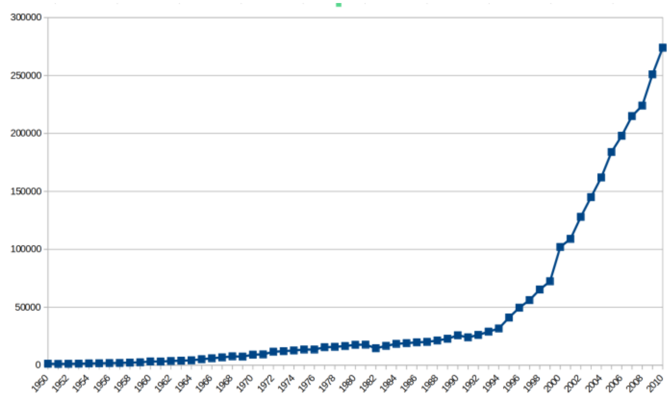
A survey of Optimal Transport for Computer Graphics and Computer Vision

Nicolas Bonneel and Julie Digne
CNRS, univ Lyon 1

Eurographics State of the Art Report 2023

Why a survey on Optimal Transport?



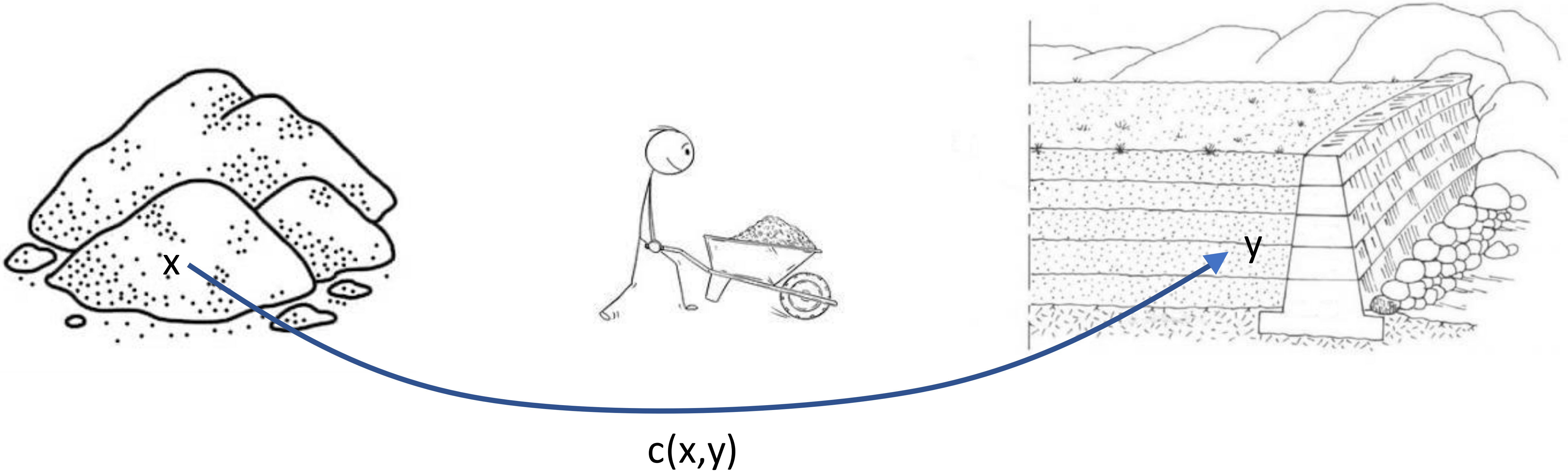
Number of papers containing "Optimal Transport" over the years

- Starts to be well established in CG
  - Not as crazy as ChatGPT / generative AI, NeRF...
  - Still much work has been done : survey is not exhaustive
- Numerical techniques start to be fast enough for "real" applications
- No survey dedicated to applications in CG
  - Though interesting surveys on numerical techniques, e.g. [Peyré & Cuturi 2019]
  - We will also cover "numerics for the layman"
- Target audience:
  - People wanting to understand optimal transport for their own applications
  - Experts in optimal transport wanting to find new applications

Outline

- Introduction to optimal transport principles
- Common numerical solvers
- Applications:
  - Image processing / texture synthesis
  - Rendering / sampling
  - Geometry / topology
  - Animation / simulation

Intuition

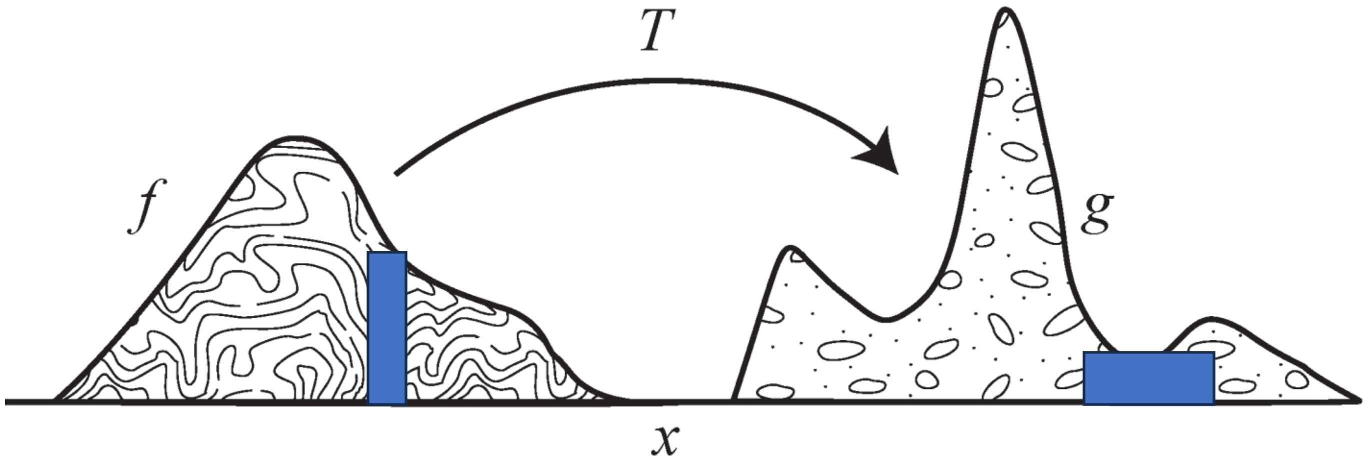


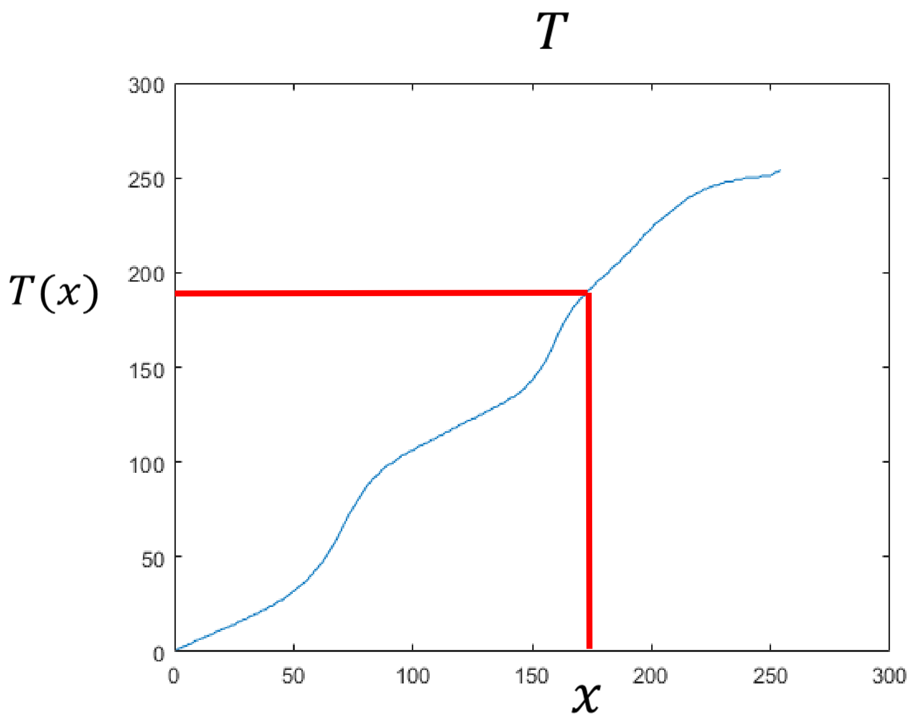
Intuition: Monge

$$W(f,g) = \min_T \int_X c\left(x, T(x)\right) f(x)\, \mathrm{dx}$$

s.t. $\quad f(x) = g(T(x))\, |\det J_T(x)|$

(typo in paper!)





Same total mass

"find a good warping between f and g with the change of variable formula"

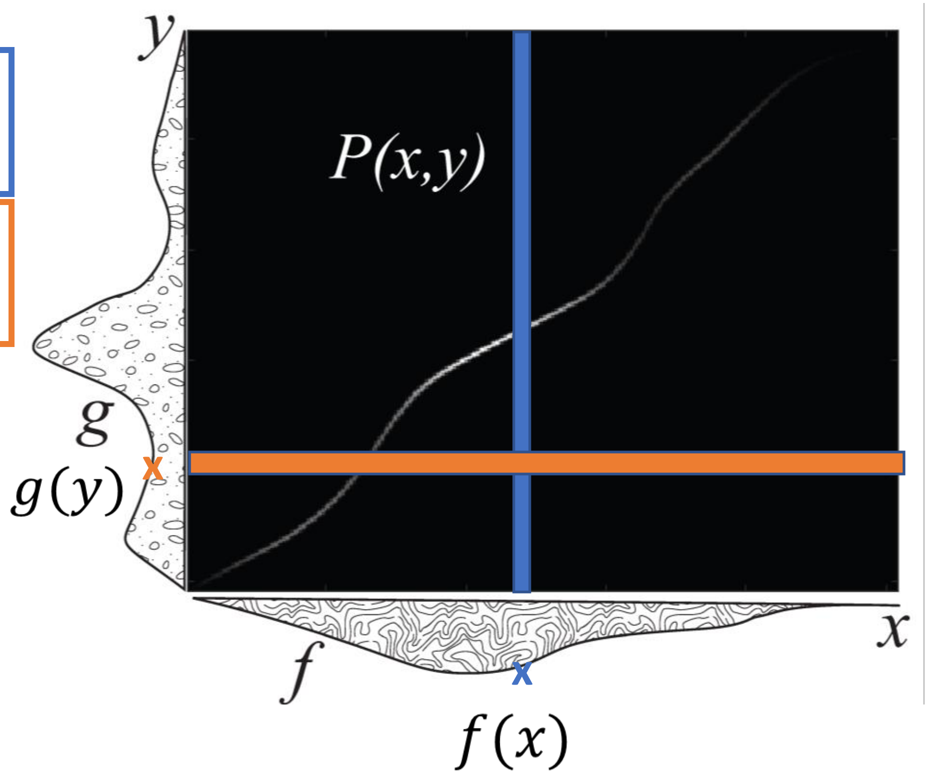
Intuition: Kantorovich

$$W(f,g) = \min \iint_{X\times Y} c(x,y)\,d\pi(x,y)$$

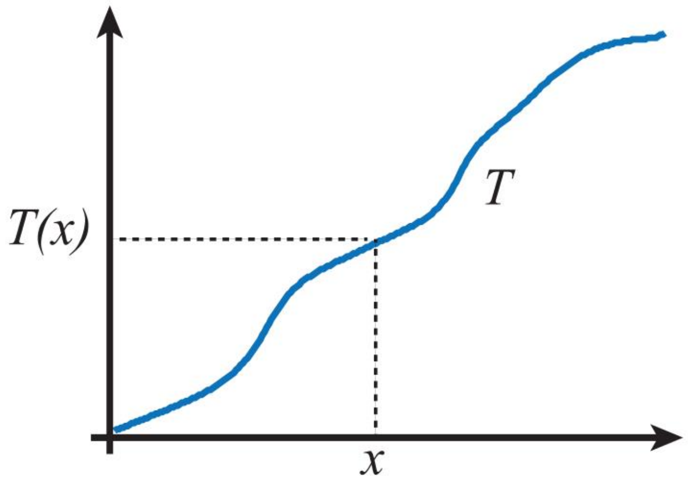
s.t. "$\int_Y d\pi(x,y) = f(x)dx$" $\quad \forall x$

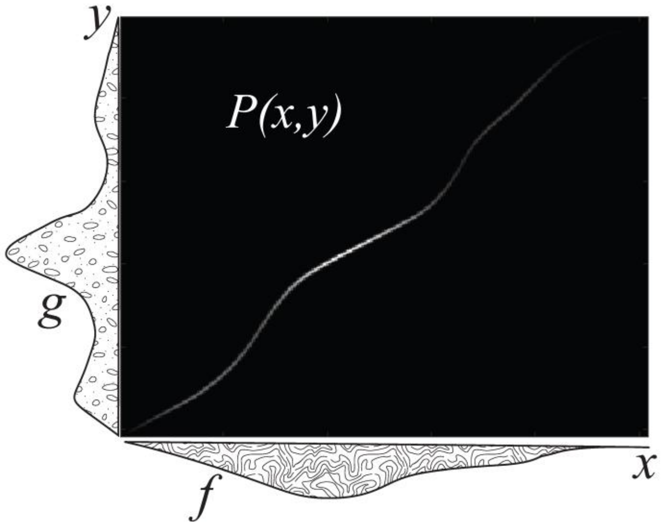
$\int_X d\pi(x,y) = g(y)dy \quad \forall y$

$d\pi(x,y) \geq 0$



Intuition: comparison





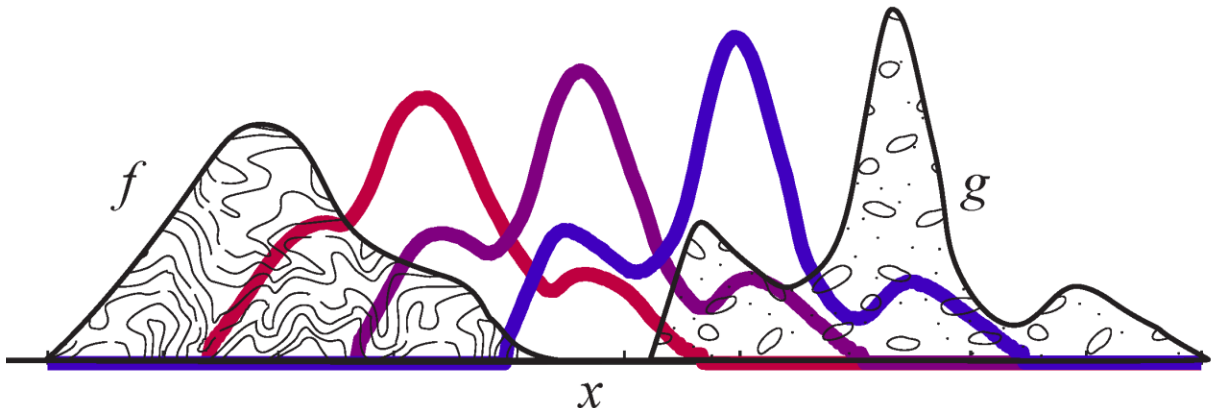
- Finds a "transport map"
- Difficult non-linear problem
- May have no solution
  (e.g., a Dirac splitting in two)
- Leads to PDEs

- Finds a "transport plan"
- Linear program
- "Always" has solution
  (i.e., under reasonable assumptions)
- Also has dual formulation

When it exists, the solution is the same.

