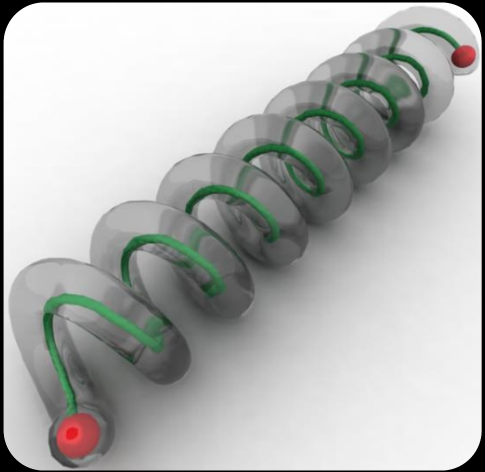
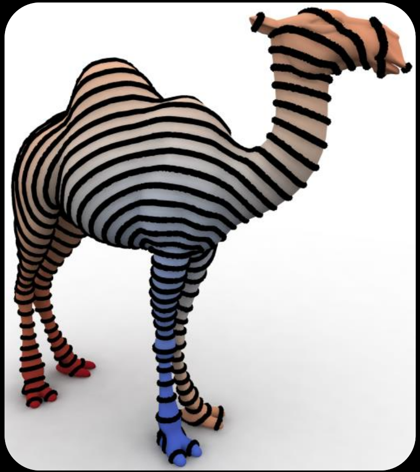
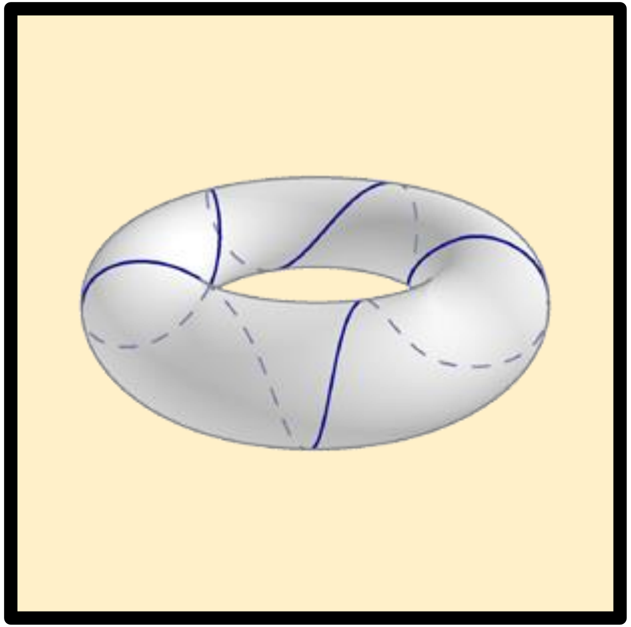
  

Earth Movers Distances on Discrete Surfaces

Justin Solomon, Raif Rustamov, Leonidas Guibas Stanford University

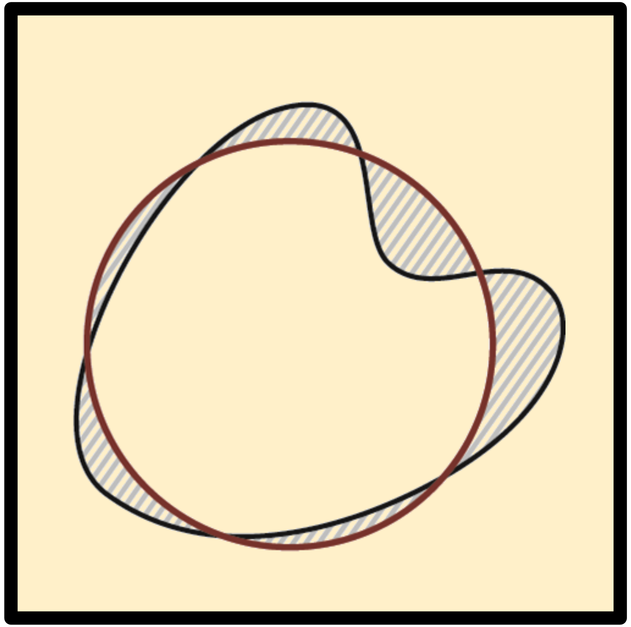
Adrian Butscher Autodesk Research

Distances in Geometry Processing



Point-to-point
"Geodesic"

