h can be safely relaxed to a conservative radius $h = \pi$. The actual mean μ_{sk} can now be obtained by interpolating between μ_s and $\mathbf{n}$ on γ_s, i.e., $\mu_{sk} = \mu_s t + \mathbf{n}(1-t)$, where Eq. (33) gives the interpolation factor

$$t = \frac{d_g(\mu_{sk}, \mathbf{n})}{d_g(\mu_s, \mathbf{n})} = \frac{\hat{\sigma}_k^2 h^2}{\sigma^2 + \hat{\sigma}_k^2 h^2}. \tag{34}$$

Finally and analogously to the Euclidean case (18), the Weiszfeld iteration step is given by the weighted sum of the resulting mean normals μ_{sk} (Figure 8(c)), with weights ω_{sk} defined as above.



Model	Lena		Face		Camel		Garg. sm.		Gargoyle	
$\|P\|$	74K (74K)		84K (84K)		87K (87K)		77K (78K)		175K (302K)	
$\|\mathcal{M}\|$	5100		8100		7850		10K		32K	
$\|Q\|$	3700		84K		72K		38K		107K	
Iters	50		16		10		20		10	
ms	WLOP	CLOP	WLOP	CLOP	WLOP	CLOP	WLOP	CLOP	WLOP	CLOP
Init	17	11	9	9	17	17	10	10	9	9
HEM		10		18		20		18		11
F_1	161	19	837	77	379	33	423	54	981	167
F_2	11	7	677	110	234	30	172	31	405	42
Total	189	47	1523	214	630	100	605	113	1395	229
SU F_1	5.55		8.81		7.15		5.88		5.51	
SU F_2	1.57		6.15		7.80		5.55		9.64	
SU Tot.	**4.02**		**7.12**		**6.30**		**5.35**		**6.09**	

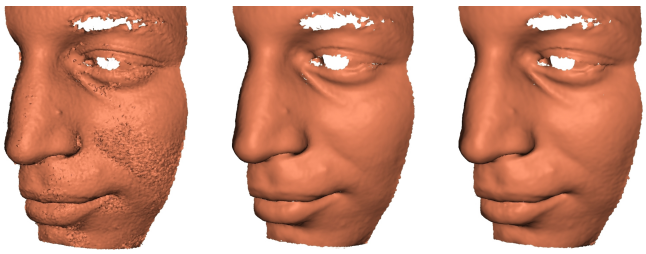
Figure 9: *Model statistics and individual timings in ms. Speedups (SU) are given for attraction (F_1) and repulsion (F_2) separately, as well as for the whole CLOP operator compared to a corresponding WLOP GPU implementation. The top graphs give individual timings of each phase normalized by WLOP total time.*

10 Evaluation and Results

10.1 Performance

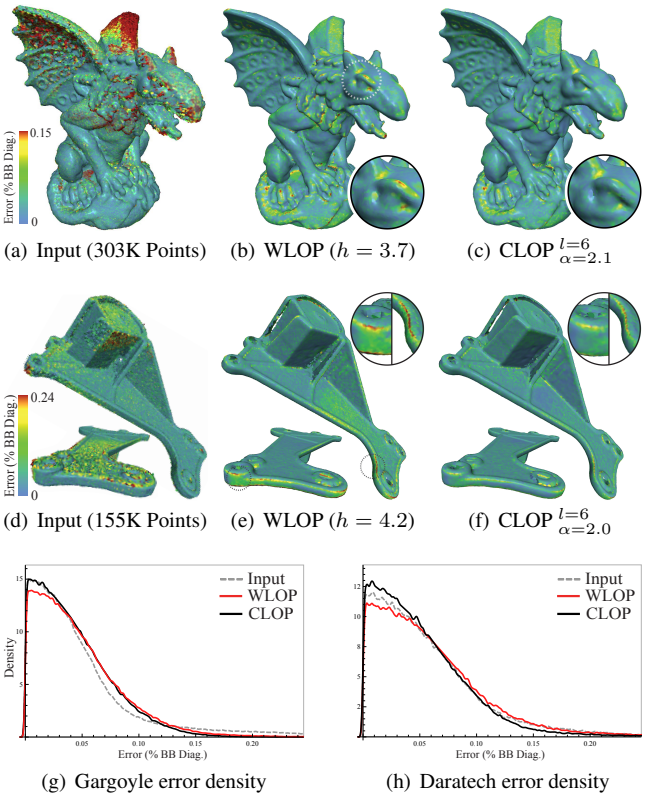
To achieve interactive reconstruction performance, we have used a rasterization-based GPU implementation that allows for fast grid-based neighbor queries [Preiner et al. 2012]. All local kernel operations in the main stages of the algorithm are executed by quad rasterization on the projected images of the input point set P, the Gaussian component locations of $\mathcal{M}$, and the particle set Q, which are all stored in individual A-buffers. The Gaussian mixture computation, CLOP iterations, optional consecutive CSW (normal estimation) and final rendering are performed in each individual frame based on the input points P projected into this A-buffer. Note that since particle positions change over the CLOP iterations, possible A-buffer overflows commonly slightly reduce the number of total particles that finally remain for rendering. The particle counts $|Q|$ listed in Figure 9 therefore always denote the average amount over all iterations. To be able to fully assess the performance of our system, we do not currently exploit any frame-to-frame coherence. However, common temporal-coherence approaches could accelerate our system even further [Liao et al. 2013]. Note that for normal estimation, we simply orient the normals towards the camera when rendering the reconstructed point cloud. Obtaining globally consistent normals would require global computations like orientation propagation [Huang et al. 2009], which are too expensive for an interactive setting.