Often, $c(x, y) = \|x - y\|_p^p \Rightarrow W_p^p$
W_p is a distance

Wasserstein barycenters



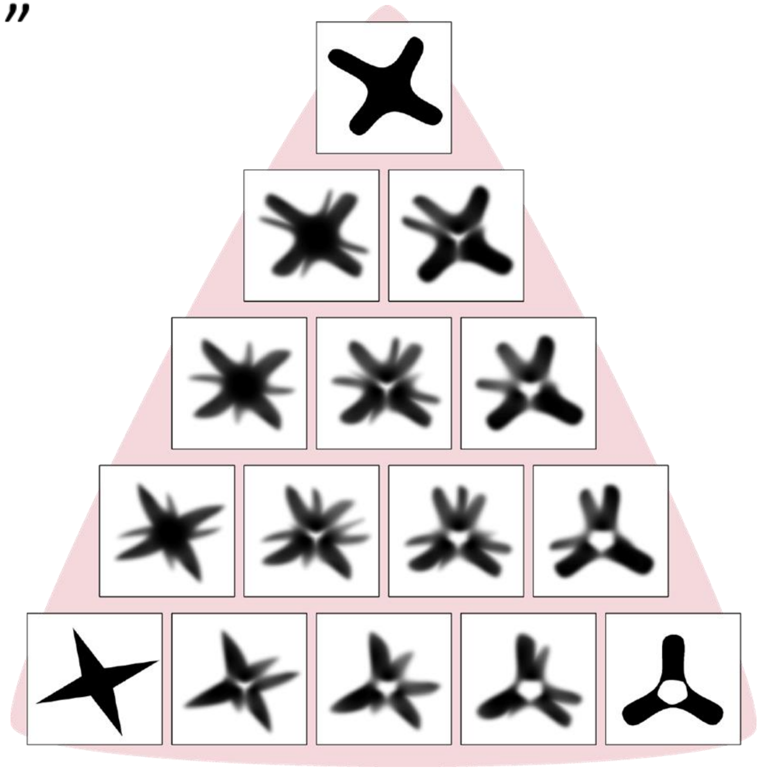
Interpolating between 2 probability distributions f and g

→ Displacement interpolation

→ Uses maps $T_t = (1-t)\mathrm{Id} + t.T$

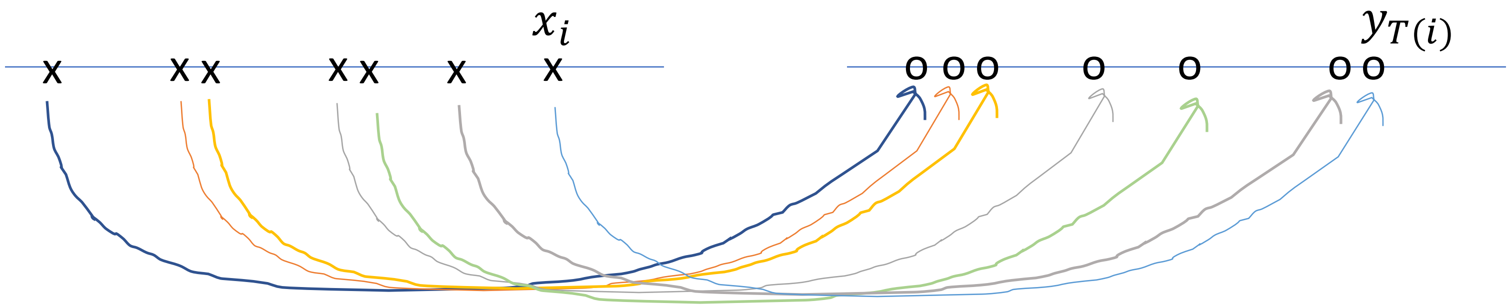
Wasserstein barycenters

- Displacement interpolation generalizes to $n \geq 2$ distributions
- Interpolation called "Wasserstein Barycenter"
- Minimizes: $\mathrm{b} = \min_h \sum_k \lambda_k W_2^2(h, f_k)$



Simple cases

- Optimal transport and barycenters trivially solved for
  - 1-d distributions with c(x,y) convex function of $||x - y||$
    - Solutions expressed through cumulative density functions $F(x) = \int_{-\infty}^{x} f(t)dt$
    - $W(f, g) = \int_0^1 c\big(F^{-1}(x) - G^{-1}(x)\big)dx$
    - $T(x) = G^{-1}\big(F(x)\big)$: always the same map
    - $B^{-1}(x) = \sum_k \lambda_k F^{-1}(x)$ and then barycenter b obtained by inversion and differentiation
  - 1-d discrete distributions (same setting): $\mu = \sum_{i=1}^{n} \delta_{x_i}$, $\nu = \sum_{i=1}^{n} \delta_{y_i}$



Simple cases

- Optimal transport and barycenters trivially solved for
  - Gaussian distributions with $c(x, y) = \|x - y\|^2$
    - $W_2^2(\mathcal{N}_0, \mathcal{N}_1) = \mathrm{tr}\big(\Sigma_0 + \Sigma_1 - 2\Sigma_{0,1}\big) + \|\mu_0 - \mu_1\|$ with $\Sigma_{0,1} = \left(\Sigma_0^{\frac{1}{2}} \Sigma_1 \Sigma_0^{\frac{1}{2}}\right)^{1/2}$
    - $T(x\ ) = \Sigma_{0,1} x$
    - Barycenter: $\mathcal{N}(\mu, \Sigma)$ with $\mu = \sum_k \lambda_k \mu_k$ and iterations

$$\Sigma^{(n+1)} = \sum_k \lambda_k \left(\sqrt{\Sigma^{(n+1)}}\Sigma_k \sqrt{\Sigma^{(n+1)}}\right)^{1/2}$$

Numerical methods

Numerical Algorithms

- Most common algorithms:
  - Linear programming
  - Entropy-regularized solutions
  - Geometric constructions for the semi-discrete problem
  - Sliced approximations
  - PDEs

Linear programming

- Solves the discrete problem

$$\min_P \sum_{ij} c(x_i, y_j)\, P_{ij}$$
$$s.t. \quad \sum_j P_{ij} = f(x_i)$$
$$\sum_i P_{ij} = g(y_j)$$
$$P_{ij} \geq 0$$

- Network simplex in $\mathcal{O}(n^3 \log n)$; in practice, behaves in $\mathcal{O}(n^2)$ on random data
- Implementations: [Bonneel et al. 2009], Python Optimal Transport (POT)

Entropy-regularized optimal transport

$$\min_P \sum_i \sum_j c(x_i, y_j) P_{ij} - \varepsilon E(P)$$

Entropy, i.e., “blur”

+constraints, rewritten as :

$$\min_{P \in \mathcal{U}(f,g)} \langle C, P \rangle - \varepsilon E(P)$$

with $\mathcal{U}(f, g)$ matrices whose rows sum to f and columns to g

and $E(P) = -\sum P_{ij} (\log(P_{ij}) - 1)$ is the entropy, ε a small constant

[Cuturi 2013, Benamou et al. 2014]

Entropy-regularized optimal transport

$$\min_{P \in \mathcal{U}(f,g)} \langle C, P \rangle - \varepsilon E(P)$$

- Can be rewritten as a projection:

$$\min_{P \in \mathcal{U}(f,g)} KL(P, \xi)$$

where $\xi = \exp\left(-\frac{C}{\varepsilon}\right)$

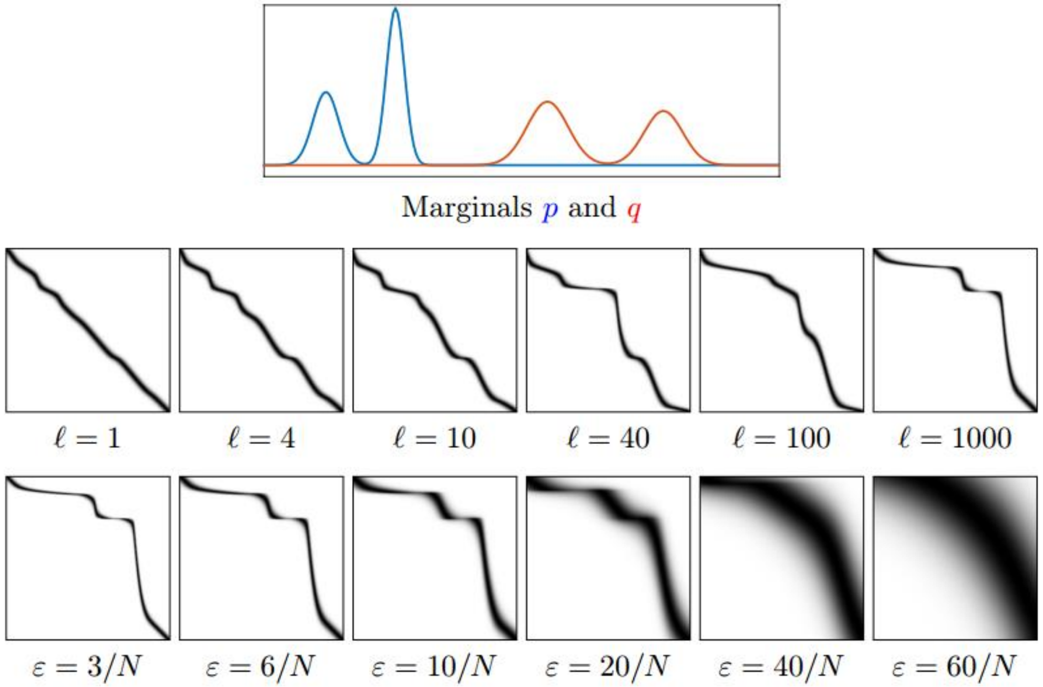
and $KL(P, \xi) = \sum P_{ij} \left(\log\left(\frac{P_{ij}}{\xi_{ij}}\right) - 1 \right)$ the Kullback-Leibler divergence

Entropy-regularized optimal transport

- Iteratively projecting on constraints: Sinkhorn algorithm (here, for $c(x,y) = \|x-y\|^2$)

- $u^{(\ell)} = \dfrac{f}{\text{Gaussian\_convolution}(v^{(\ell)})}$
- $v^{(\ell+1)} = \dfrac{g}{\text{Gaussian\_convolution}(u^{(\ell)})}$

- $P_{ij} = u_i^{(\infty)}\, e^{-c_{ij}/\varepsilon}\, v_j^{(\infty)}$



[Solomon 2015]

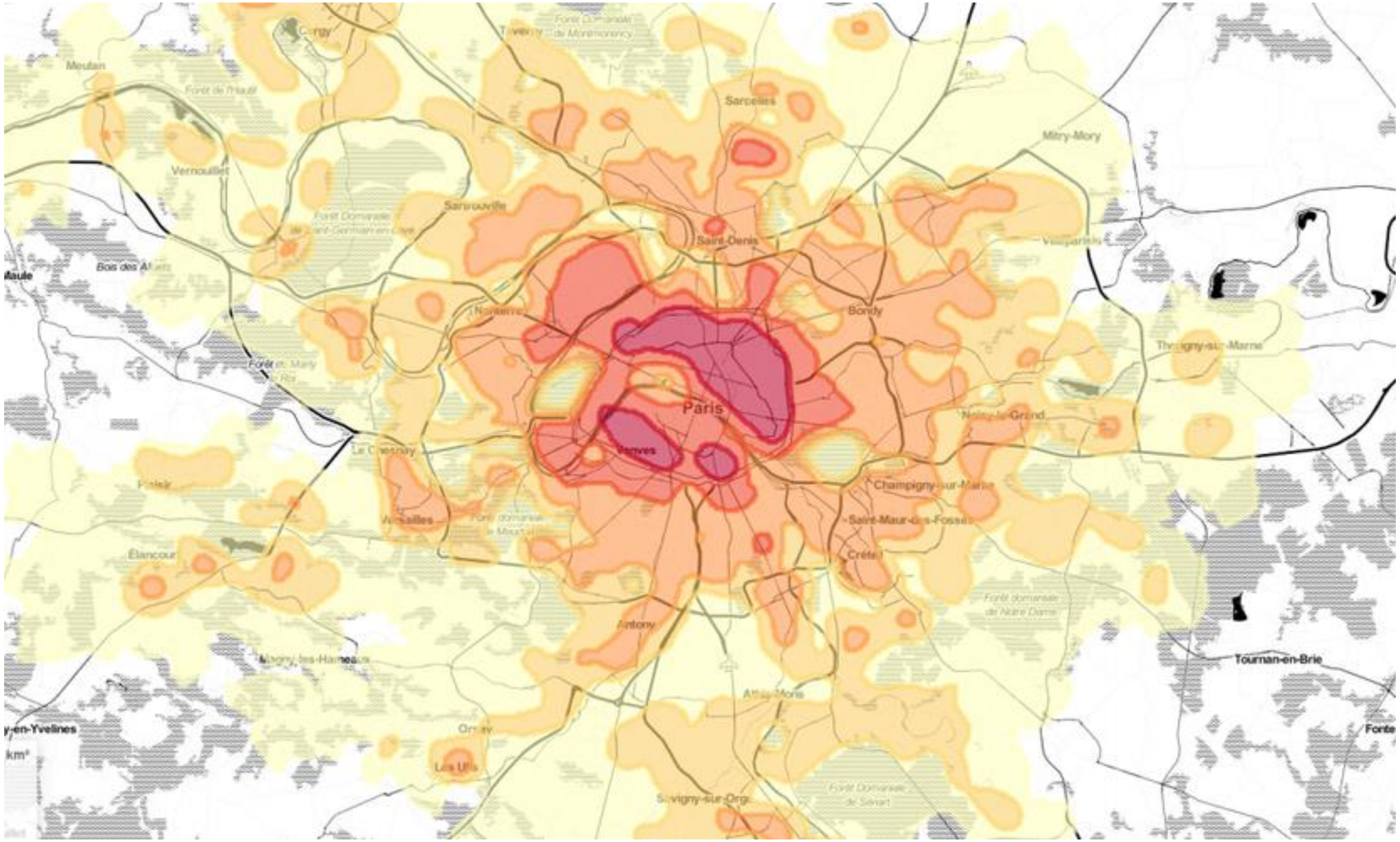
Entropy-regularized optimal transport

- Easily adapted to Wasserstein barycenters

[Solomon 2015]