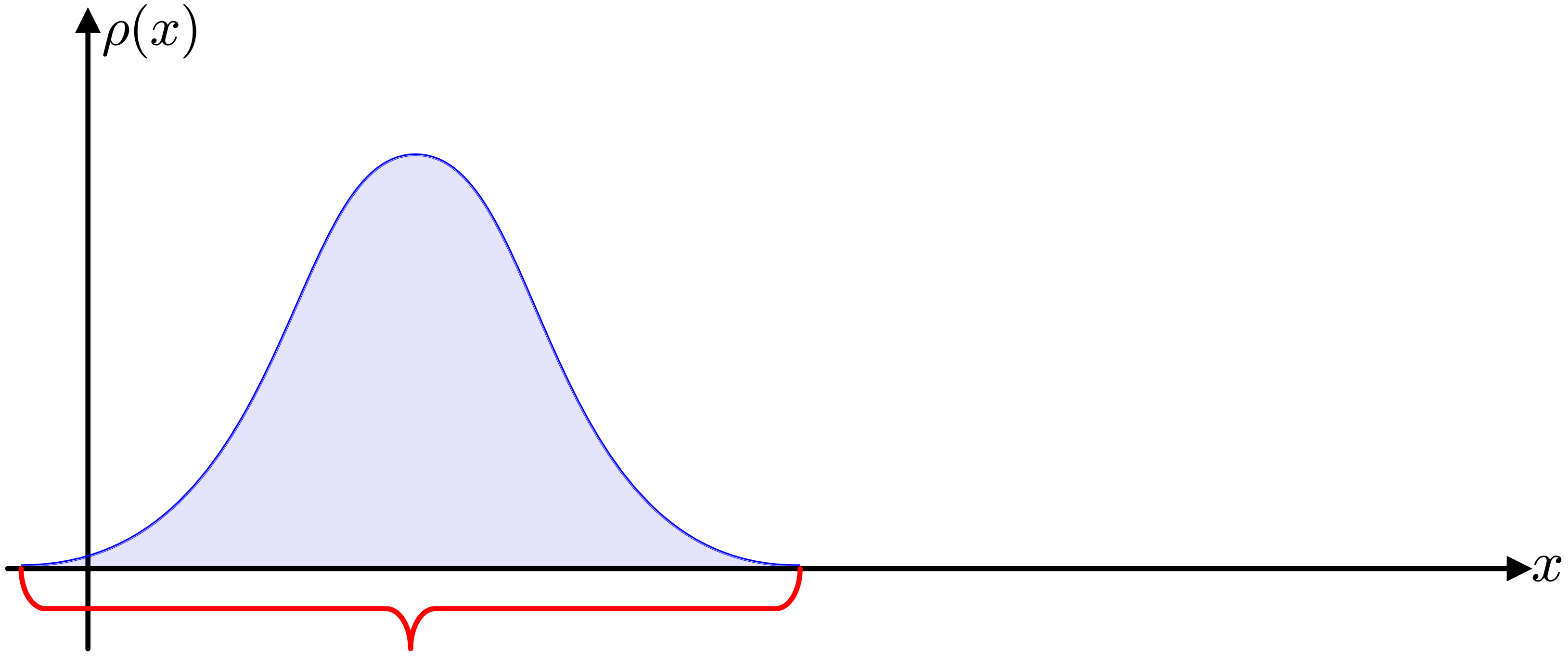
Feature-to-point



Feature-to-feature

Torus by M. Irons, signed distance by R. Kolluri, curve distance by C. Wu

Probabilistic Geometry

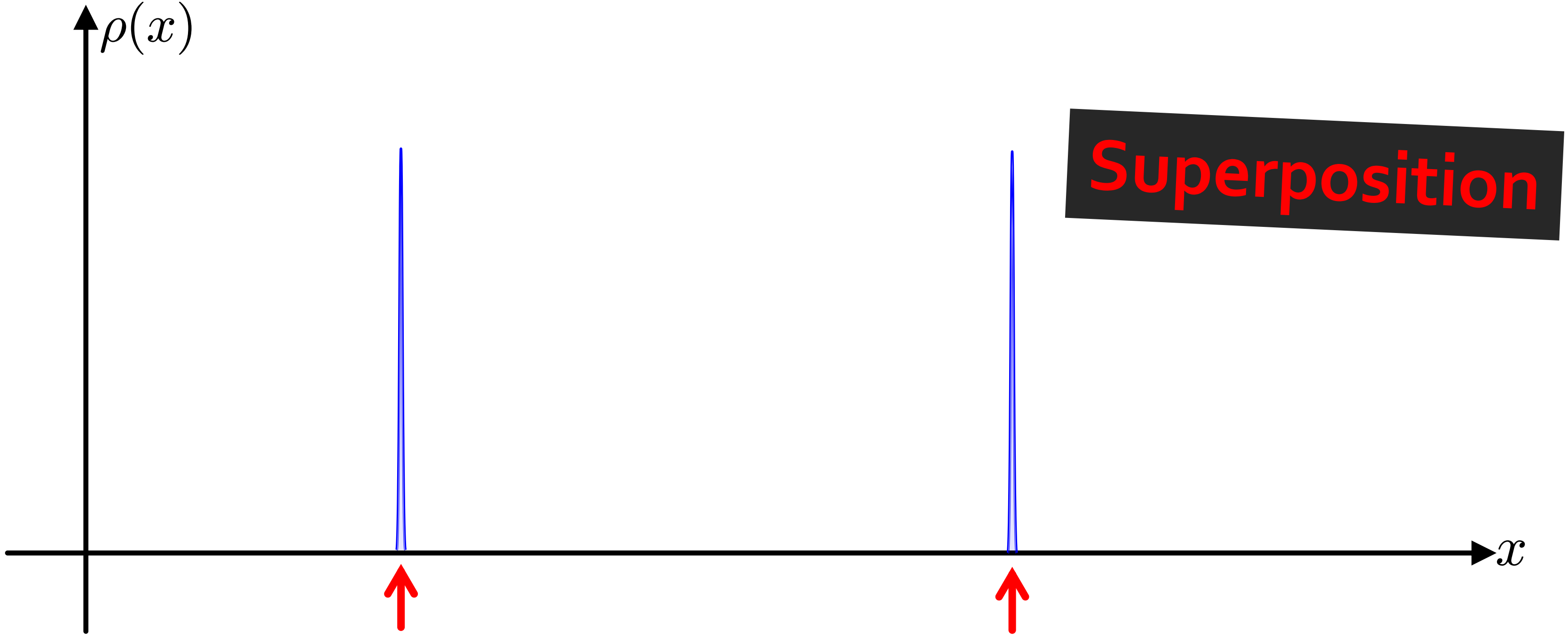


"Somewhere over here."

Probabilistic Geometry

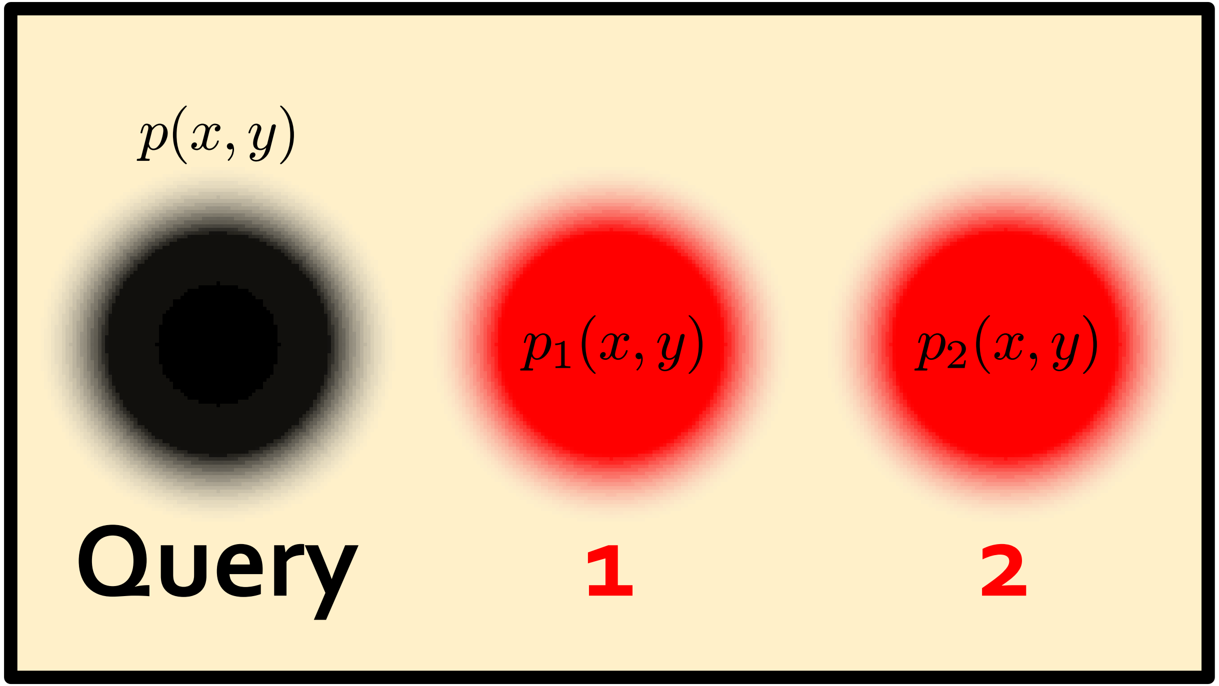
"Exactly here."

Probabilistic Geometry



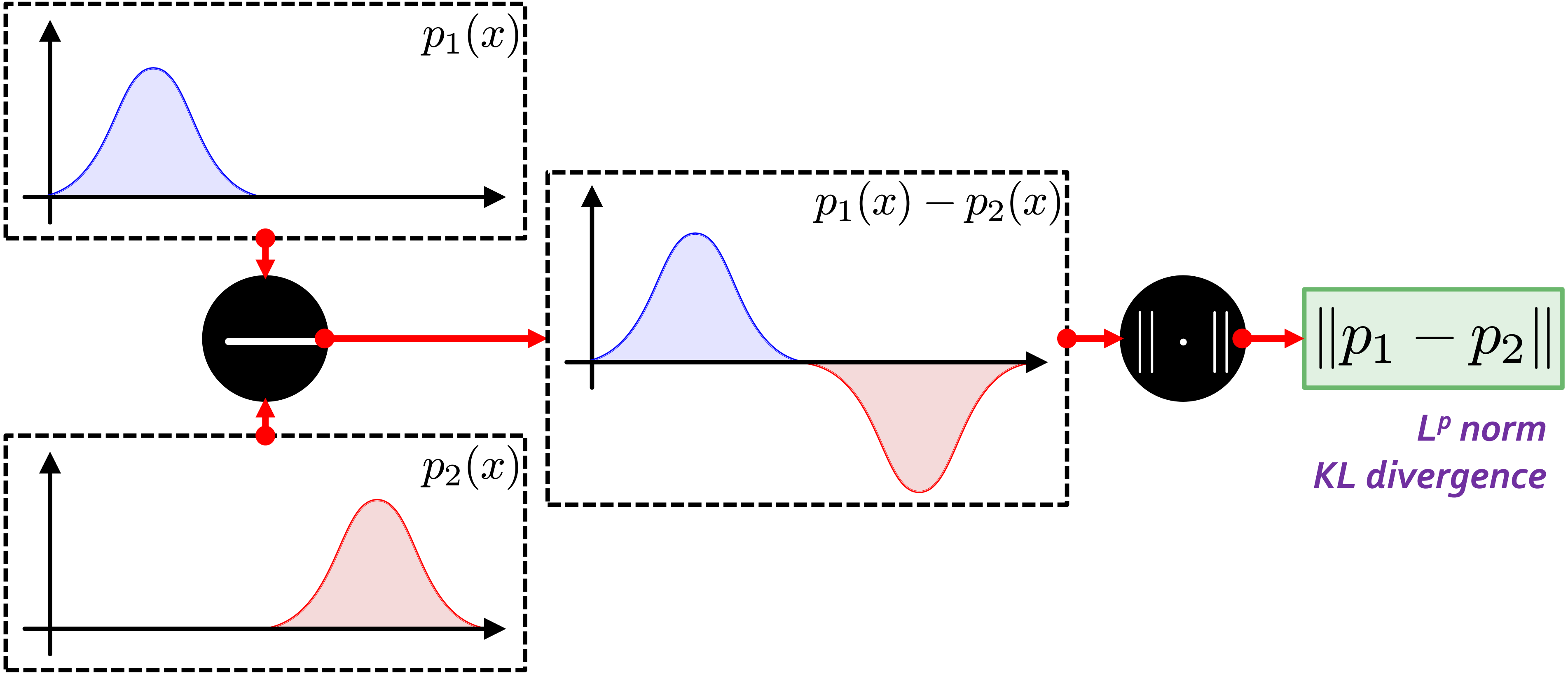
"One of these two places."