All results were produced on a PC with an Intel i7 4470K 3.5 GHz CPU and NVIDIA GeForce GTX TITAN GPU. A framebuffer resolution of 1700×880 was used in all our performance tests and the results shown in the video. Fig. 9 summarizes statistics and performance measures for the 5 tested models (Lena, Face, Camel, Gargoyle and small Gargoyle) and plots the relative speedup of CLOP over WLOP. The given point set cardinalities $|P|$ denote the points left to operate on *after* A-buffer projection. The *original* number of input points is given in brackets. The performance numbers include individual timings for creating the mixture model (HEM) and evaluating the continuous attraction compared to the discrete attraction (F_1), as well as the accelerated repulsion compared to the full repulsion computation (F_2). The Camel model as shown in Figure 1 was generated with a virtual scanning framework [Berger et al. 2013], using 18 individual scans. The parameters have been set to generate a realistic but relatively high level of noise and outliers. Please refer to the supplemental material for details. All results were produced with weighting enabled (WLOP vs. weighted CLOP) since it doesn't incur additional costs on either side. We observe an overall speedup of up to 7 times the WLOP performance, while producing a practically indistinguishable reconstruction. Fig. 10 gives a visual comparison for the noisy face model, where CLOP outperforms WLOP by a factor of 7. The results show that even when reconstructing the model with a large number of particles (about the same as the input model), only a low number of Gaussians ($\sim 10\%$) are required to represent the input point cloud, leading to significant speedups in the attraction evaluation (up to a factor of 9). See also the accompanying video for further results and comparisons.



(a) Input (b) WLOP (1523 ms) (c) CLOP (214 ms)

Figure 10: *(a) Small-kernel splat reconstruction on the Face model showing heavy registration errors. (b) After WLOP (16 iterations). (c) After CLOP with 10% Gaussian components.*



(a) Input (303K Points) (b) WLOP ($h = 3.7$) (c) CLOP $^{l=6}_{\alpha=2.1}$

(d) Input (155K Points) (e) WLOP ($h = 4.2$) (f) CLOP $^{l=6}_{\alpha=2.0}$

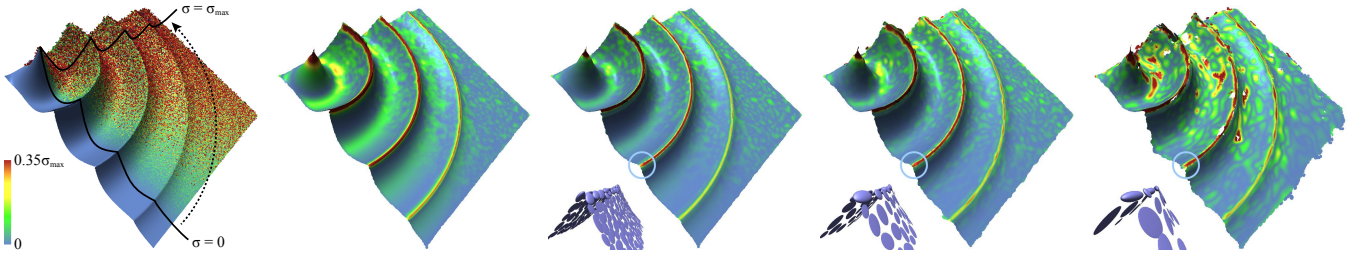
(g) Gargoyle error density (h) Daratech error density