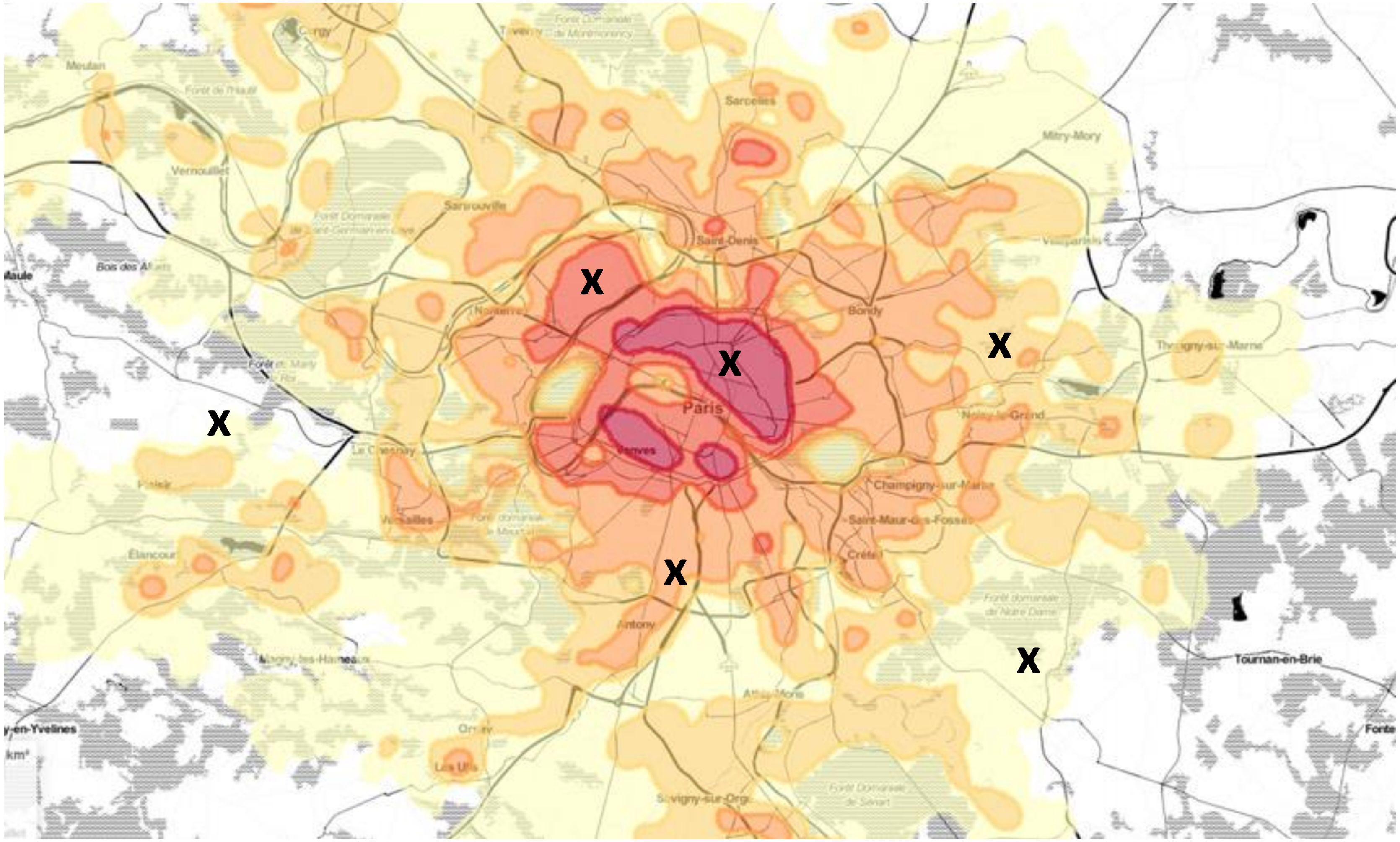
- Ideal for grids
- Works for scattered data and for other costs too
  - replace “Gaussian convolution” by matrix-vector multiplication
- Generalizes when distributions do not have the same mass
- However regularized OT between f and f is not 0
  - "Sinkhorn divergences" solves the issue by symmetrizing regularized OT [GeomLoss]

Semi-discrete Optimal Transport



Population density f

Semi-discrete Optimal Transport



Set of bakeries, factories, …?

Semi-discrete Optimal Transport

No constraint on production: population go to their nearest bakery/factory/… regardless of population density

Semi-discrete Optimal Transport

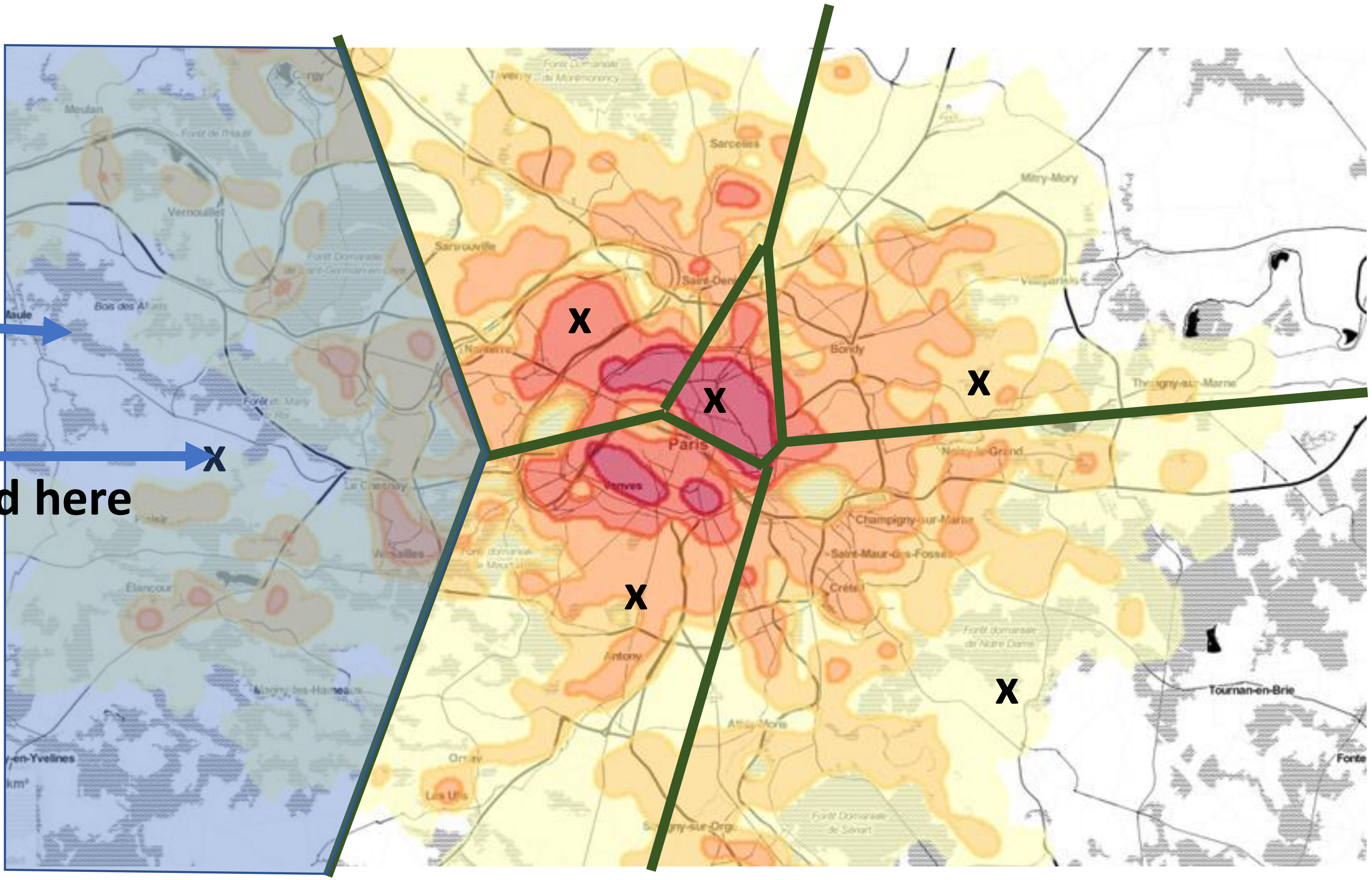
Limited production: population go to the nearest bakery/factory **with sufficient production**!

Semi-discrete Optimal Transport

(needs for) population here

=

quantity produced here



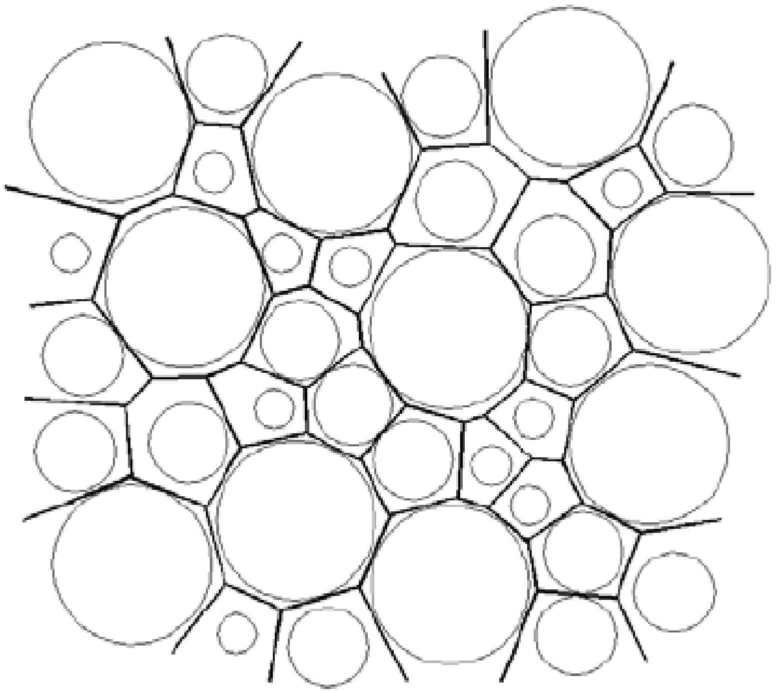
Limited production: population go to the nearest bakery/factory **with sufficient production**!

Power diagram (Laguerre diagram)

- A partition s.t. each point x is assigned to its closest site x_i with weight w_i

$$\|x - x_i\|^2 - w_i \leq \|x - x_j\|^2 - w_j \quad \forall j$$

Like a Voronoi diagram, but with weights $\{w_i\}$



Semi-discrete Optimal Transport

- Goal: find a set of weights $\{w_i\}$ such that $area(cell_i) = mass_i$
- Dual formulation: $\{w_i\}$ are (opposite of) selling prices
- Intuition: "bakery i increases its selling price $-w_i$ until sufficiently many clients leave, or decreases its selling price until sufficiently many clients come"
- If everybody increase its price by same amount: no change

Semi-discrete Optimal Transport

- Easy way: gradient ascent:
  - Compute power diagram of $\{x_i\}$ with weights $\{w_i\}$
  - $w_i \leftarrow w_i + \epsilon . \left(\lambda_i - \int_{x \in Cell_i} d\mu\,(x)\right)$
  - Iterate

- Better way:
  - Newton: second order minimization
  - Requires Hessian (easy)

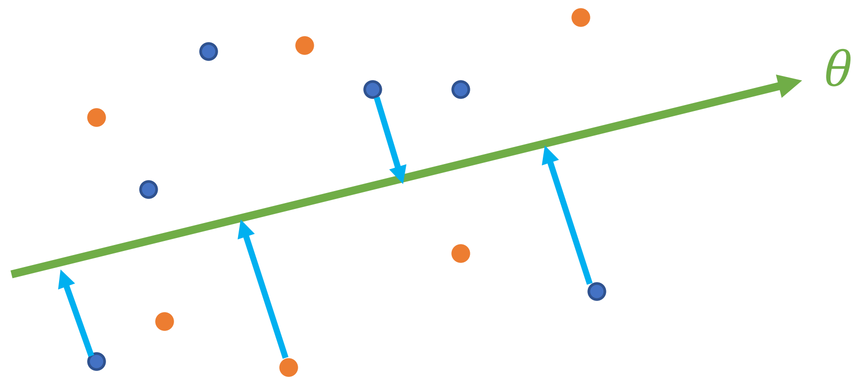
- Generalizes to non quadratic costs
  - E.g., Apollonius diagrams for $c(x, y) = \|x - y\|$