Fuzzy Distances



Which is closer, 1 or 2?

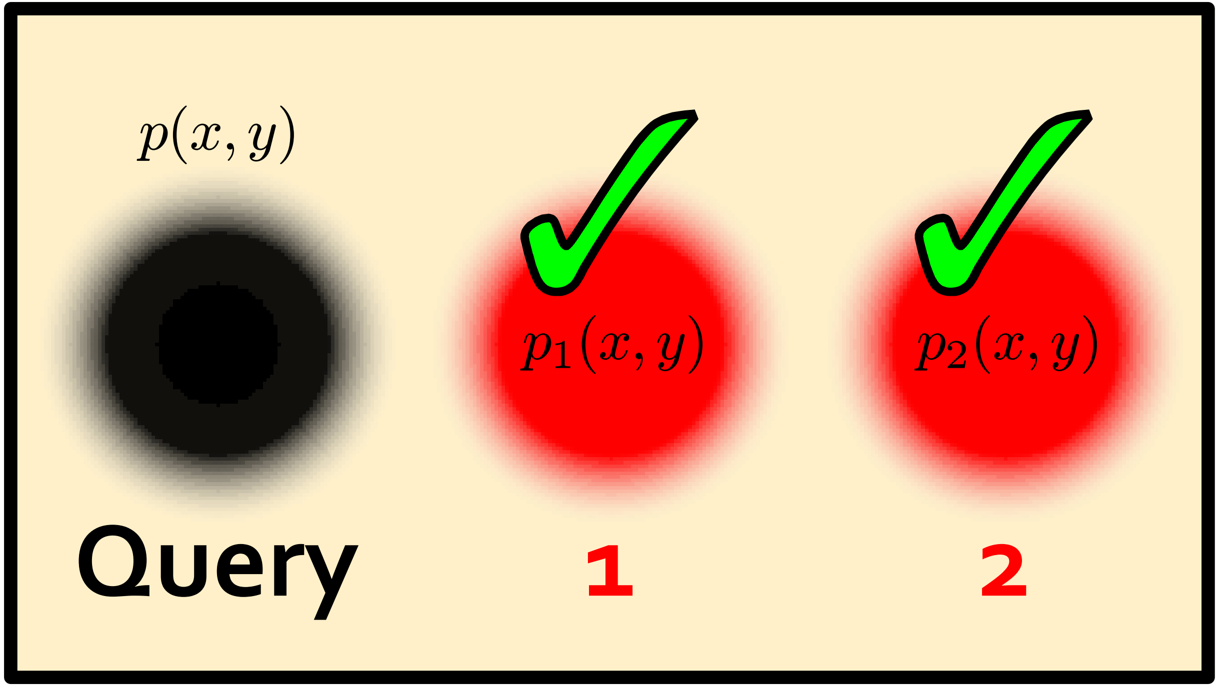
Typical Measurement



L^p norm

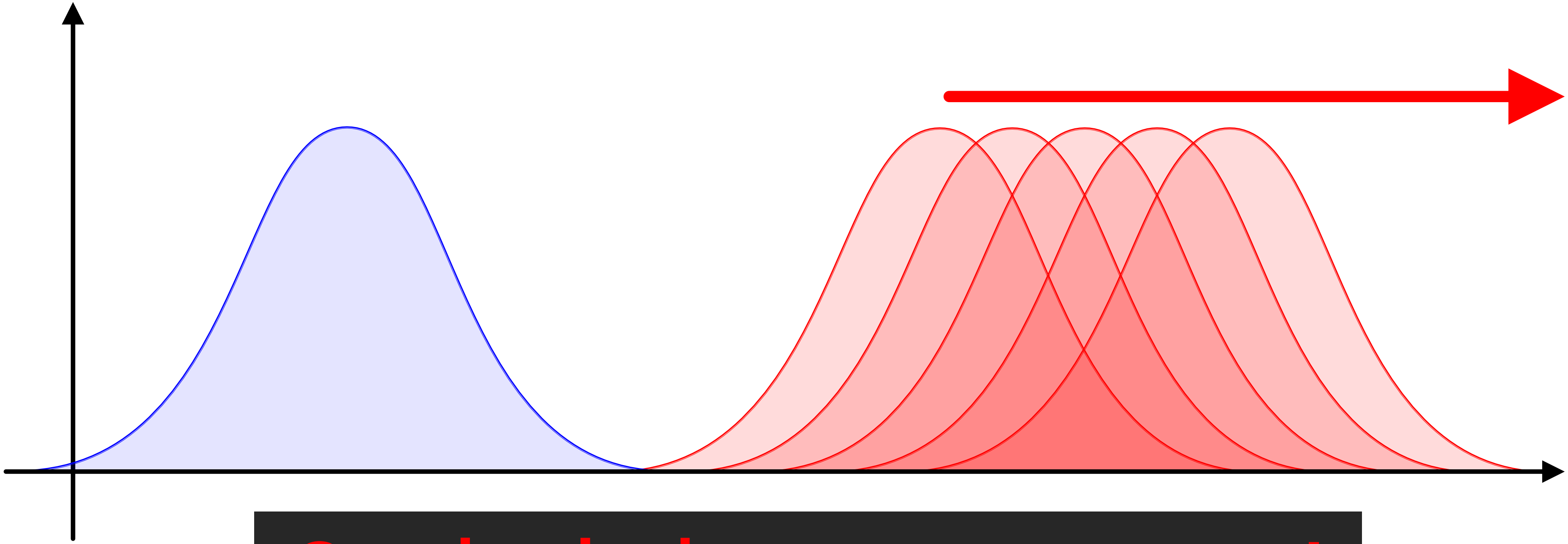
KL divergence

Fuzzy Distances



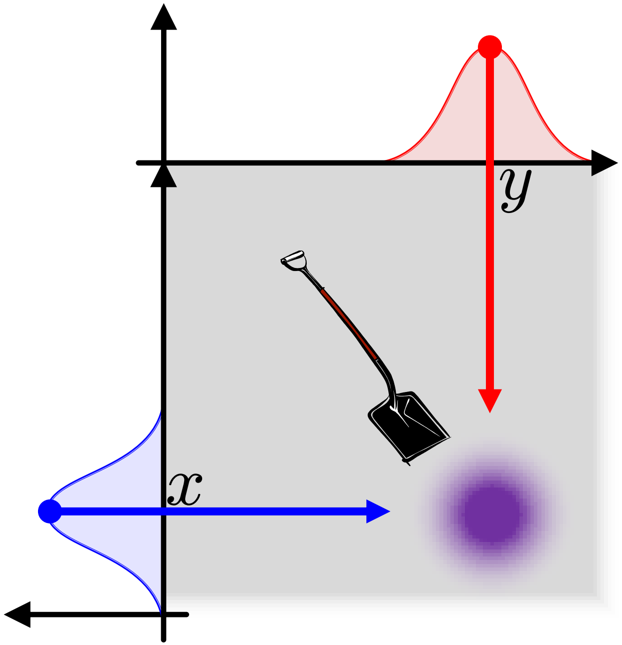
Which is closer, 1 or 2? Equidistant.

What Went Wrong?



Overlap is the wrong measure!