Figure 11: *Reconstruction error on two noisy registered scans (a),(d). Heatmaps compare the error between (b),(e) the WLOP (#particles = #points, 20 iterations) and (c),(f) CLOP reconstruction. Kernel sizes h are given in % of the BB diagonal. For both cases, the detail lenses and the respective error distribution functions (g),(h) demonstrate the superior accuracy of our method.*

10.2 Reconstruction Quality

In this section, we analyze the reconstruction quality of our method in depth and evaluate its accuracy against the original WLOP algorithm. To allow for an accurate evaluation that is not biased due to the A-buffer based particle loss described above, we use an exact reference implementation for all accuracy measurements. We will show that although our approach runs several times faster than its discrete counterpart, its continuous nature is able to produce a reconstruction of comparable or even better accuracy.

Accuracy. We study the reconstruction error of CLOP vs. WLOP on two models exhibiting different characteristics (Fig. 11). To provide exact reference models for measurement, we used the virtual scanning framework by Berger et al. [2013]. Both models were resampled by 16 virtual scans exhibiting a moderate amount of additive Gaussian noise. These were registered using locally weighted ICP [Brown and Rusinkiewicz 2007] using a realistic amount of ro-



(a) Input ($|P| = 500K$) (b) WLOP (c) CLOP $_{\alpha=2}$, $|M| = 25K$ (d) CLOP $_{\alpha=3}$, $|M| = 4700$ (e) CLOP $_{\alpha=4}$, $|M| = 1200$

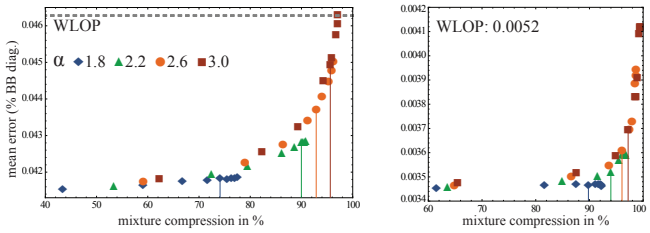
Figure 12: *(a) Point sampling of a radially symmetric ripple function, containing sharp edges of various angles, and exhibiting Gaussian noise increasing from 0 to σ_{max} with the angle of rotation. We compare the reconstruction error of WLOP (b) and (c)-(e) CLOP at various α and mixture reductions. Heat map colors relate to error amplitudes. Note that with a suggested $\alpha = 2$, we are able to achieve an overall lower approximation error than with WLOP.*

tational misalignment, which is a common source of outliers. Note that we generated a smaller version of the Gargoyle for the performance tests in Section 10.1 using the same parameters with lower scanner resolution. The left column in Figure 11 shows the resulting noisy input point clouds, the middle and right columns give the WLOP and CLOP reconstructions. Splat colors indicate particle errors $E(q) = \langle q - p, N_p \rangle$, measuring the distance of q to the tangent plane in the nearest reference surface point p, with N_p being its normal. Interestingly, the error heat maps and detail lenses indicate a generally superior behavior of our method over WLOP, especially at regions of high curvature. Only in few isolated regions like at the Gargoyle's ear, the clustering between very close-by misaligned scans leads to a slightly higher error than using WLOP. Fig. 11(g) and 11(h) plot the error density functions (particle error on abscissa vs. density on ordinate) of CLOP against WLOP and the input data. Both graphs show that CLOP produces more low-error particles and less high-error particles than WLOP, thus providing an overall better reconstruction quality.

A detailed error analysis for various cases is given in Figure 12. Here we use a synthetic data set (Fig. 12(a)) designed to show varying levels of noise as well as sharp and smooth features. Figure 12(b) shows the WLOP reconstruction after 20 iterations. As expected, the error is maximal at the sharp edges. The presence of noise in the data leads to a more noisy particle alignment, although only at a very subtle level. However, even in the region that has no noise in the input, WLOP produces homogeneous regions of error in the curved trenches of the function. This is because particles are repulsed in tangential direction instead of along the curvature of the surface, and thus is less visible as the surface gets more planar. Figure 12(c) shows the corresponding CLOP result at reasonable parameters. Compared to WLOP, we observe a clear reduction of the error regions in both the flat trenches and the sharp edges (especially apparent at the conic apex) of the function. We suspect that CLOP's overall better quality can be attributed to the continuous attractive energies, which provide a smoother and thus more robust description of the geometry than the singular attractive points used by WLOP. This might positively affect the stability of the attractive particle movements against the perturbing repulsion forces, and thus allow for a more controlled and overall better optimization.