Area of cell I
= population in this region

Mass of Dirac i
= number of breads
produced by bakery i

Sliced Optimal Transport

- For discrete measures, consider 1-d projections



- $W_{sliced}(\mu,\nu)^2 = \int_{S^{d-1}} W(P_\theta(\mu), P_\theta(\nu))^2 d\theta$
- Monte Carlo estimation: $\int_{S^{d-1}} F(\theta)d\theta \approx \frac{1}{K}\sum_k F(\theta_k)$ with $\theta_k \sim \mathcal{U}(S^{d-1})$

Sliced Optimal Transport

- Not exactly like OT
- Can be used to compute barycenters by gradient descent (easy)
- Generalizes to continuous distributions
  - Requires Radon transform
- Generalizes to point sets with different numbers of Diracs (harder)

Dynamical formulation

- Writing conservation PDE of a quantity f advected by vector field v

$$\frac{\partial f_t}{\partial t} + \mathrm{div}(f_t\, v) = 0$$

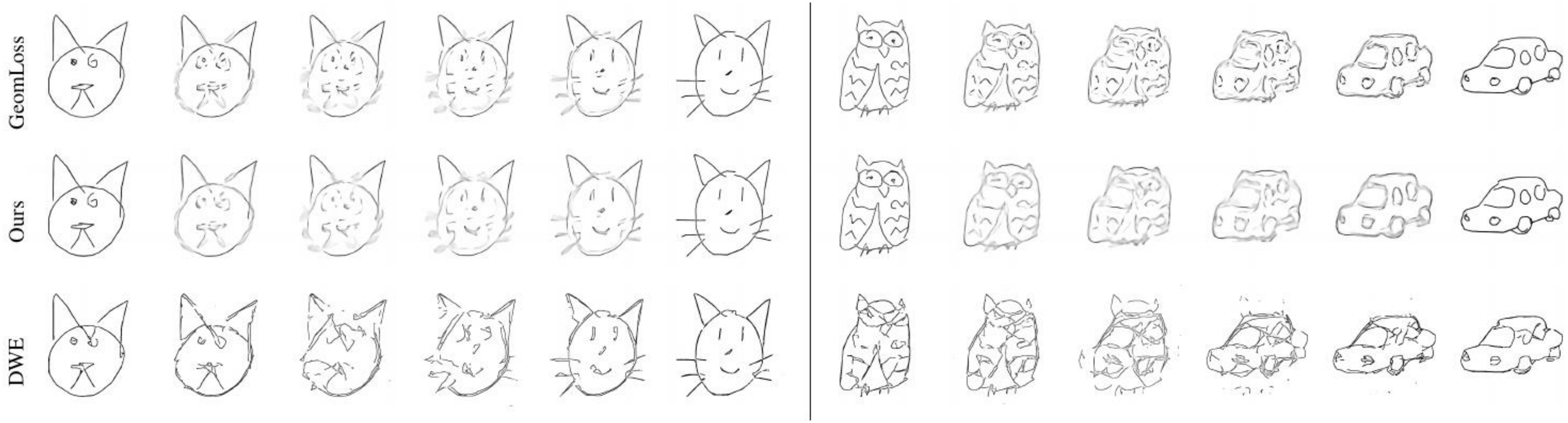
For $c(x, y) = \|x - y\|^2$, $f_0 = f$ and $f_1 = g$, find $v(x, t)$ minimizing:

$$\min_{v, f_t} \int_0^1 \int_X \|v(x, t)\|^2\, f_t(x, t)\, \mathrm{d}x\, \mathrm{d}t$$

More complex solvers, generally for low-dimensional grids.

Other solvers

- Deep learning
  - Compute embedding [Courty 2018] : computing barycenters and distances
  - Learning barycenters [Lacombe 2022] : only barycenters

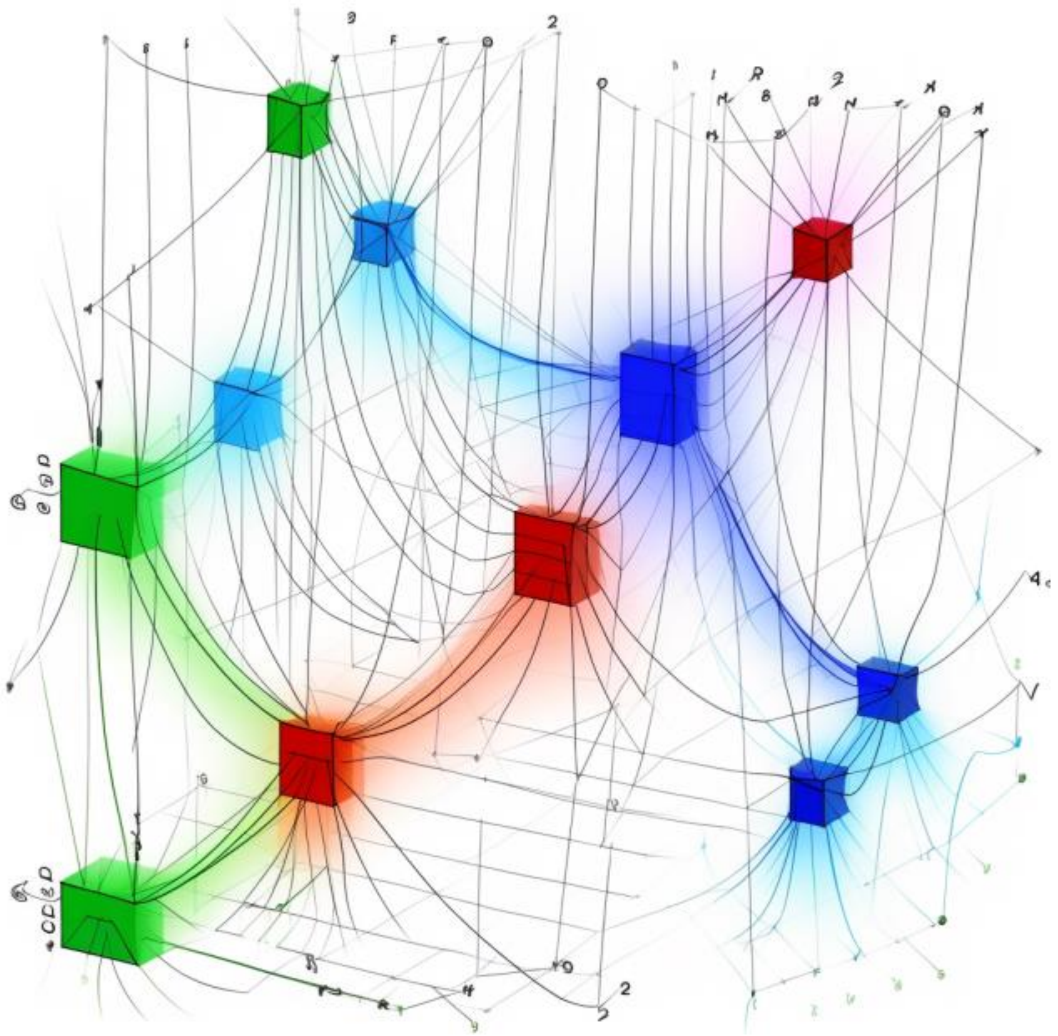


Common uses of optimal transport

- As a distance between distributions
  - In neural network or other fitting losses
  - To best approximate a continuous distribution with discrete primitives (e.g., stippling), to fit surfaces to point clouds, for retrieval...
- As a way to match discrete distributions
  - E.g., matching pixel colors
- As a way to interpolate between distributions
  - To produce nice "warps" (e.g., BRDF highlights)
- As a way to enforce mass preservation
  - E.g., for surface parameterization (area preserving maps), fluid simulation (incompressibility), computational optics (redirecting light distributions)

Applications

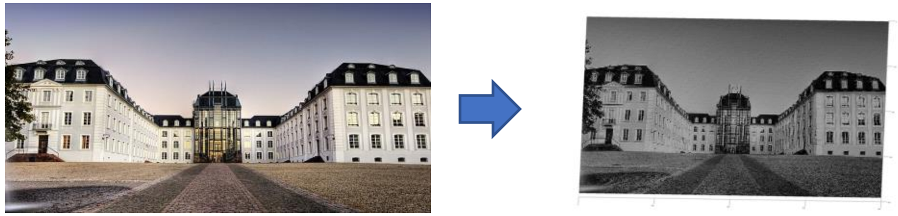
Applications to Image Processing



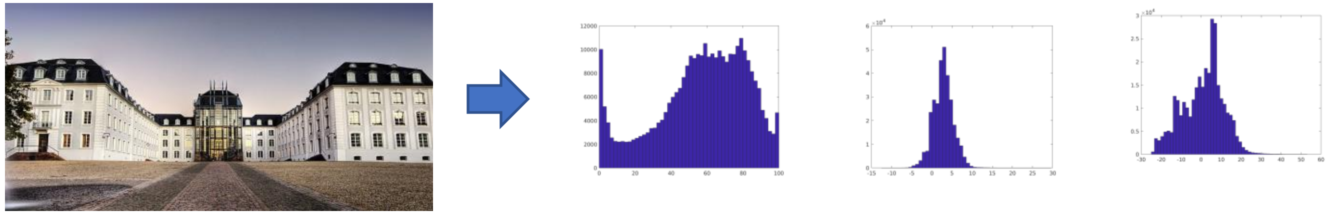
"an illustration of the mathematical theory of Monge's optimal transport applied to image processing"

Applications to Image Processing

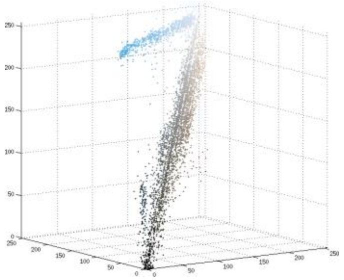
- Image gray values as a distribution



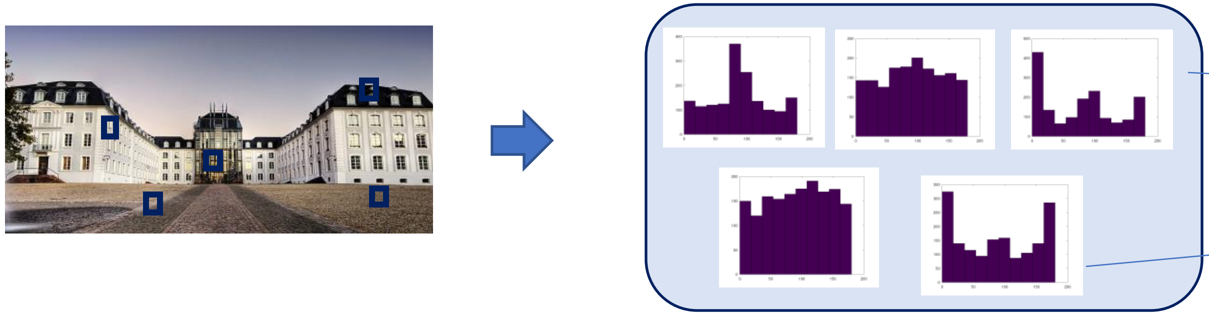
- Image color histogram as probability distributions

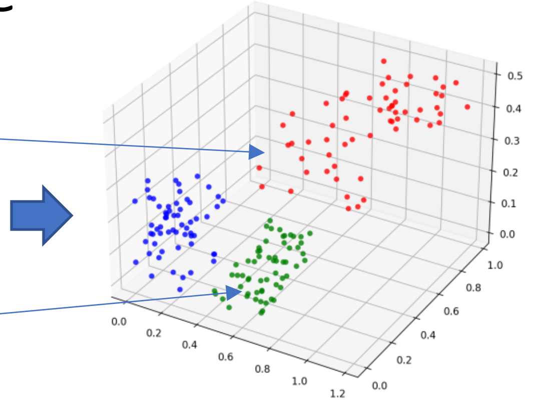


or



- … Or as a distribution of features in feature space



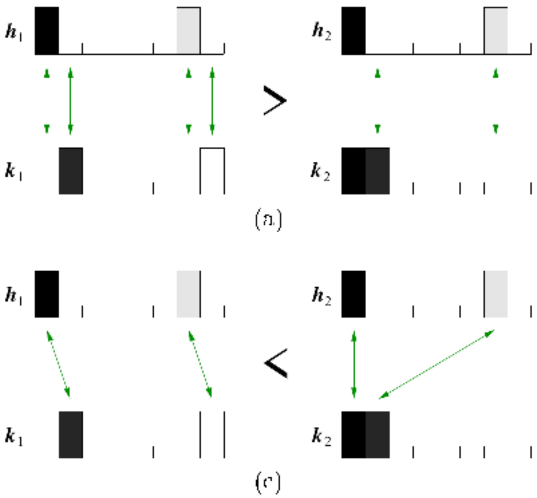


Applications to Image Processing

Image retrieval

[Rubner 2000]

- Features = Image colors in CIE-Lab [Rubner 2000]
  - Clustering in color space
- Features = SIFT descriptors, shape contexts, spin images [Ling 07]
- Improving solvers:
  - Unbalanced case [Pele 08]
  - Thresholding ground distances [Pele08,09]



[Rubner 2000]