Alternative: Earth Mover's Distance

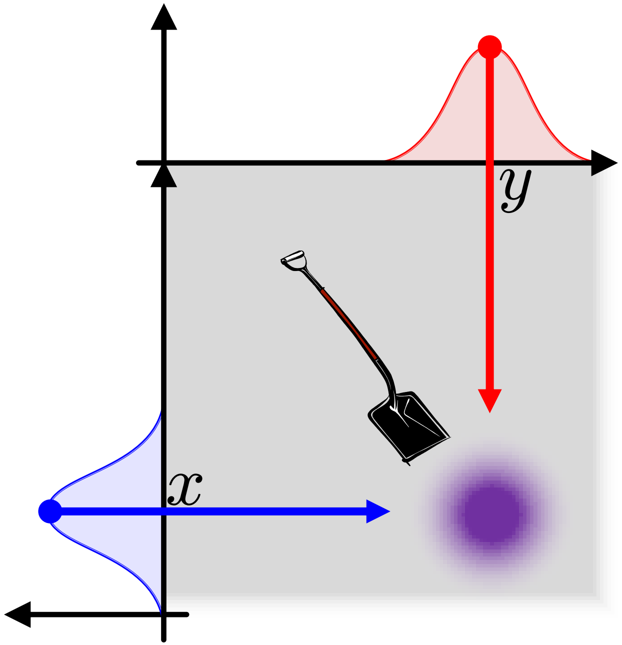


Cost to move mass m from x to y:

$$m \cdot d(x, y)$$

Move mass from one distribution to the other

Alternative: Earth Mover's Distance



$$\min_T \sum_{ij} T_{ij} d(x_i, x_j)$$

$m \cdot d(x, y)$

$$\text{s.t.} \sum_j T_{ij} = p_i$$

Starts at p

$$\sum_i T_{ij} = q_j$$

Ends at q

$$T \geq 0$$

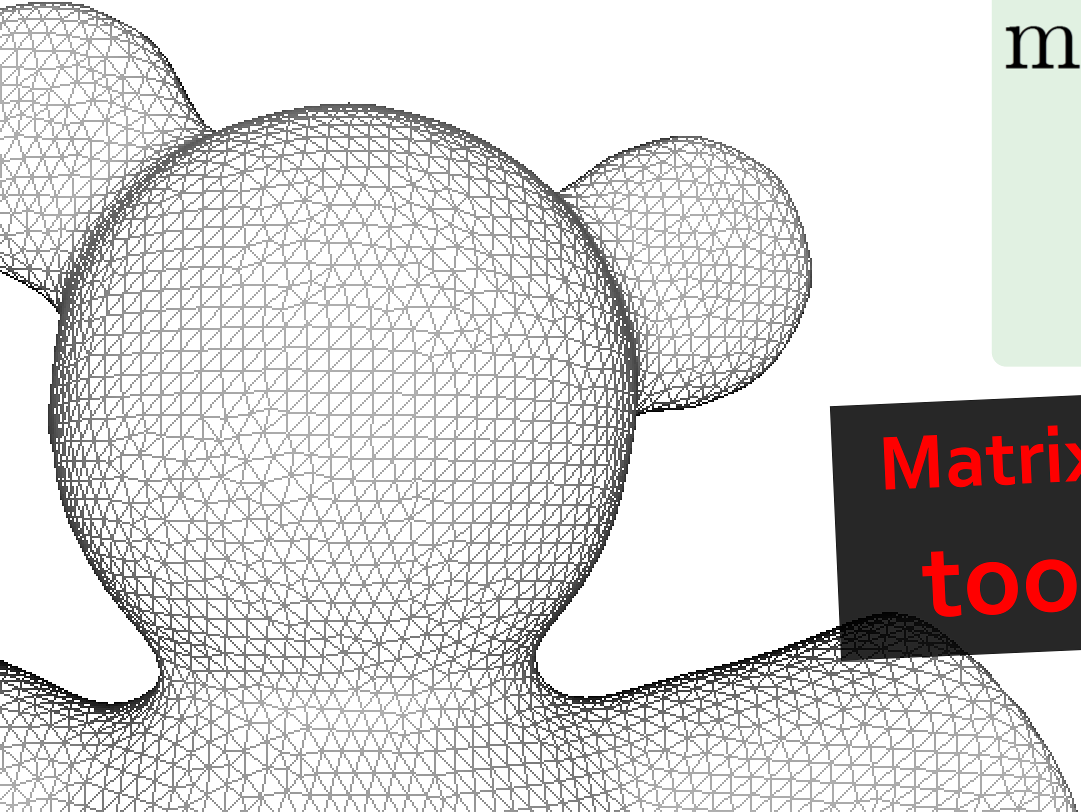
Positive

Move mass from one distribution to the other

Earth Mover's Distance

- **Many names**
  Wasserstein distance, transportation distance, Mallows distance
- **Theoretically sound**
  Regularity properties, continuous and discrete formulations
- **Popular option**
  Computer vision, machine learning, operations, graphics

Computer Graphics Applications



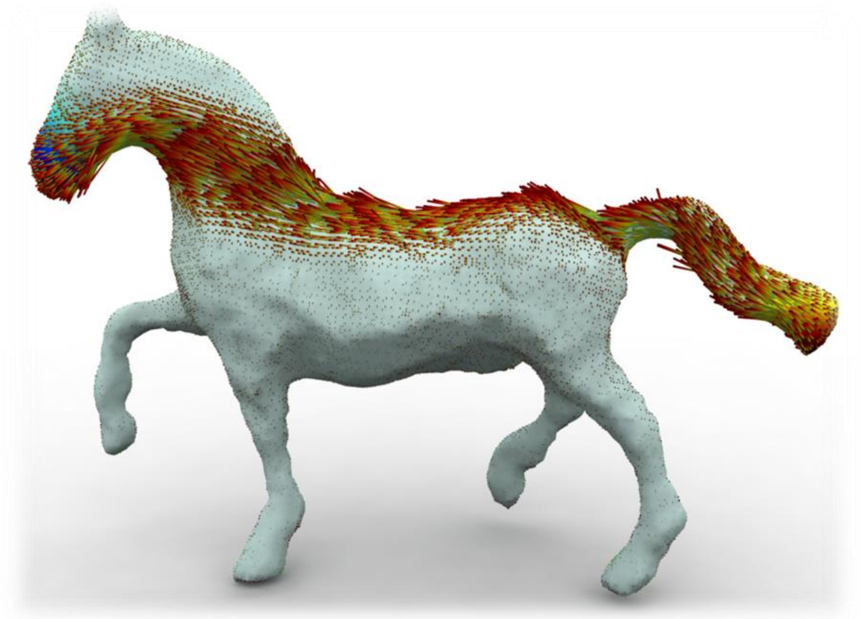
$$\min_T \sum_{ij} T_{ij} d(x_i,$$
$$\text{s.t.} \sum_j T_{ij} = p_i$$
$$\sum_i T_{ij} = q_j$$

Matrix T_{ij} is too big!

Precompute $d(x_i, x_j)$ for all i, j!

Our approach:

Use Eulerian Flow



Probabilities *advect* along the surface

New discretization, optimization, and (consequently) applications!

Think of probabilities like a fluid

Alternative Formulation

Theoretical version: "Beckmann problem"

Total work

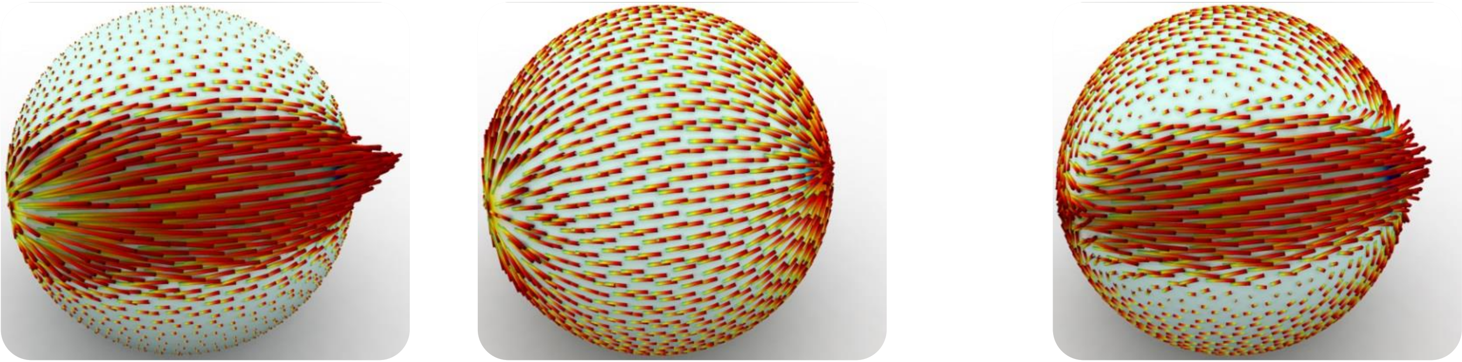
$$\begin{aligned} \inf_J \ & \int_M \|J(x)\|\, dx \\ \text{s.t. } & \nabla \cdot J(x) = \rho_1(x) - \rho_0(x) \\ & J(x) \cdot n(x) = 0 \ \forall x \in \partial M \end{aligned}$$

Advects from ρ_0 to ρ_1

Scales linearly

Hodge Decomposition of J

$$J(x) = \nabla f(x) + \mathcal{R} \cdot \nabla g(x)$$



Curl-free

Div-free

$$\nabla \cdot J = \Delta f = \rho_1 - \rho_0$$

New idea!