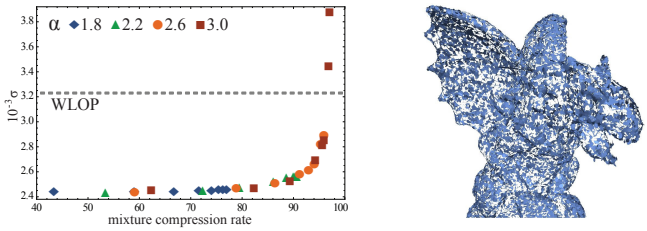
Effect of Clustering. As our method relies on a reduced representation of the input data, i.e., a Gaussian mixture, we are interested in the effect of this reduction on reconstruction quality, especially in the presence of high frequencies. We thus investigate the reconstruction error for different levels of compression. Figure 12 (c) to (e) show CLOP reconstructions (same kernel size and iteration count) for increasing mixture compression (increasing α at 8 HEM levels). While for $\alpha = 2$ we have observed an improved accuracy over WLOP, the quality drops with successive levels of compression. Figure 12(e) shows that choosing an extreme mixture compression still achieves moderate reconstruction quality in noise-free regions, but breaks down for stronger noise levels, leading to a corrupt reconstruction with large errors and irregularities in the particle distribution (holes). The insets depict the Gaussian mixture at a sharp edge (illustrated by 1-σ-isoellipsoids), and show that a sufficiently strict HEM regularization (i.e., small α) produces almost no signal blurring.

The reduction of the Gaussian mixture is controlled by the regularization α and the number of clustering levels. Figure 13 plots the actual compression rates (abscissa) for the Gargoyle and the Ripple data set against the mean reconstruction error (ordinate, in % of bounding box diagonal) for different values of α and HEM levels. The plots show that in general, the mean error lies clearly below the WLOP error level for reasonable compression rates, and only starts to increase for strong compressions, depending on the complexity of the model: For the Gargoyle, this happens at 80%, while the ripple model, containing more smooth and flat regions, can be compressed up to 94% without significantly increasing the reconstruction error. We also see that for a given compression rate, lower values for α require more HEM clustering levels to achieve the same compression, but for $\alpha \lesssim 2.2$, also bound the compression so that the region of rapidly increasing error is avoided. Lower α also produces a lower error at a given mixture size due to a stronger regularization. For these reasons we recommend to use an $\alpha \approx 2.0$.



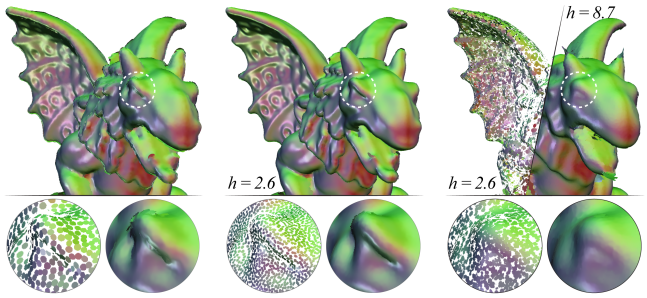
(a) Gargoyle (Fig. 11) (b) Ripple function (Fig. 12)

Figure 13: *Mean error development with increasing mixture compression, given by various values of α and $1-9$ levels. For each α, the marker lines indicate the level where the additional compression falls below 2.5%. The dashed line gives the corresponding WLOP error (above plot range in (b)).*



(a) (b)

Figure 14: *(a) Sampling regularity σ of the Gargoyle model with increasing levels of compression. (b) WLOP on a sub-sampled Gargoyle with same parameters and reduction rate as in Fig. 11(c).*



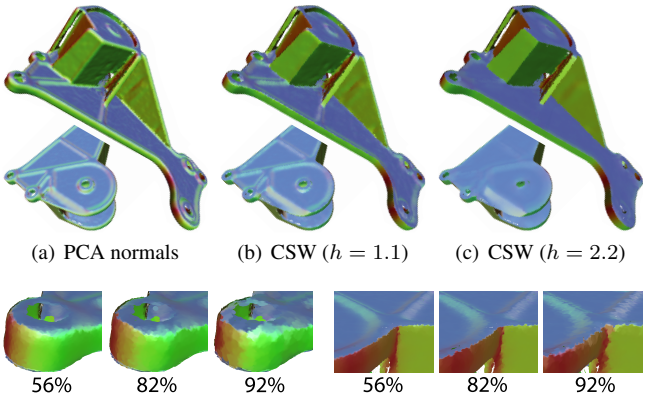
(a) Input ($39.5K$) (b) CLOP ($198K$) (c) WLOP ($198K$)