Applications to Image Processing

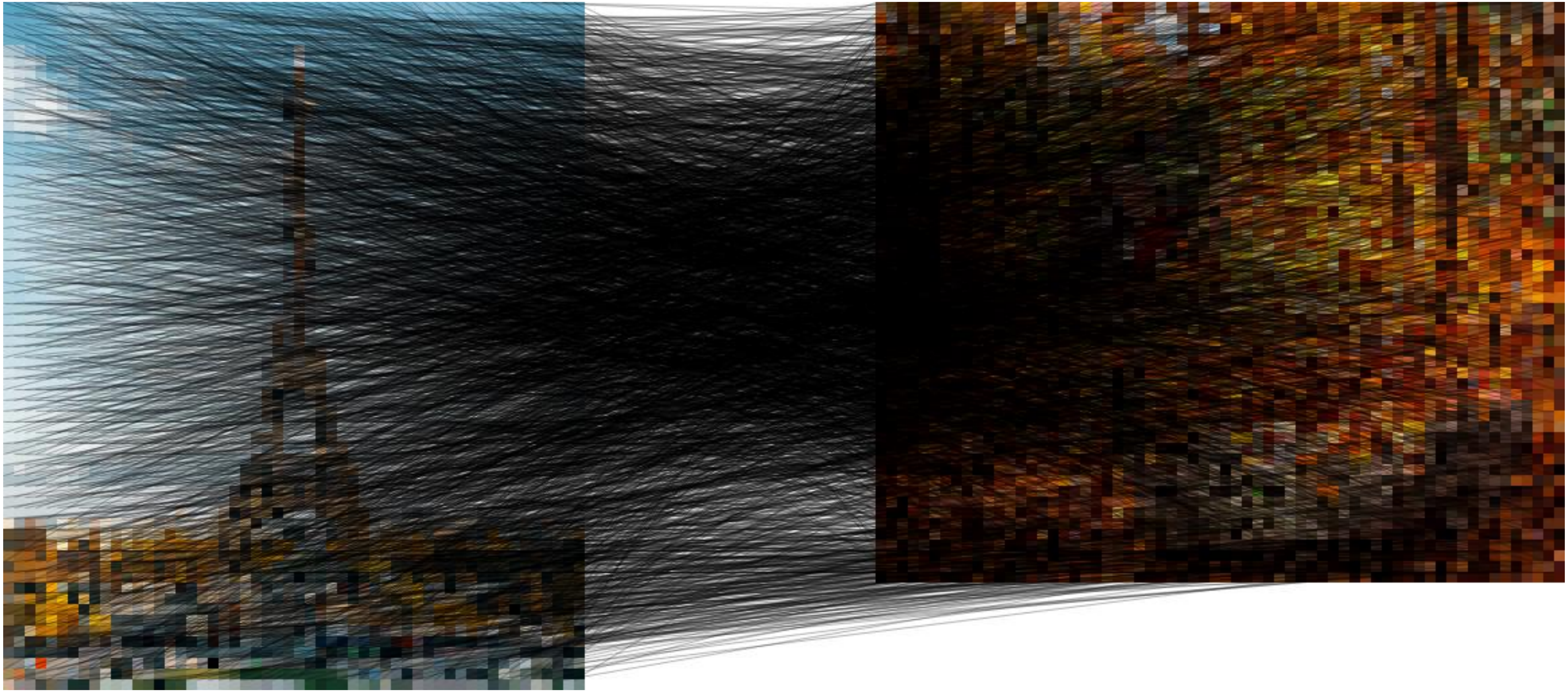
Color Grading

Input photo

Target style

Input photo

Target style



Input photo

Target style

Input photo

Target style

Applications to Image Processing

Color Grading

- Every Flavor of OT:
  - Sliced OT [Pitié 05, Bonneel 15]
  - Entropy regularized OT [Solomon 15]
  - Simplex solver on color clusters [Morovic 03, Bonneel 11, Rabin 15 ...]
  - Gaussian distributions [Pitié 07]
- Color distribution in CIE-Lab = 1D OT for luminance, 2D for chrominance [Bonneel 13, Solomon 15]
- Quantization artifacts [Rabin 10, Rabin 14, Chizat 18, Bonneel 19]

[Rabin 10]

Applications to Image Processing

Color Grading



Photoshop

Gaussian OT
[Pitié 2007a}

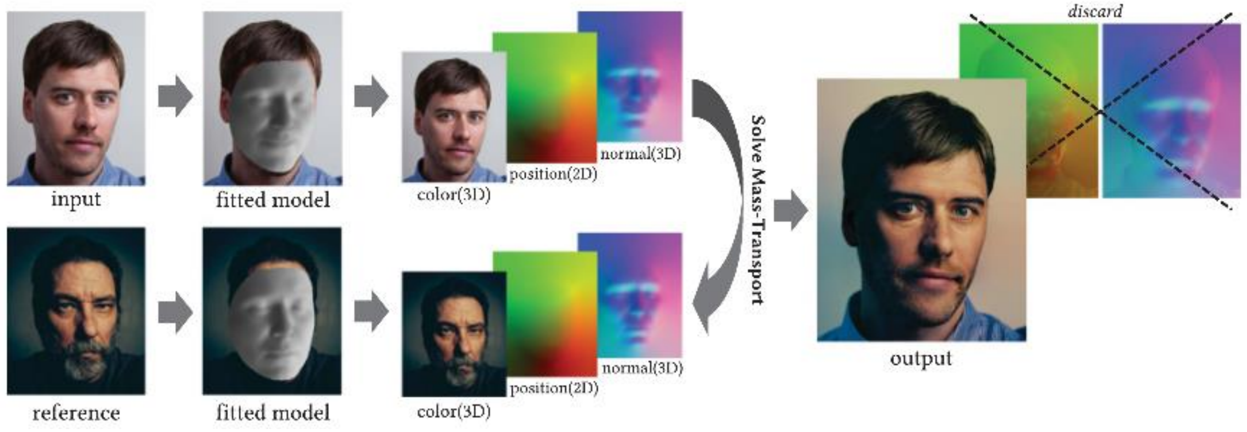
Continuous
sliced OT
[Pitié 2007 b]

Discrete
sliced OT
[Bonneel 15]

Applications to Image Processing

Style Transfer

- Style = Gram matrix of the CNN features [Gatys 2016], matching done through OT [Kolkin 19]
- Semi-discrete OT between synthesized patches and sample patches [Galerne 18], with a multiscale approximation [Leclaire 21]
- Portrait relighting: OT between histograms of normals, positions, colors. (Monge formulation)

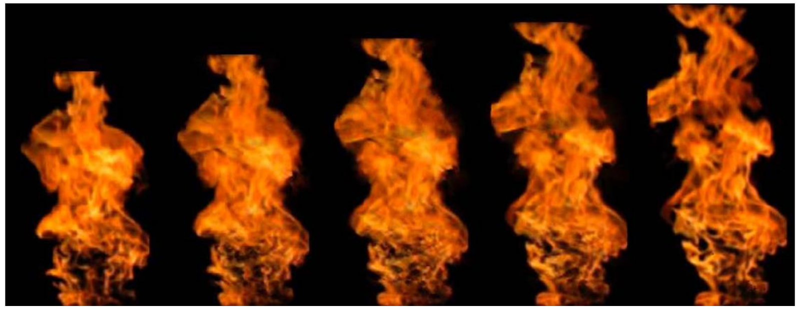


[Shu 17]

Applications to Image Processing

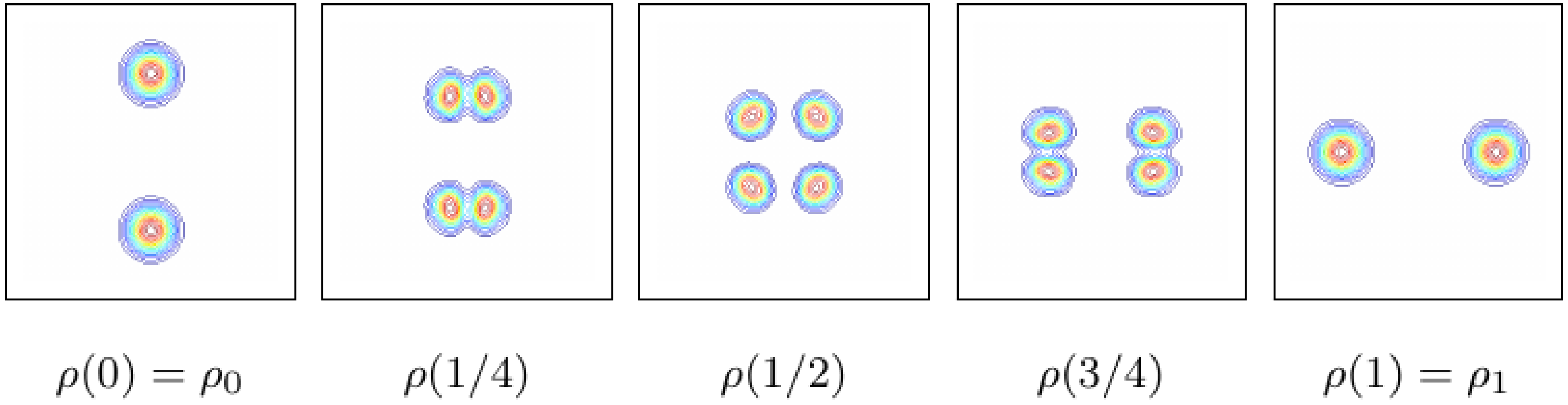
Image and Shape Interpolation

- Low-structure data (smoke, fluids)



[Haker 04]

- Known issue: no structure preservation



[Hug 2015]

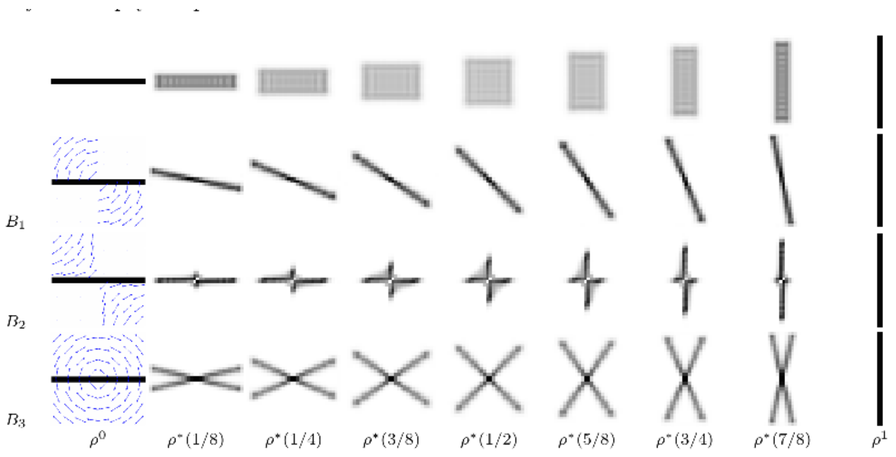
Applications to Image Processing

Image and Shape Interpolation

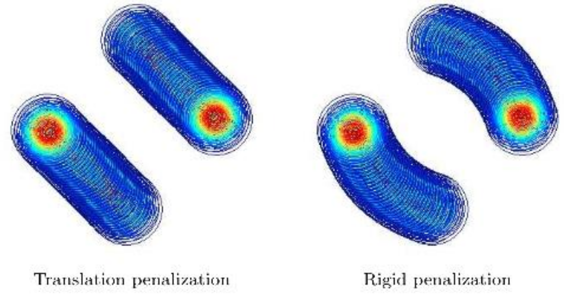
- Adding physics constraints [Hug 2015]
  - Anisotropic force fields
  - Penalization on the velocity

$$K(\rho, m, v) = \frac{1}{2}\int_{\Omega}\int_{0}^{1} ||m - \rho v||^2 \,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

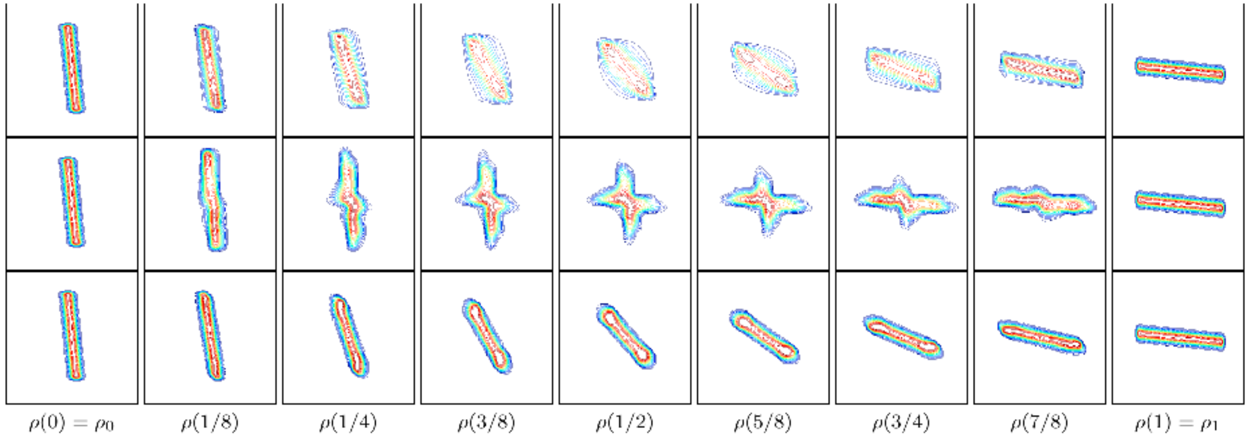
$$R(v) = \frac{1}{2}\int_0^1 ||(\nabla_{\mathbf{x}} v(t,\cdot) + (\nabla_{\mathbf{x}} v(t,\cdot))^T)/2||^2_{L^2(\Omega)}\,\mathrm{d}t$$



[Hug 2015]



[Hug 2015]

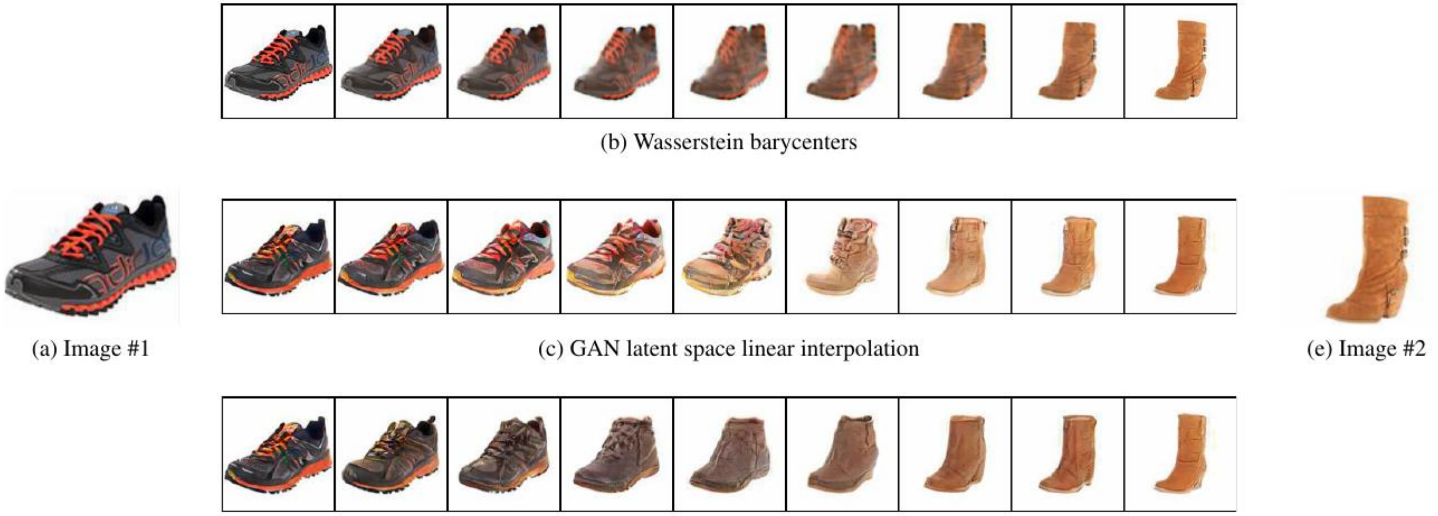


[Hug 2015]