Fast Optimization

1. $$\Delta f = \rho_1 - \rho_0$$ **Sparse SPD linear solve for *f***

2. $$\inf_g \int_M \|\nabla f(x) + \mathcal{R} \cdot \nabla g(x)\| \, dx$$

Unconstrained and convex optimization for *g*

Fast Optimization

1. $$\Delta f = \rho_1 - \rho_0$$ Sparse SPD linear solve for *f*

2. $$\inf_g \int_M \|\nabla f(x) + \mathcal{R} \cdot \nabla g(x)\| \, dx$$

Unconstrained and convex optimization for *g*

- **Piecewise-linear FEM, optimized via ADMM**
- **Spectral approximation *(optional)***

$$g(x) = a_1\phi_1(x) + a_2\phi_2(x) + a_3\phi_3(x) + \cdots$$
$$\Delta\phi_k = \lambda_k\phi_k$$

Satisfies triangle inequality!

Fast Optimization

function ADMM-WASSERSTEIN(ρ_0, ρ_1)

$\triangleright$ ρ_0, ρ_1 have one value per vertex

$\triangleright$ Concatenate B_t's vertically to obtain B

$f \leftarrow \Delta^{+}(\rho_1 - \rho_0)$ $\triangleright$ Solve for gradient part

$v \leftarrow \nabla f$ $\triangleright$ Compute gradient vector field

for $i \leftarrow 1, 2, 3, \ldots$ $\triangleright$ Iterate until convergence

$z_t \leftarrow B_t c + w_t - \frac{y_t}{\beta}$ $\triangleright$ Update vector field J

$$\alpha_t \leftarrow \begin{cases} 1 - \frac{1}{\beta \|z_t\|} & \beta\|z_t\| > 1 \\ 0 & \text{otherwise} \end{cases}$$

$J_t \leftarrow a_t z_t$

$\triangleright$ Update coefficients; can pre-factor

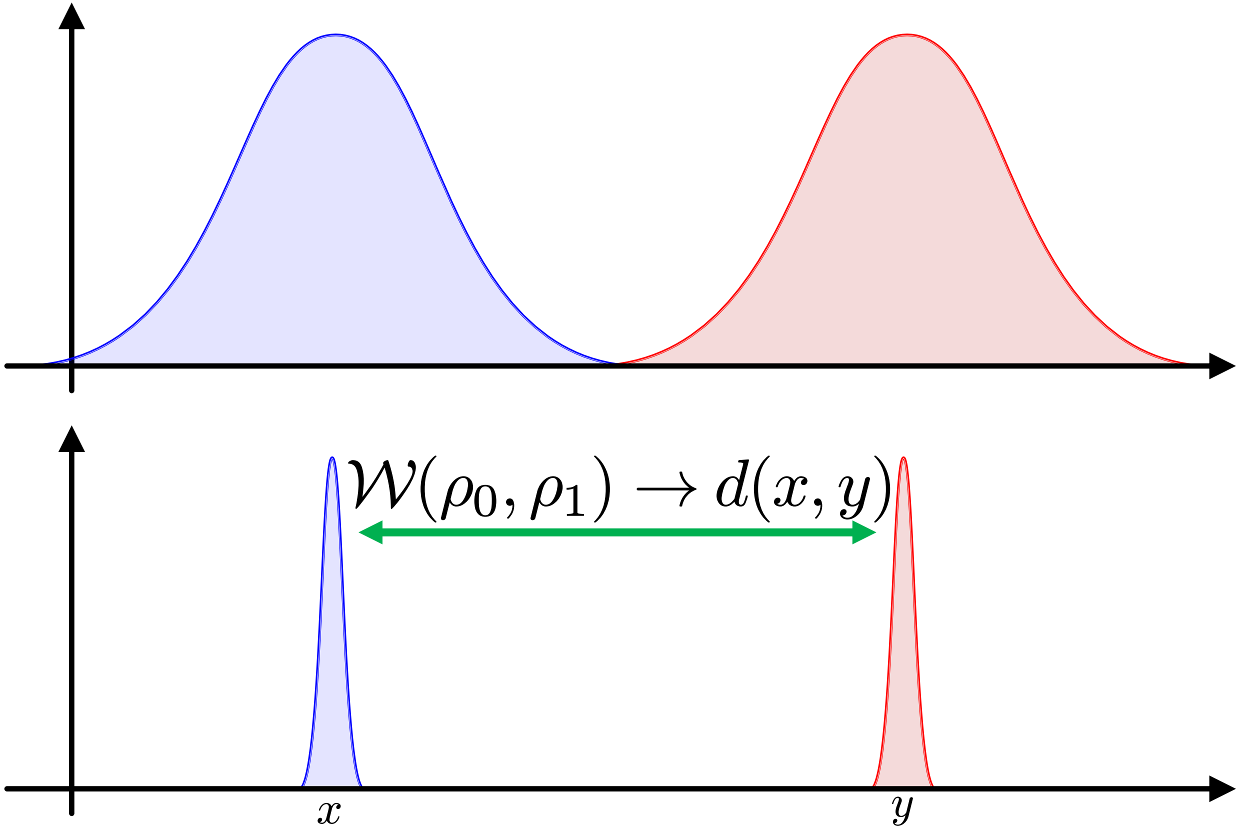
$$c \leftarrow \left(\sum_t B_t^\top B_t\right)^{-1} \left[\sum_t B_t^\top \left(\frac{y_t}{\beta} + J_t - w_t\right)\right]$$

$y_t \leftarrow y_t + \beta(J_t - B_t c - w_t)$ $\triangleright$ Update dual

return $J_t \ \ \forall t \in T$

Iterations are fast and easy to implement!