Figure 15: *Sparse Gargoyle (a) upsampled to 500% of the input point count, (b) using CLOP ($h = 2.6$), (c) using WLOP ($h = 2.6$ and 8.7). Lenses show splat distributions and surface details for the input, CLOP upsampling, and large-kernel WLOP upsampling. Kernel sizes h are given in % of the bounding box diagonal.*

Number of HEM Levels. For a given α, we want to use enough HEM levels to achieve sufficient compression, but also not waste performance on additional levels that do not substantially reduce the mixture further. Figure 13 shows that different input models show different compression potential. To take this into account, we abort clustering when the additional compression afforded by a level falls below a given threshold. The markers at the abscissas indicate the final mixture compression for a threshold of 2.5% additional compression.

Point Regularity. In all our experiments, the continuous formulation of the attractive energies has shown to provide an improved sampling regularity of the resulting particles, which we measure by the variance σ of nearest neighbor distances [Huang et al. 2009]. Figure 14(a) plots σ for different α and $1 - 8$ HEM levels for the Gargoyle from Figure 11(c). In contrast to the regularity achieved by WLOP (dashed line), the continuous attractive energies allow for smoother particle movements, resulting in a notably lower σ up to a critical point of compression ($\alpha = 3$), where the smooth energy distribution cannot be sufficiently described by the remaining Gaussians anymore. Note that this regularity improvement is independent of the one achieved by interleaved repulsion (Section 8), which we have activated for both the CLOP and WLOP evaluation.

Usable Amount of Particles. As shown by Lipman et al. [2007], the original LOP operator is problematic when using more particles than there are points in the model. If a given kernel contains too few attractive points to sufficiently describe a smooth energy density, particles tend to collapse into irregular clusters. A larger



(a) PCA normals (b) CSW ($h = 1.1$) (c) CSW ($h = 2.2$)

(d) Quality of normals with increasing mixture compression ($h = 1.1$)

Figure 16: *CSW on the Daratech model. (a) Splat normals using local PCA, (b) and (c) CSW normals with different kernel sizes. (d) Shows the quality reduction of the normals with increasing compression of the spherical mixture (given in % of input model size).*

kernel dampens this effect, but also comes with stronger smoothing. Therefore, the idea of accelerating WLOP by subsampling the input point set also reduces the usable amount of particles. Fig. 11(c) shows the CLOP reconstruction of the Gargoyle ($303K$ input points, $303K$ particles) using $35K$ *Gaussians*. Fig. 14(b) gives the WLOP result on the model after subsampling to *35K points*. Using the same particle count and kernel size as CLOP, WLOP fails to obtain a sufficient sampling regularity for a faithful reconstruction. In contrast, CLOP allows using much more particles than input points, and can therefore even be used to *upsample* a point cloud. E.g., based on a sparse sampling of the Gargoyle model with $39.5K$ points (Fig. 15(a)), we used CLOP to project $198K$ particles (500%) into its Gaussian mixture with $22K$ components, using a sufficiently small kernel bandwidth (Fig. 15(b)). A similar low-bandwidth upsampling using WLOP leads to particle collapses, while a sufficiently larger bandwidth destroys fine features (Fig. 15(c)).

Robust Normals. Finally, we investigate the effect of different CSW kernel sizes on the resulting alignment of the splat normals (Section 9.3). A common normal computation method is local tangent plane fitting using PCA [Hoppe et al. 1992], which is quick enough to be suitable for online reconstruction [Preiner et al. 2012], but suffers from typical smoothing artifacts. Figure 16(a) shows the CLOP result of the Daratech data set from Figure 11(f) with PCA normals (surface colors correspond to splat normals). Note the rounding of the model's sharp edges. Figure 16(b) illustrates the result after applying 13 iterations of the continuous spherical Weiszfeld steps on the CLOP result, where $|\mathcal{M}| = 17.8\%$ of the input point cloud. We used a manually tuned kernel of 1.1% of the BB diagonal, which optimally reconstructs the fine bracings at the model's front and backside. In contrast, a kernel of twice this size flattens subtle features, but also robustly aligns splats on more significant edges more faithfully (Figure 16(c)). Figure 16(d) shows details of the model for various levels of compression ($\alpha = 1.5, 2.0, 2.5, 5$ levels). As in spatial Continuous LOP, an increasing mixture compression leads to a successive reduction in quality. An optimal screen-space reconstruction of the Daratech model with $155K$ points ($135K$ after screen-space projection) requires 124 ms for CLOP + 56 ms for CSW (with kernel size $h = 1.1\%$ for both passes). Figure 17 shows a robust online normal alignment of a Kinect stream using CSW.