Applications to Image Processing

Image and Shape Interpolation

- Constrain the barycenters to follow an image prior (GAN prior) [Simon 20]



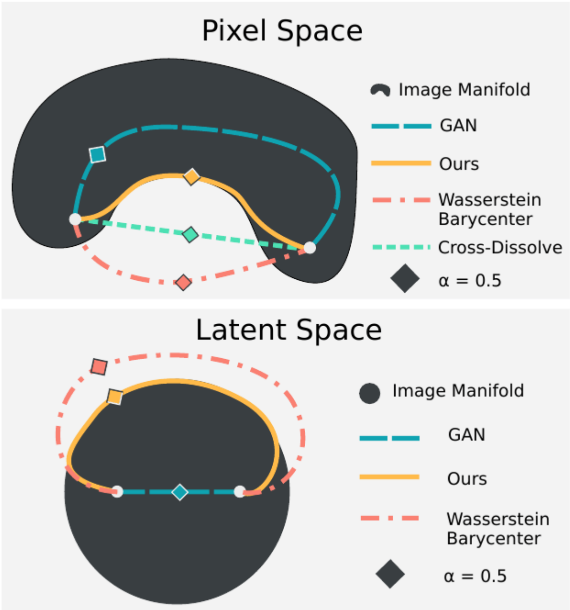
(a) Image #1

(b) Wasserstein barycenters

(c) GAN latent space linear interpolation

(d) Ours – WBP with a GAN as an image prior

(e) Image #2



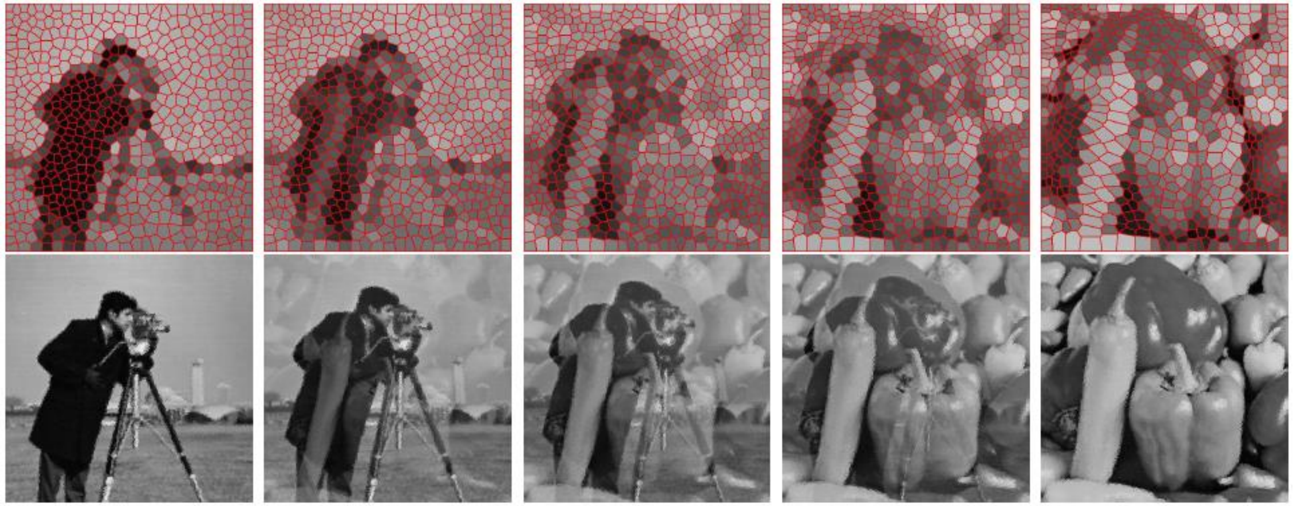
[Simon 20]

$$p_\alpha = \begin{cases} \underset{q \in \Sigma_n, r}{\arg\min} \; (1-\alpha)\mathcal{W}_2^2(p_1, q) + \alpha \mathcal{W}_2^2(p_2, q) \\ \text{s.t.} \quad r \in \mathcal{M}, \; r = q. \end{cases}$$

Applications to Image Processing

Image and Shape Interpolation

- OT is not efficient for *general* images, but can work for specific cases.
- Used to illustrate OT algorithms.

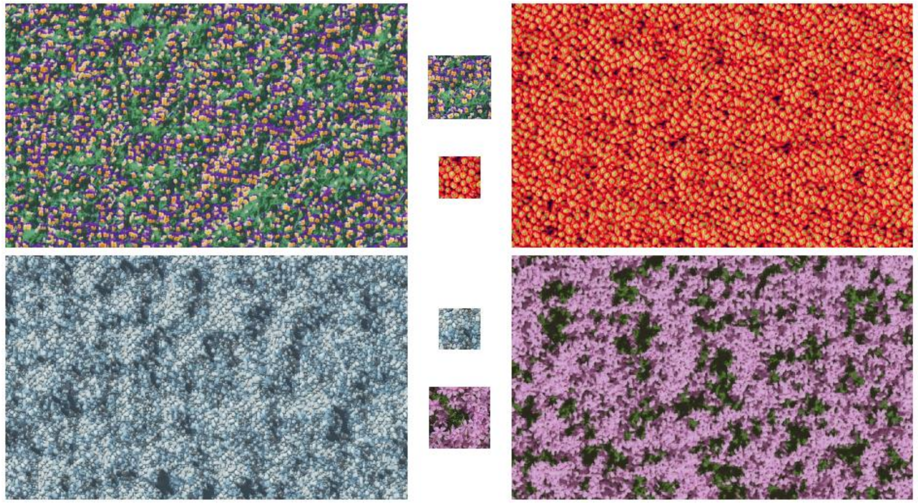


[Mérigot 11]

Applications to Image Processing

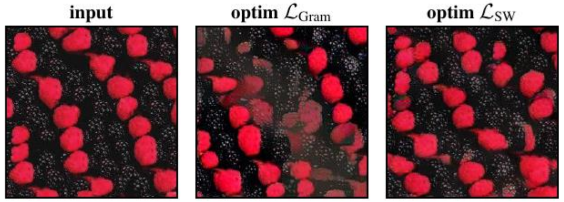
Texture synthesis

- Heeger-Bergen [1995]: steerable pyramid coefficients, iterative synthesis by sub-band matching using 1d OT.
- Tartavel [2016] replaces the matching with sliced OT.
- Spot Noise texture [Galerne 2018]
  - Semi-discrete OT between 3x3 spot noise patches and exemplar patches
  - Aggregation by averaging
  - [Leclaire 21]: faster approximation



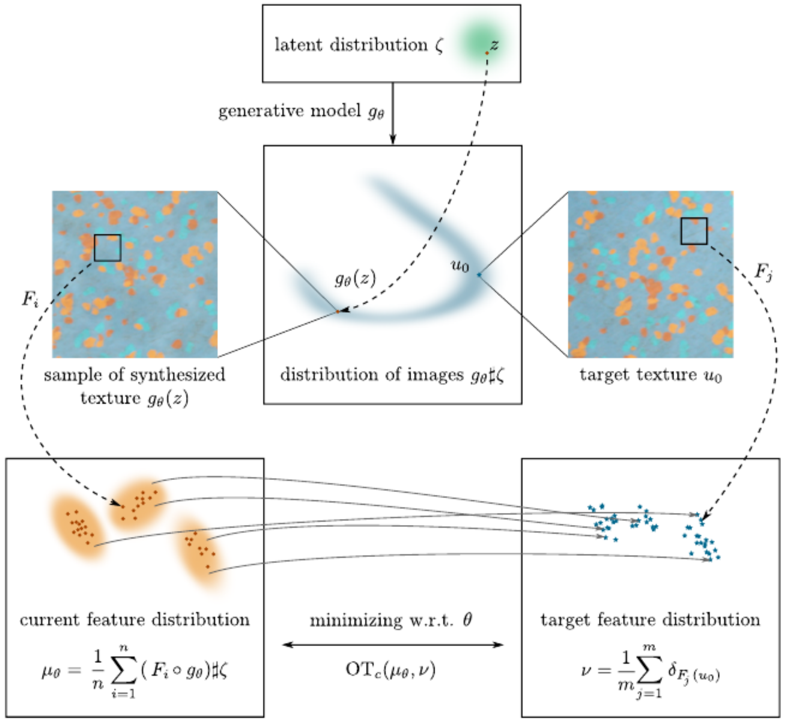
[Galerne 2018]

Applications to Image Processing
Texture synthesis



[Heitz 21]

- OT as a loss for Deep Learning texture synthesis
  - Replace the Gram matrix loss with a sliced Wasserstein loss between CNN features [Heitz 21]
  - Semi-discrete OT between texture features synthesized by a generator and target texture features [Houdard 22]
- OT for preprocessing textures
  - Blending patch colors using Gaussian distribution blending closed form formula. Ensure Gaussian distribution by computing an OT (LP). [Heitz 18]



[Houdard 22]

Applications to Image Processing

Natural images generation

- Generate images that look like an input set of images
- Generative Adversarial Networks (GAN): replace the Jensen-Shannon divergence with a Wasserstein-1 distance [Arjovsky 17]
  - Discriminator becomes a *Critic*

$$\mathcal{L}_{WGAN}(G, C) = \mathbb{E}_{x\sim\mu_G}[C(x)] - \mathbb{E}_{x\sim\mu_{ref}}[C(x)]$$

  - By weight clipping, optimal C is such that *:*

$$\mathcal{L}_{WGAN}(G, C) = W_1(\mu_{ref}, \mu_G)$$

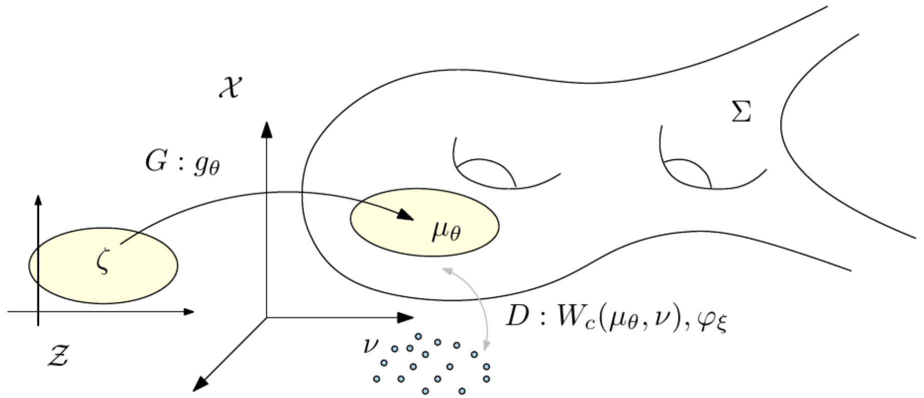
  - Avoids vanishing gradients and mode collapse
  - Better training by replacing weight clipping with a gradient penalty [Gulrajani 17]

[Gulrajani 17]

Applications to Image Processing

Natural images generation

- Wasserstein GAN with semi-discrete OT [Lei 19]
  - G computes a transport map from latent space to data manifold
  - D computes a Kantorovich potential
  - If the cost is strictly convex, G writes in closed form from optimal D

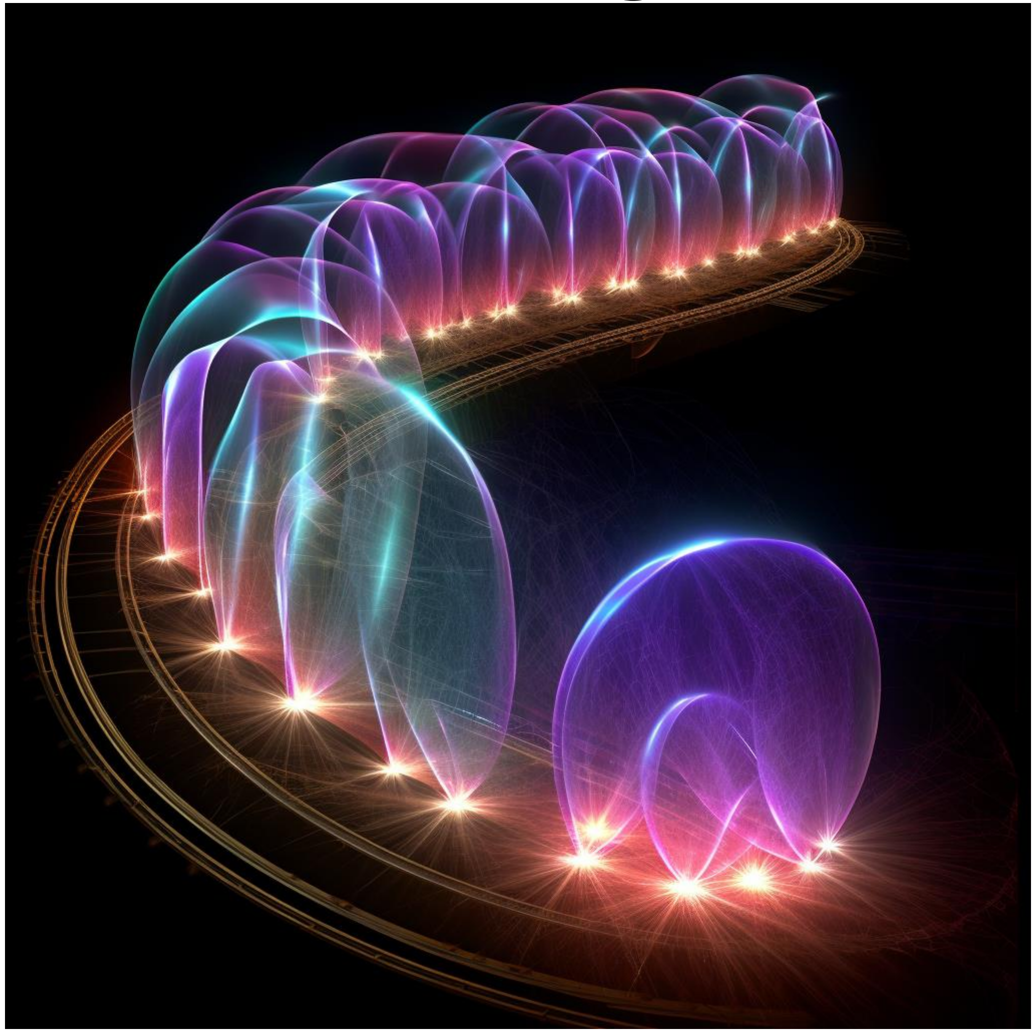


[Lei 2019]

- GAN now superseded by denoising diffusion models [Ho 2020]
  - Bounds on the W1 distance between the output distribution and the target distribution [De Bortoli 22]

Generative modeling = synthesizing a distribution mimicking an input discrete distribution. OT is a natural tool in this context.