Pointwise Distance



Pointwise Distance

Geodesic

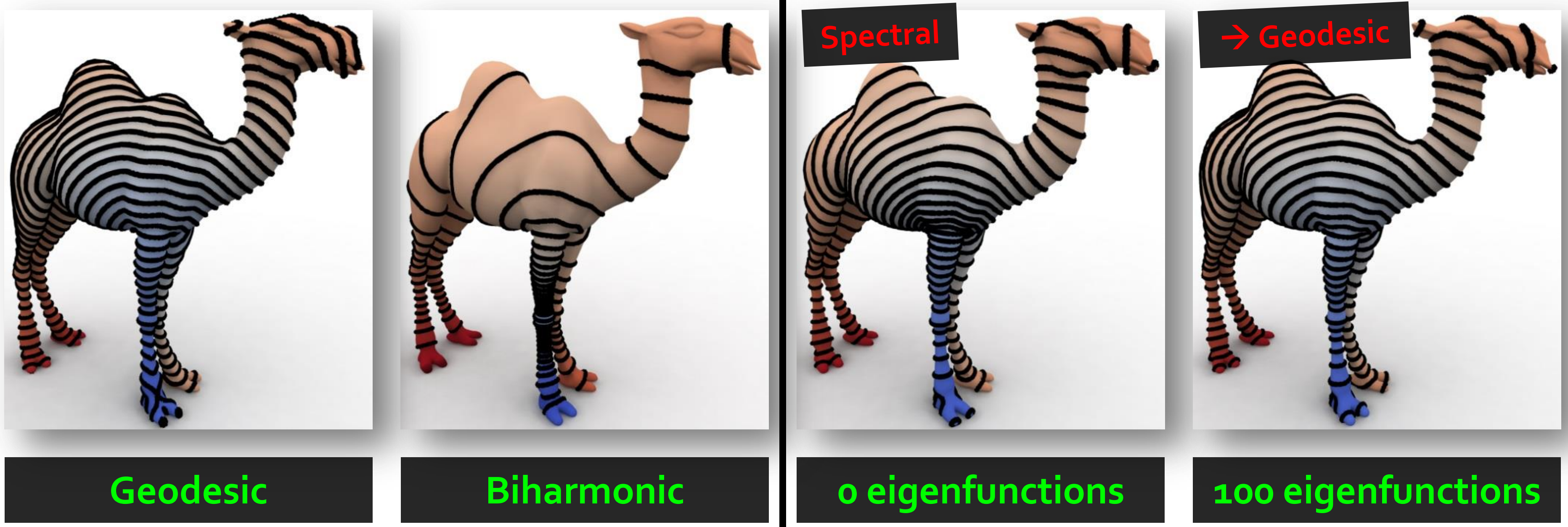
Biharmonic

0 eigenfunctions

100 eigenfunctions

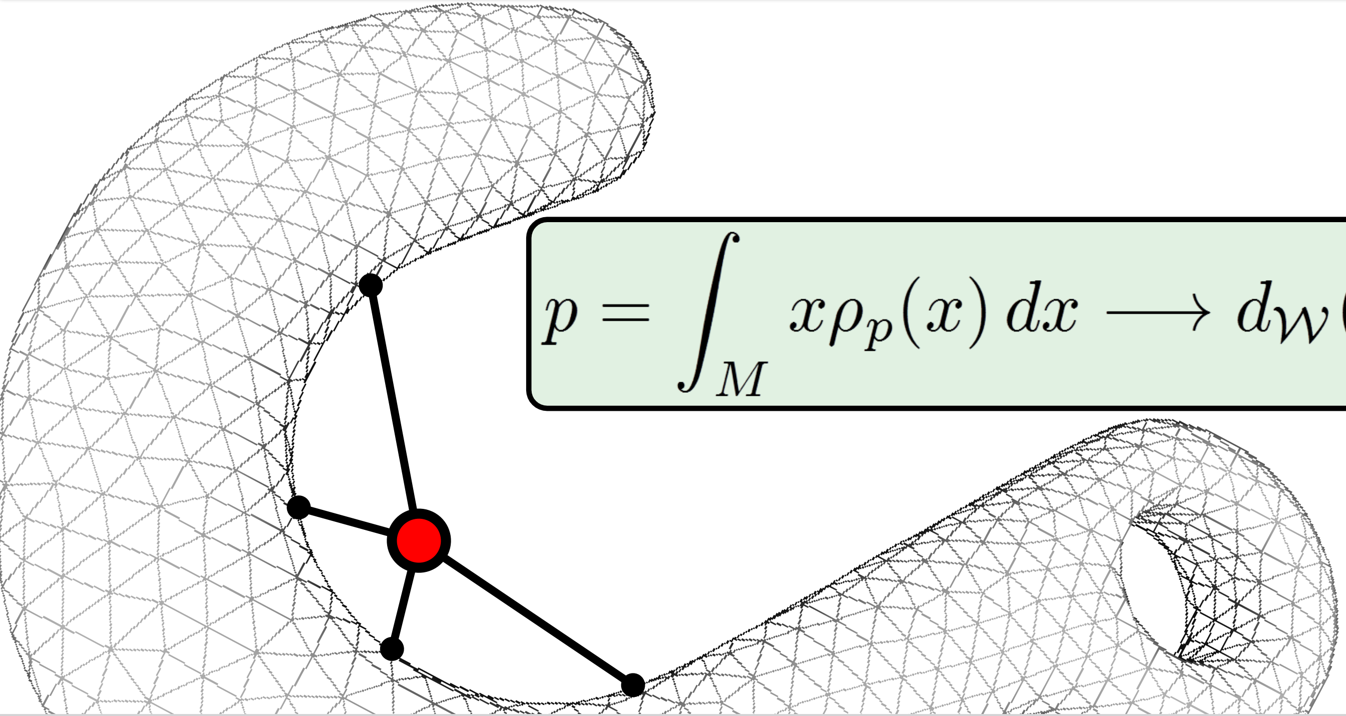
Proposition: Satisfies triangle inequality.

Pointwise Distance



Proposition: Satisfies triangle inequality.

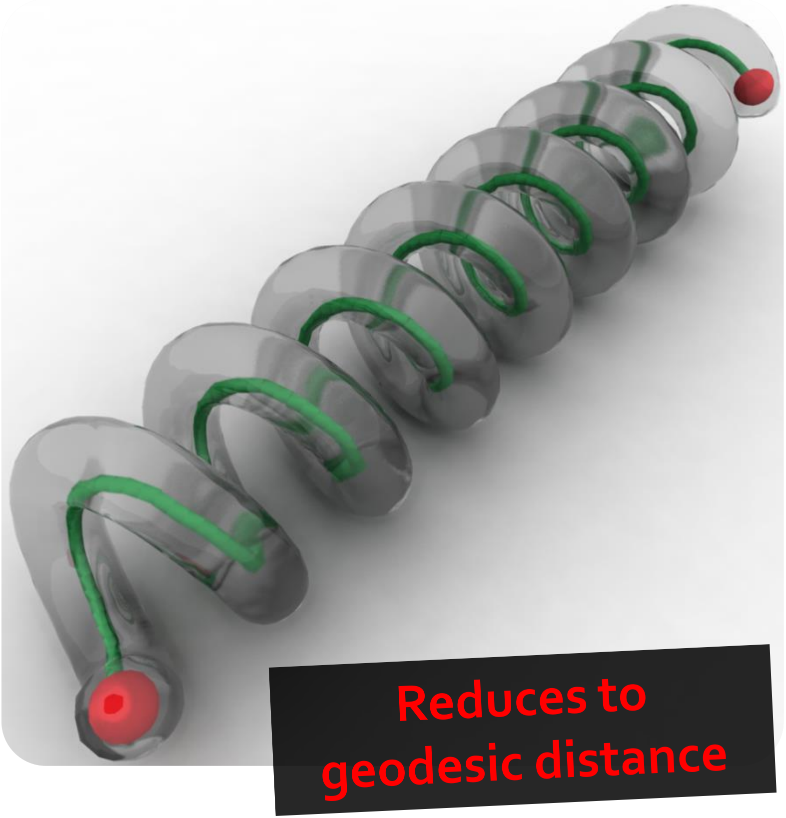
Volumetric Distance



$$p = \int_M x\rho_p(x)\,dx \longrightarrow d_{\mathcal{W}}(p,q) \equiv \mathcal{W}(\rho_p, \rho_q)$$

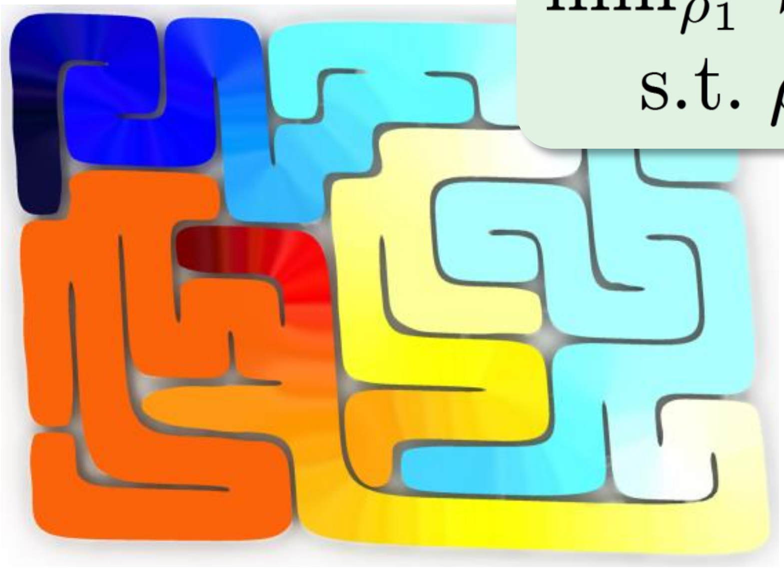
Use barycentric coordinates (mean value)