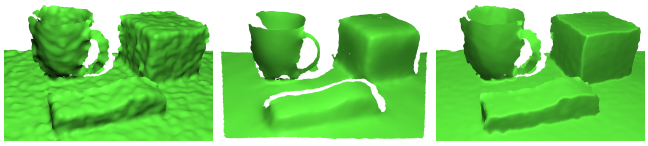


Figure 17: *Screen capture during online reconstruction of a Kinect stream (left). L_2-based splat reconstruction (middle) results in typical smoothing artifacts, while the proposed CSW normal reconstruction (right) faithfully reconstructs sharp features.*

11 Limitations

Like its discrete variants, CLOP is a kernel-based operator, and therefore inherits their kernel-related characteristics: larger bandwidths are required in cases of stronger noise, which then leads to stronger smoothing. Since CLOP exploits the fact that dense regions of low curvature can be compressed to very few Gaussians without loss of geometric information, the compression potential, and thus the achievable speedup, is always bound by the geometrical complexity of the input data (see Figure 13). As discussed in Section 5.3, the choice of the regularization term α is a trade-off between clustering efficiency and reconstruction quality. In order to maintain high accuracy, there is a bound on suitable values for the regularization term α, and thus also on the possible clustering speed. However, the resulting quality also depends on a proper mixture initialization bandwidth as mentioned in Section 5.3. The initializing kernel radius r has to be large enough (above noise level) to give the initial Gaussians proper orientation, but should be small enough to prevent from blurring the data in the first place. Currently, we use a density-adaptive radius r, which showed satisfactory results. However, in input data of strongly varying noise, a more flexible, locally noise-aware initialization could probably produce an even better reconstruction quality.

12 Conclusions and Future Work

We have introduced a novel surface reconstruction technique which applies the robust Weighted LOP operator to a continuous representation of a point set. We showed how to regularize the hierarchical EM algorithm to cluster the point set to a Gaussian mixture in a geometry-preserving manner, and derived an analytic formulation of LOP's attraction forces for this much more compact representation. This way, our method runs several time faster than the original discrete variant, while being able to produce comparable or even superior accuracy for reasonable regularization terms ($\alpha \approx 2$). Furthermore, CLOP provides an overall better sampling regularity, does not introduce constraints in the number of resampling particles (unlike subsampling approaches based on discrete LOP), and is even capable of point-cloud upsampling. We have also shown that our continuous formulation can be applied to the spherical domain for efficient robust feature-preserving normal estimation.

This paper has shown that a continuous reformulation of point-particle forces is highly versatile. In the future, we therefore want to generalize and apply it to a range of discrete algorithms.

Acknowledgements

The Face model was provided by Hui Huang. The Camel was sampled from a mesh animation provided by Robert Sumner and Jovan Popovic from the Computer Graphics Group at MIT. We thank the anonymous reviewers for their constructive comments. Thanks also to Matthew Berger for his support for the virtual scanning framework, and Thomas Auzinger and Philipp Schlegel for their valuable input. This work was supported by the EU FP7 project "Harvest4D" (no. 323567), and the EU FP7 People Programme (Marie Curie Actions) under REA Grant Agreement no. 290227.

References

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (Jan.), 3–15.

AMENTA, N., AND KIL, Y. J. 2004. Defining point-set surfaces. *ACM Trans. Graph. 23*, 3 (Aug.), 264–270.

AVRON, H., SHARF, A., GREIF, C., AND COHEN-OR, D. 2010. L1-sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. 29*, 5 (Nov.), 135:1–135:12.

BANERJEE, A., DHILLON, I. S., GHOSH, J., AND SRA, S. 2005. Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res. 6* (Dec.), 1345–1382.