Applications to Rendering



"a realistic illustration of the mathematical theory of Monge-Kantorovich's optimal mass transport applied to light transport simulation for rendering"

Applications to Rendering

Image stippling & Monte Carlo sampling

- Image stippling
  - Considers grayscale image = probability density function
  - For artistic purpose, and for printing with black ink droplets
  - Extends to colors

"BNOT" (Blue Noise Through Optimal Transport), [De Goes 2012]

Applications to Rendering

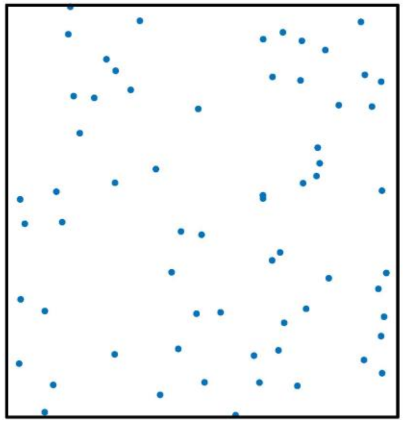
Image stippling & Monte Carlo sampling

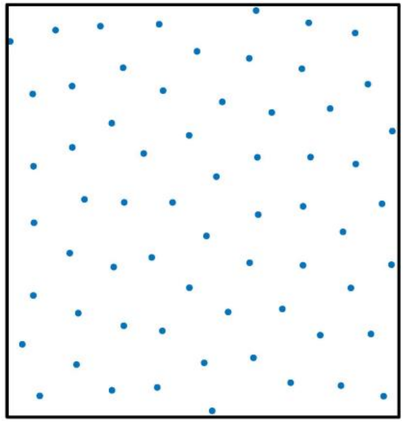
- Monte Carlo sampling
  - Numerical integral evaluation:

$$\int_{\Omega} f(x)dx \approx \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}$$

where x_i follow law of probability density function p

  - When $p(x) = 1$, converges faster when $\{x_i\}_i$ are not i.i.d. but blue noise





Applications to Rendering

Image stippling & Monte Carlo sampling

- In both cases
  - Problem = finding samples that approximate function best
  - Typical approach Lloyd-like:
    - Semi-discrete OT (exact, entropic or sliced)
    - Center samples
    - Repeat.
- Differences:
  - Image stippling is 2-d, rendering n-d
  - Image stippling: non-uniform density ; rendering: considers uniform case
  - For rendering: may enforce uniformity in projections
  - Image stippling: may interleave colors

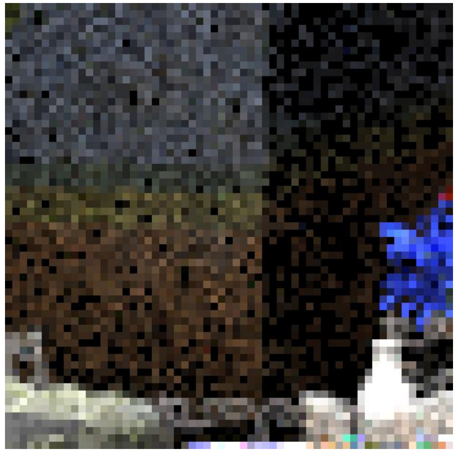
Applications to Rendering

Image stippling & Monte Carlo sampling

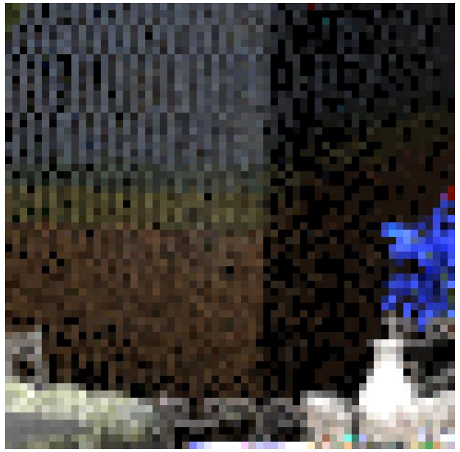
- For rendering:

ground truth

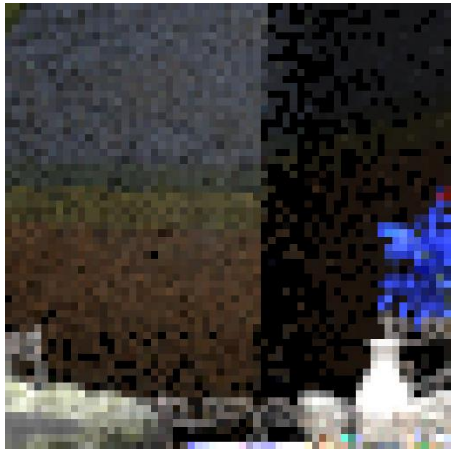
ground truth



random



Sobol'

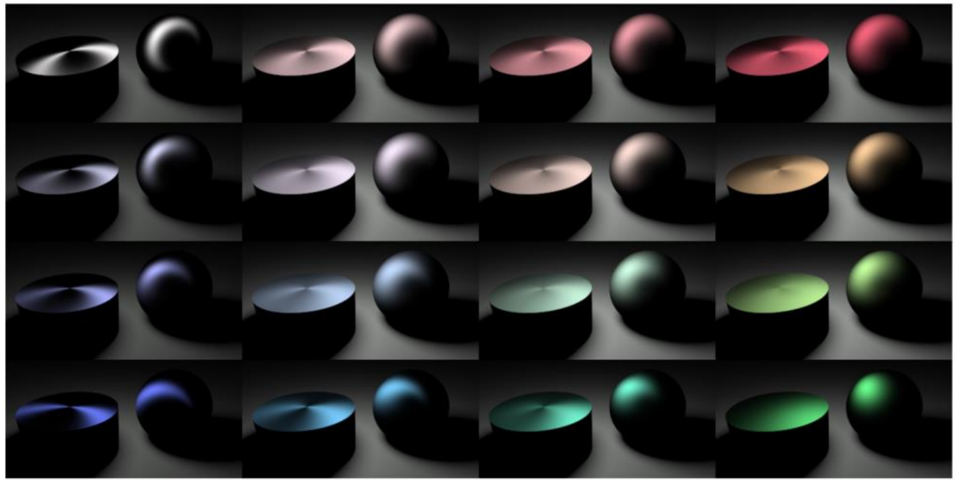


Sliced OT [Paulin 2020]

- Stippling: generalization to curves [Lebrat 2019]

Applications to Rendering

Reflectance manipulation

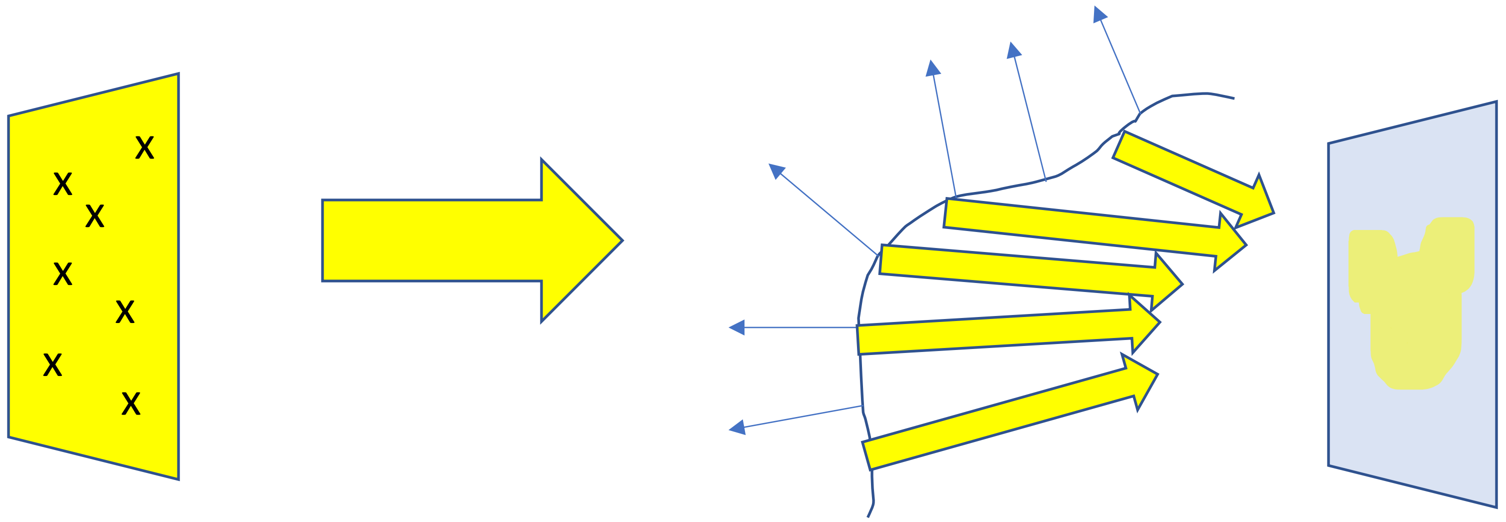


- Interpolating between different BRDFs
  - 2 BRDFs [Bonneel 2009]
  - n BRDFs [Solomon 2015]
  - n car paint BTFs [Golla and Klein 2018] (flakes interpolation)
- Densifying sparse BRDFs
  - By interpolating between incident directions [Ward 2014]
  - By projecting on dataset of dense BRDFs [Bonneel 2016]
- OT between BRDF degradations correlate with perception [Lavoué 2021]

Applications to Rendering

Computational optics and imaging

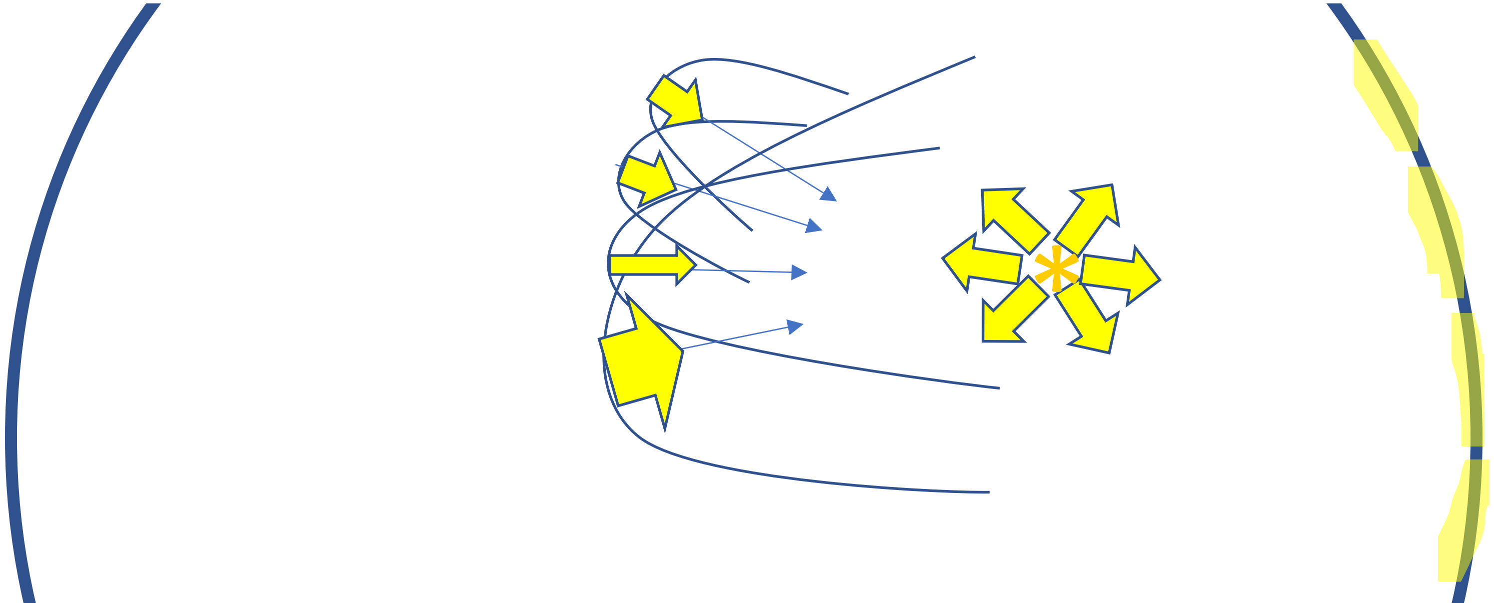
- Transporting light distribution to given target (mirrors or refractors)
  - Semi-discrete OT with discretized flat light [Schwartzburg 2014]
  - With obtained assignment, recover normal with Snell law
  - Then, surface reconstruction from normals