Volumetric Distance

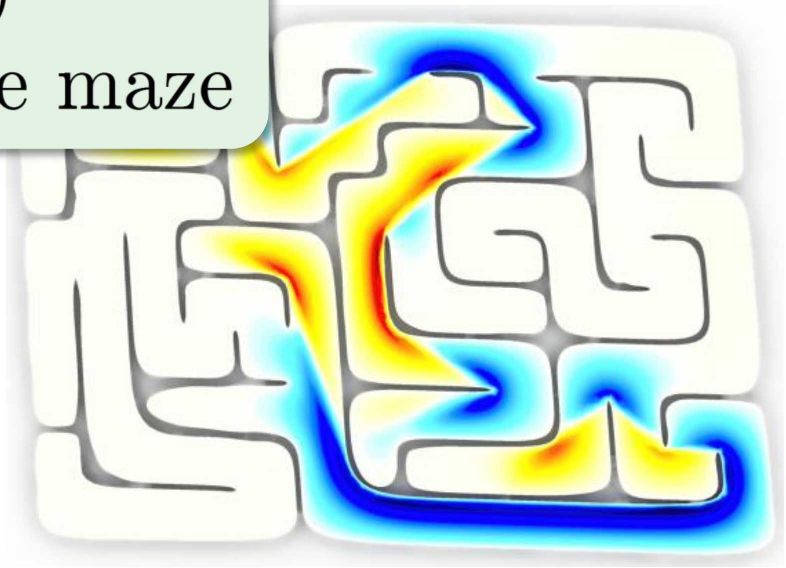


EMD in Optimization

$$\min_{\rho_1} \mathcal{W}(\rho_0, \rho_1)$$
$$\text{s.t. } \rho_1 \text{ outside maze}$$

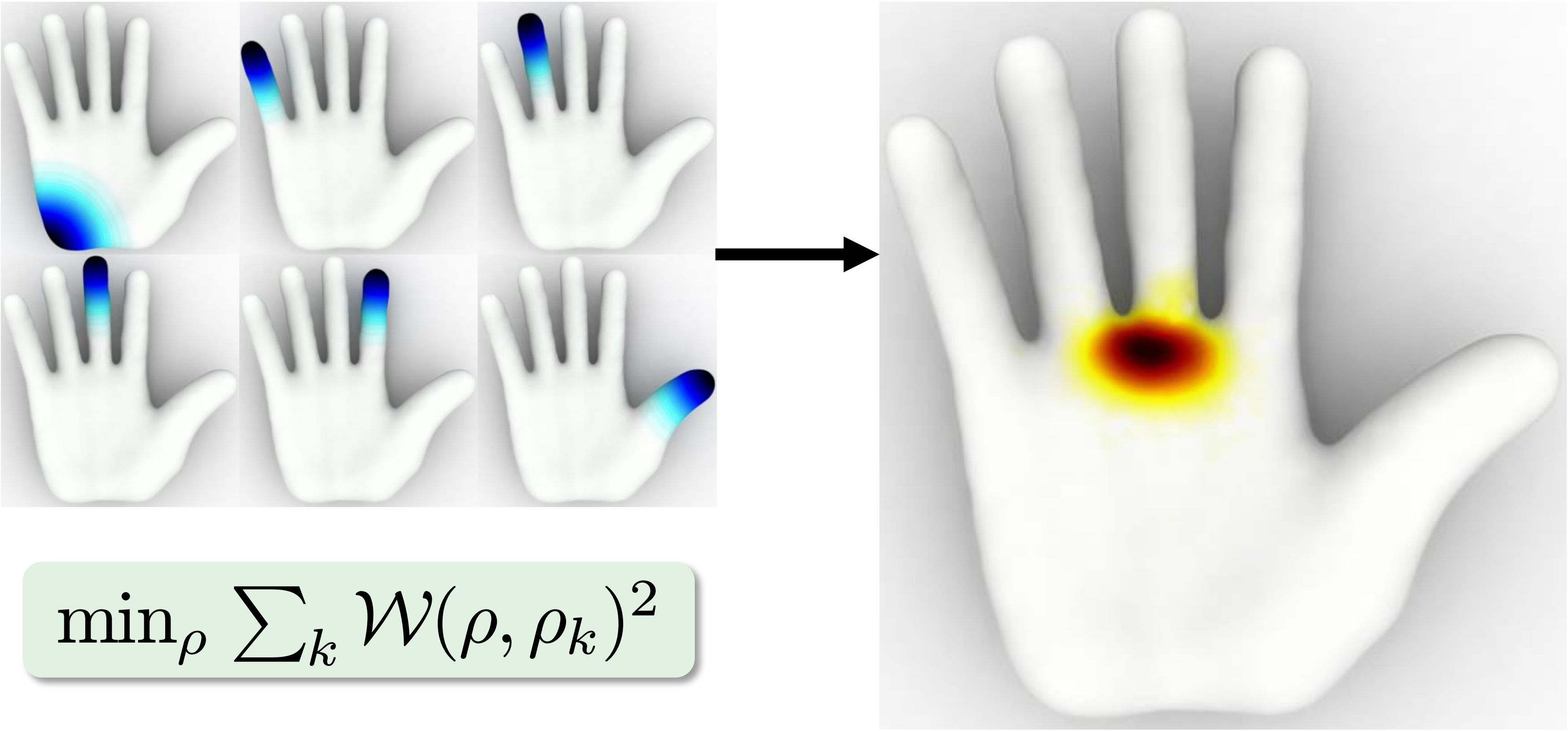


Curl-free



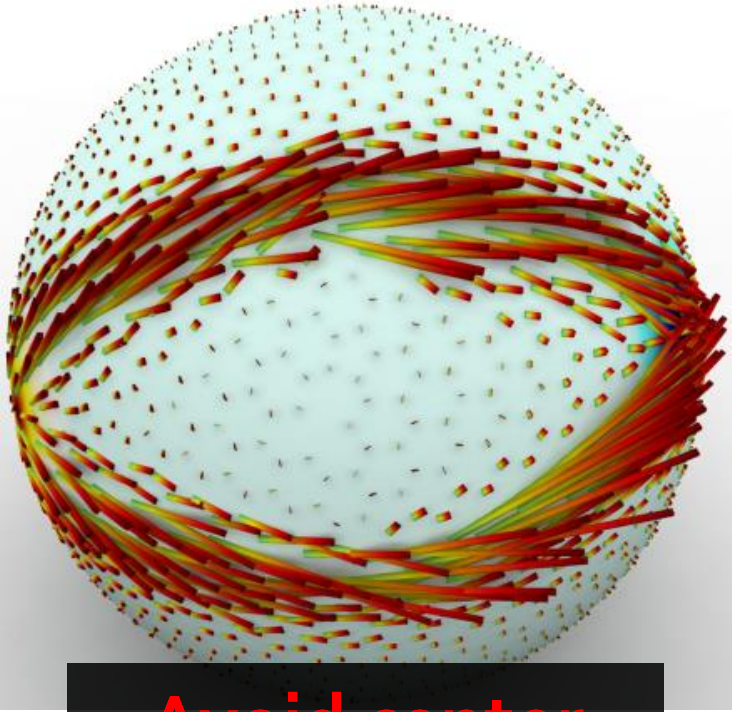
Div-free

Barycenter Computation



$$\min_\rho \sum_k \mathcal{W}(\rho, \rho_k)^2$$

Variations of EMD

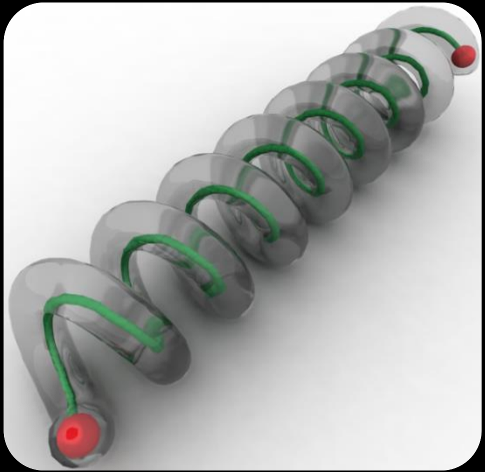


Avoid center

Distance to feature

What's Next?