BERGER, M., LEVINE, J. A., NONATO, L. G., TAUBIN, G., AND SILVA, C. T. 2013. A benchmark for surface reconstruction. *ACM Trans. Graph. 32*, 2 (Apr.), 20:1–20:17.

BOULCH, A., AND MARLET, R. 2012. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum 31*, 5, 1765–1774.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. *ACM Trans. Graph. 26*, 3 (July).

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of Conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '01, 67–76.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B 39*, 1, 1–38.

DEY, T. K., AND SUN, J. 2006. Normal and feature approximations from noisy point clouds. In *Proceedings of Conference on Foundations of Software Technology and Theoretical Computer Science*, vol. 4337 of *Lecture Notes in Computer Science*, 21–32.

FLEISHMAN, S., COHEN-OR, D., AND SILVA, C. T. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (July), 544–552.

GARCIA, V., NIELSEN, F., AND NOCK, R. 2010. Hierarchical Gaussian mixture model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

HARTLEY, R., AFTAB, K., AND TRUMPF, J. 2011. L1 rotation averaging using the weiszfeld algorithm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 3041–3048.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '92, 71–78.

HUANG, H., LI, D., ZHANG, H., ASCHER, U., AND COHEN-OR, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. 28*, 5 (Dec.), 176:1–176:7.

IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., AND FITZGIBBON, A. 2011. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of Symposium on User Interface Software and Technology*, ACM, UIST '11, 559–568.

JAKOB, W., REGG, C., AND JAROSZ, W. 2011. Progressive expectation–maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum 30*, 4 (June).

KALOGERAKIS, E., SIMARI, P., NOWROUZEZAHRAI, D., AND SINGH, K. 2007. Robust statistical estimation of curvature on discretized surfaces. In *Proceedings of the Eurographics Symposium on Geometry Processing*, SGP '07, 13–22.

KAZHDAN, M., AND HOPPE, H. 2013. Screened poisson surface reconstruction. *ACM Trans. Graph. 32*, 3 (July), 29:1–29:13.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, SGP '06, 61–70.

LI, B., SCHNABEL, R., KLEIN, R., CHENG, Z., DANG, G., AND SHIYAO, J. 2010. Robust normal estimation for point clouds with sharp features. *Computers & Graphics 34*, 2 (Apr.), 94–106.

LIAO, B., XIAO, C., JIN, L., AND FU, H. 2013. Efficient feature-preserving local projection operator for geometry reconstruction. *Computer-Aided Design 45*, 5, 861–874.

LIPMAN, Y., COHEN-OR, D., LEVIN, D., AND TAL-EZER, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. 26*, 3 (July).

MARDIA, K., AND JUPP, P. 2009. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley.

OZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum 28*, 2.

PETERSEN, K. B., AND PEDERSEN, M. S., 2012. The matrix cookbook, nov. Version 20121115.

PREINER, R., JESCHKE, S., AND WIMMER, M. 2012. Auto Splats: Dynamic point cloud visualization on the GPU. In *Proceedings of Eurographics Symposium on Parallel Graphics and Visualization*, 139–148.

VASCONCELOS, N. 1998. Learning mixture hierarchies. In *Proceedings of Advances in Neural Information Processing Systems*, MIT Press, NIPS'98, 606–612.

WALTER, B., BALA, K., KULKARNI, M., AND PINGALI, K. 2008. Fast agglomerative clustering for rendering. In *IEEE Symposium on Interactive Ray Tracing*, 81–86.

ZHENG, Q., SHARF, A., WAN, G., LI, Y., MITRA, N. J., COHEN-OR, D., AND CHEN, B. 2010. Non-local scan consolidation for 3d urban scenes. *ACM Trans. Graph. 29* (July), 94:1–94:9.

ZHOU, K., HOU, Q., WANG, R., AND GUO, B. 2008. Real-time kd-tree construction on graphics hardware. *ACM Trans. Graph. 27*, 5 (Dec.), 126:1–126:11.

Appendix: HEM Algorithm Outline

Algorithm 1 outlines the complete hierarchical EM procedure for computing a mixture $\mathcal{M}$ describing both the point distribution (by Gaussians Θ_s) and the spatially associated distribution of normals (by WN distributions Φ_s).