- Quadratic ground distance
- Other representations
  Point clouds? Polygon soup? Graphs?
- Faster optimization

Earth Movers Distances on Discrete Surfaces

Thanks!

Matlab code online!

Timings

Mesh	n_{vert}	d_g	d_h	d_b	$d_{\mathcal{W}}^{0}$	$d_{\mathcal{W}}^{20}$	$d_{\mathcal{W}}^{100}$
Bearing	3182	0.050	0.002	3.52	3.86	30.8	41.4
David	5197	0.096	0.003	10.09	6.18	86.5	121.2
Dog	3716	0.056	0.002	4.66	3.27	38.7	59.8
Teapot	3900	0.063	0.002	6.25	3.87	45.2	57.9
Man	10050	0.18	0.006	42.2	23.2	312.0	511.9

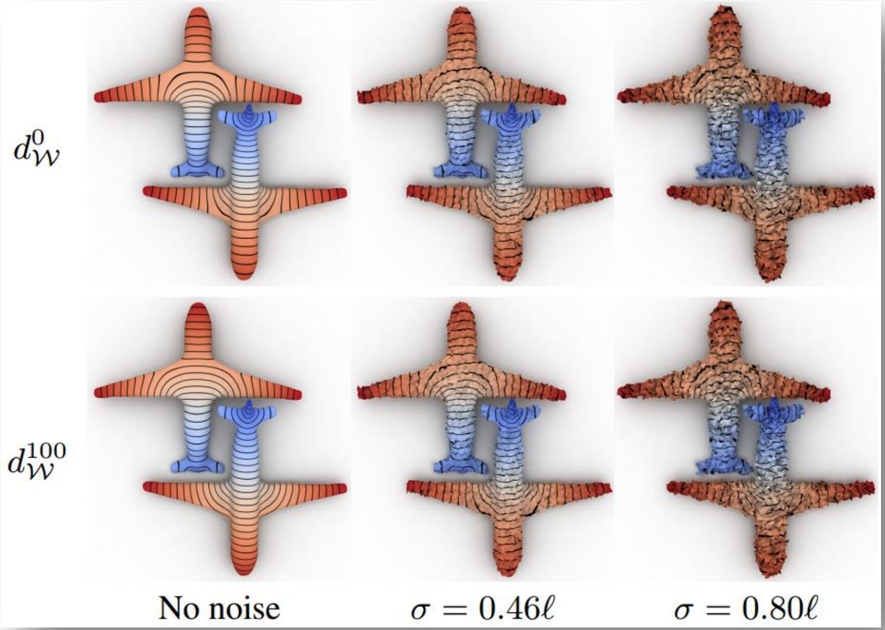
Single-source all-targets

Timings

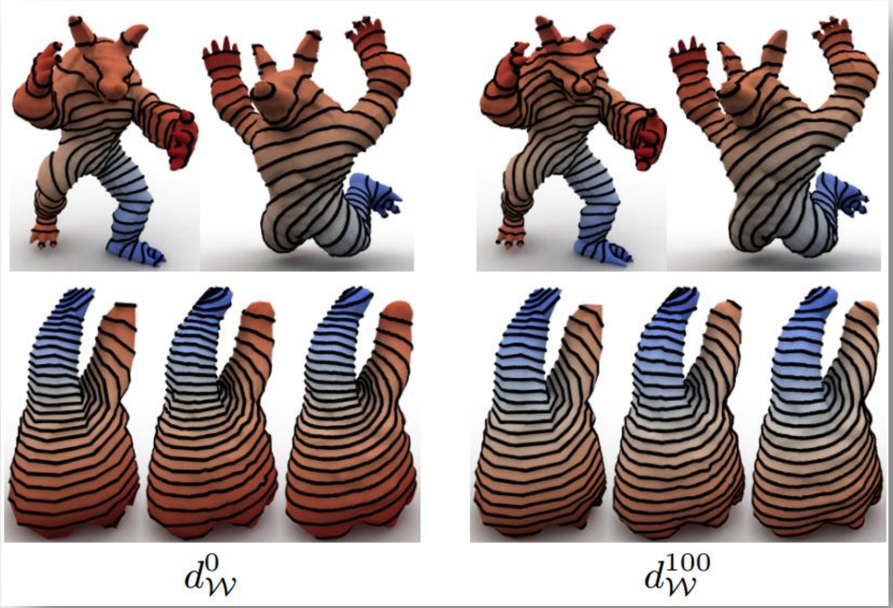
Mesh size		M for d_g		M for d_h		M for d_b		M for $d^0_{\mathcal{W}}$	
n_{vert}	n_{tri}	**2**	**100**	**2**	**100**	**2**	**100**	**2**	**100**
2k	4k	0.06	2.60	0.03	0.23	0.03	0.58	0.03	1.22
4k	9k	0.13	6.25	0.05	0.45	0.06	1.42	0.06	2.84
8k	16k	0.24	11.76	0.10	0.97	0.14	4.97	0.14	7.33
16k	32k	0.70	34.93	0.20	1.97	0.33	13.07	0.34	18.45
53k	105k	2.74	121.94	0.71	10.36	1.03	51.99	0.97	68.53
111k	222k	8.06	432.28	2.04	15.14	10.91	289.02	11.00	322.11

All-pairs for sample of *M* points

Robustness

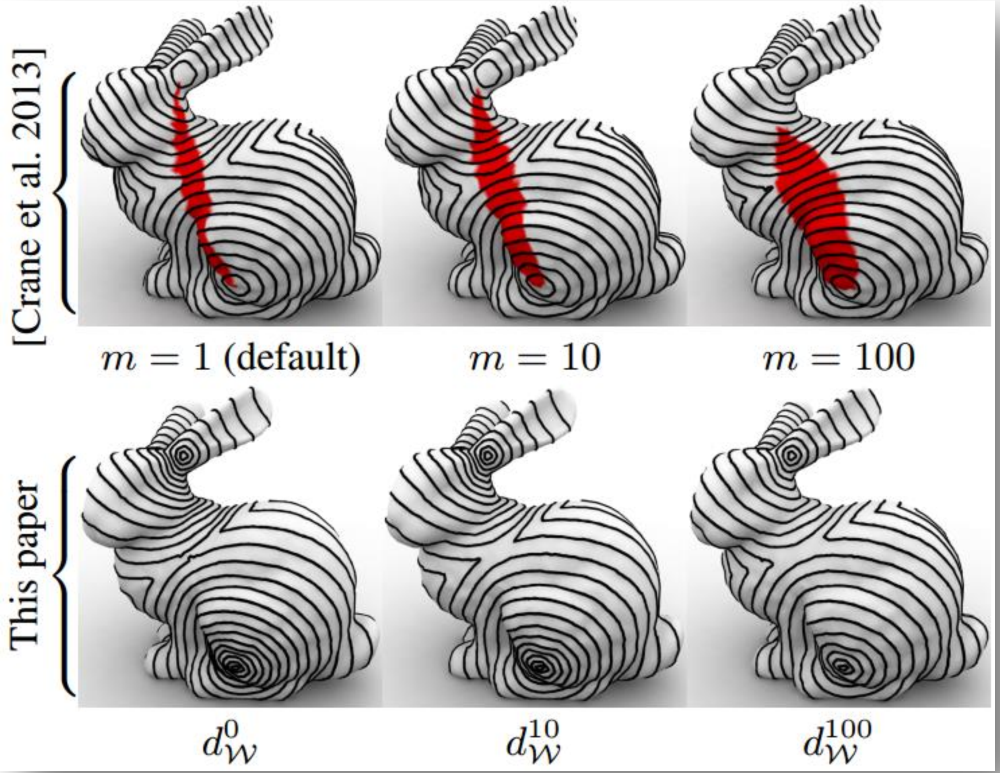


Perturbation



Isometry and remeshing

Triangle Inequality



Fix p and q; red points are where $d(p,\cdot) + d(\cdot,q) < d(p,q)$.