Algorithm 1: Outline of the HEM computation of mixture $\mathcal{M}$ including both spatial and spherical components.

input : point set $P = \{p_j\}_{j \in J}$, levels l_{max}, regularization α
output: mixture $\mathcal{M}^{(l_{max})} = \{w_s, \Theta_s, \Phi_s\}$

1 **foreach** $j \in J$ **do in parallel**
2 $\quad r \leftarrow \alpha_0 \cdot$ *NearestNeighborDist*(p_j); //Section 5.3
3 $\quad (\Sigma_j^{(0)}, v_j) \leftarrow$ *KernelCovDensity*(p_j, r); //Eq. (12)/Sec. 7
4 $\quad \mathbf{m}_j \leftarrow$ *MinEigenVec* $(\Sigma_j^{(0)})$;
5 $\quad w_j^{(0)} \leftarrow (v_j |P|)^{-1}$; //Eq. (26)
6 $\quad \Theta_j^{(0)} \leftarrow (\mathbf{p}_j, \Sigma_j^{(0)})$;
7 $\quad \Phi_j^{(0)} \leftarrow (\mathbf{m}_j, \rho^{(0)})$; //Section 9.2
8 **end**
9 $\mathcal{M}^{(0)} \leftarrow \{w_j^{(0)}, \Theta_j^{(0)}, \Phi_j^{(0)}\}$;
10 **for** $l \leftarrow 0$ **to** $l_{max} - 1$ **do**
11 $\quad S \leftarrow$ *RandomIndexSubset* $(\mathcal{M}^{(l)}, \pi)$;
12 $\quad$ **foreach** $s \in S$ **do in parallel**
13 $\quad\quad I \leftarrow \{i \mid D_{KL}(\Theta_i^{(l)} || \Theta_s^{(l+1)}) < \alpha^2/2\}$; //Eq. (11)
14 $\quad\quad \{r_{is}\}_{i \in I} \leftarrow$ *Responsib* $(\Theta_s^{(l+1)}, \Theta_i^{(l)})$; //Eq. (8)+(9)
15 $\quad\quad (w_s, \Theta_s, \Phi_s)^{(l+1)} \leftarrow$ *UpdateMLE* $(\mathcal{M}^{(l)}, \{r_{is}\}_{i \in I})$;
16 $\quad\quad$ //Eq. (10),(29),(30)
17 $\quad$ **end**
18 $\quad \mathcal{M}^{(l+1)} \leftarrow \{w_s^{(l+1)}, \Theta_s^{(l+1)}, \Phi_s^{(l+1)}\} \cup$ *Orphans*$(\mathcal{M}^{(l)})$;
19 **end**

For each point in parallel (Line 1), we first determine a conservative radius r in Line 2 ($\alpha_0 = 2 \sim 3$). Using this radius, we perform a kernel-accumulation pass (Line 3), computing both the point's initial covariance $\Sigma_j^{(0)}$ and its local density weight v_j. The initial estimator for its surface normal $\mathbf{m}_j$ is then extracted using the smallest eigenvector of $\Sigma_j^{(0)}$ (Line 4). These quantities define the initial weight $w_j^{(0)}$, Gaussian $\Theta_j^{(0)}$ and WN $\Phi_j^{(0)}$ assigned to the j-th point (Lines 5-7), which make up the initial mixture $\mathcal{M}^{(0)}$ (Line 9). Each iteration of the following hierarchical clustering loop (Line 10) represents an EM step fitting a reduced set of *parent* components $\{\Theta|\Phi\}_s^{(l+1)}$ for the next level to the current set of *child* components $\{\Theta|\Phi\}_i^{(l)}$. Here, the i-th WN component Φ_i is always coupled to the i-th spatial component Θ_i when clustering. The parent set is initialized by randomly selecting elements from the current mixture $\mathcal{M}^{(l)}$ (Line 11), where we recommend a selection probability $\pi \approx 1/3$. For each such parent (Line 12) in parallel, Line 13 applies the regularization constraint α (Section 5.3) to select the index set I of child components it is *responsible* for and thus allowed to merge. Note that the centers $\mu_i^{(l)}$ of all child components can be found within a conservative ball with radius $\alpha \cdot \sigma_{max}$ around $\mu_s^{(l+1)}$, σ_{max} being the root of the largest eigenvalue of $\Sigma_s^{(l+1)}$. Line 14 computes the responsibilities r_{is} of the s-th component for its selected set of children, which are then used to update the maximum-likelihood estimate of the parent's parameters (Line 15). Finally, the function *Orphans* in Line 18 selects child components that are not within the responsibility set I of any parent. They have to be taken along to the next level's mixture $\mathcal{M}^{(l+1)}$ together with the updated parent components in S.