Applications to Rendering

Computational optics and imaging

- Transporting light distribution to given target (mirrors or refractors)
  - Linear program OT with paraboloid normals and target envmap [Andre 2015]
    - Assumes a point light, and surface made of parabolas pointing towards it
    - Goal: find parabola focal distances, spreading light differently towards each direction
    - Uses $c(x,y) = -\log(1 - x.y)$ with incoming light direction x and paraboloid normal y
    - Dual of OT linear program gives focal distances

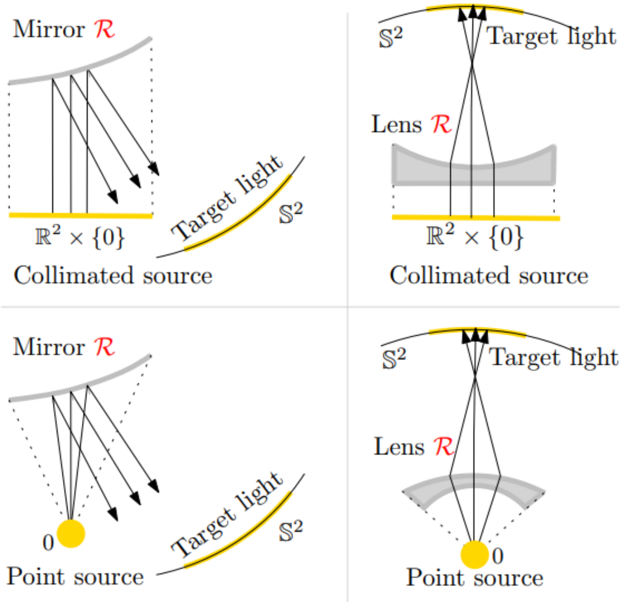


Applications to Rendering

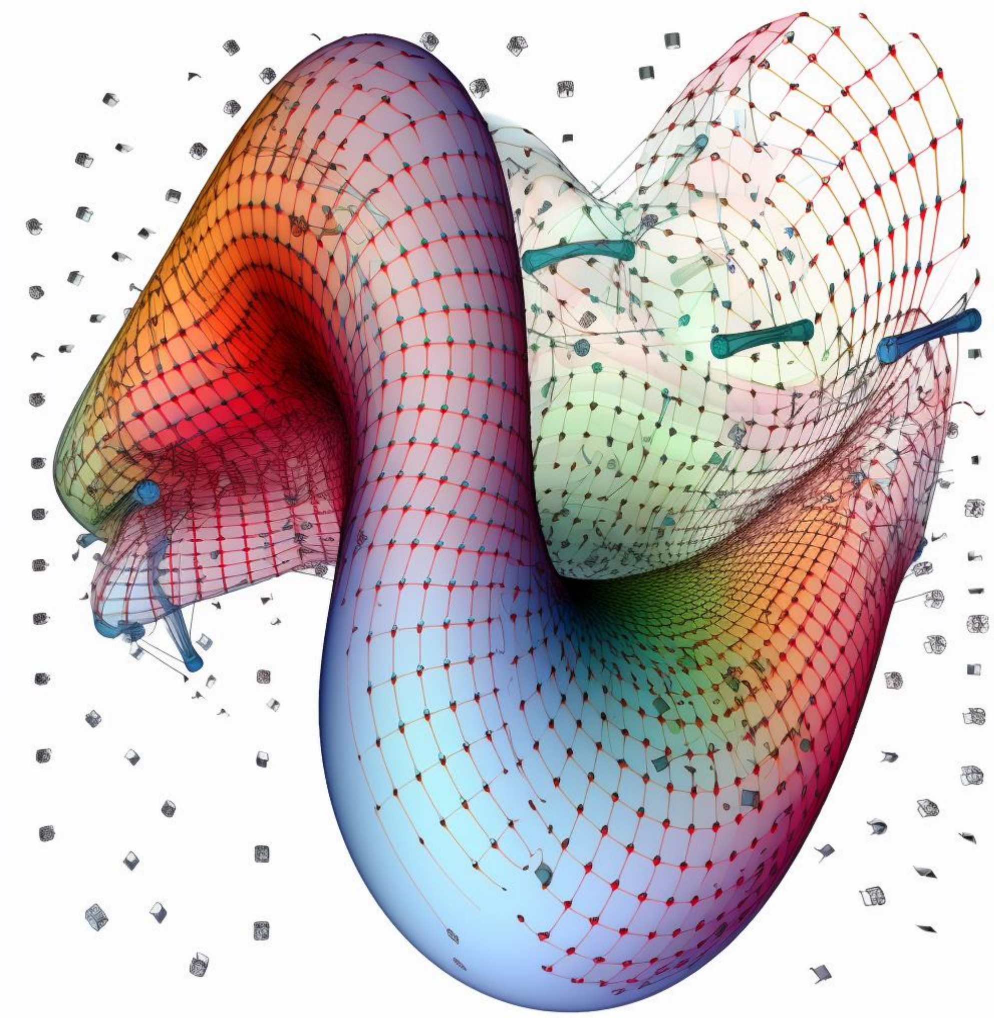
Computational optics and imaging

- Transporting light distribution to given target (mirrors or refractors)
  - Semi-discrete OT for general problems [Meyron 2018]
    - Refractor/reflector, near/far-field, collimated/punctual light source, concave/convex surfaces… in the same framework
    - Amount to standard semi-discrete OT with $c(x, y) = ||x - y||^2$

    (up to changes of variables)
    - Uses a set of parabolas



Applications to Geometry Processing

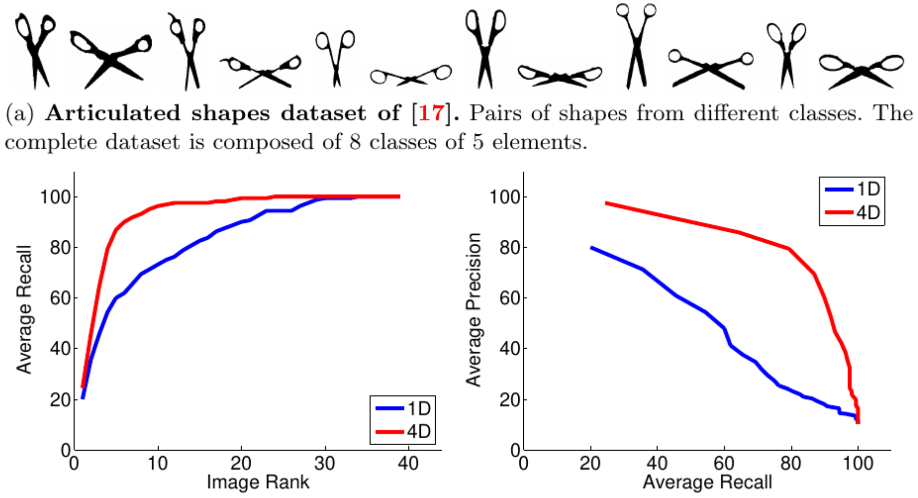


"a realistic illustration of the mathematical theory of Monge-Kantorovich's optimal mass transport applied to geometry processing"

Applications to Geometry Processing

Shape comparison and retrieval

- Shapes as a bag of descriptor
  - Descriptor: Geodesic distances to a set of anchor points [Rabin 10]
  - Sliced Wasserstein 2 as a distance metric between the descriptors.

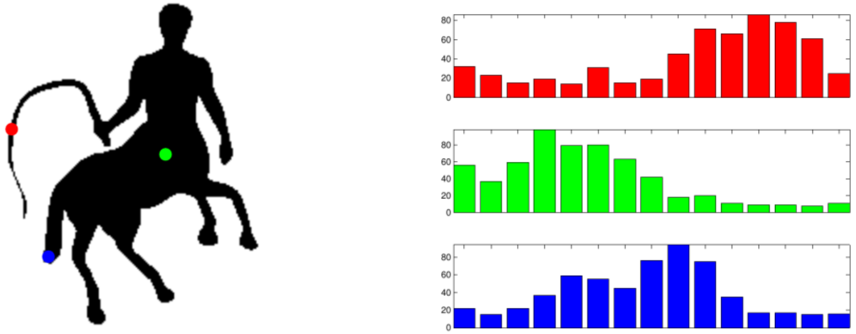


(a) **Articulated shapes dataset of [17].** Pairs of shapes from different classes. The complete dataset is composed of 8 classes of 5 elements.

(b) Recall vs Image Rank.

(c) Average Precision-Recall.

[Rabin 10]



Sampling locations $x_i \in \mathcal{S}$

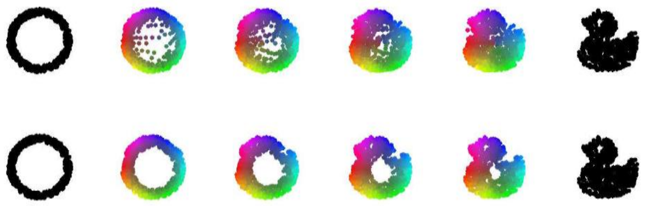
Histograms of $\{d_\Omega(x_i, y)\}_{y \in \mathcal{E}}$

[Rabin 10]

Applications to Geometry Processing

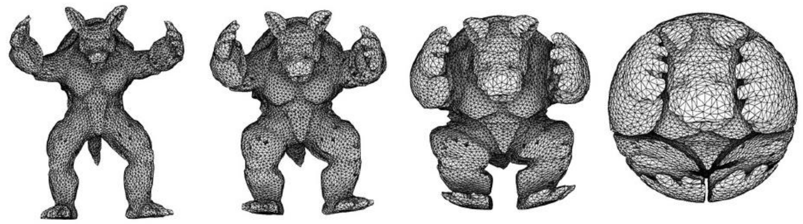
Shape interpolation

- 2D or 3D shapes for which feature matching is impossible
- Discretizing the volume as point sets+ sliced Wasserstein [Rabin 12]
  - Points sampled in the volume



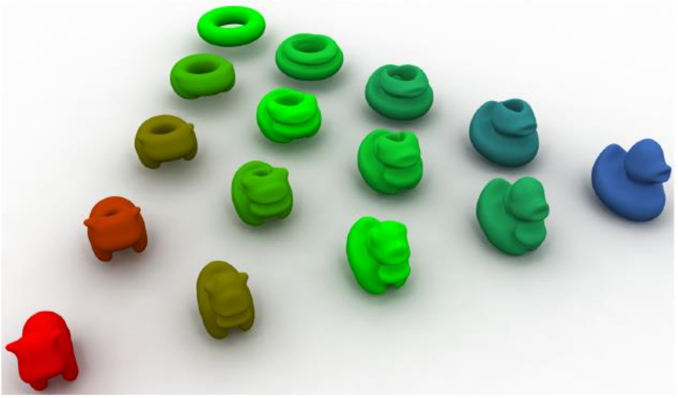
[Rabin 12]

- Semi-discrete OT to interpolate between two tetrahedral meshes [Lévy 15]
  - Recovers a topology throughout the interpolation



[Lévy 15]

- Alternative to sampling: shapes = indicator functions discretized on grids (2d or 3d)
  - Fast convolutional Wasserstein Distances [Solomon 15]



[Solomon 15]

OT-based shape interpolation: well defined but still some topological issues, mass may split, connected components appear and disappear.

Applications to Geometry Processing

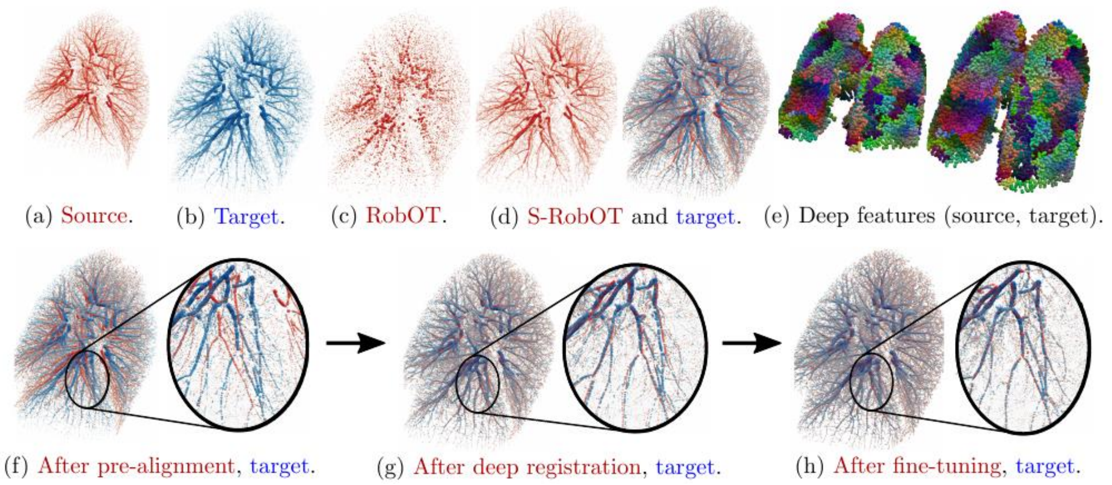
Shape registration (point cloud to point cloud)

- Matching partial point clouds using sliced Optimal Transportation [Bonneel 19]
  - Injectivity is guaranteed
  - ICP variant

- Entropy-regularized unbalanced optimal transport [Shen 21] building on [Chizat 18]

$$\mathrm{OT}_{\sigma,\tau}(\mathrm{A},\mathrm{B}) = \min_{(\pi_{i,j}) \in \mathbb{R}_{\geqslant 0}^{\mathrm{N}\times\mathrm{M}}} \sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{\mathrm{M}} \pi_{i,j} \cdot \tfrac{1}{2} \|p_i - q_j\|_{\mathbb{R}^{\mathrm{D}}}^2$$

$$+ \underbrace{\sigma^2 \, \mathrm{KL}\big(\pi_{i,j} \,||\, \alpha_i \otimes \beta_j\big)}_{\text{Entropic blur at scale } \sigma.} + \underbrace{\tau^2 \, \mathrm{KL}\big(\textstyle\sum_j \pi_{i,j} \,||\, \alpha_i\big)}_{\pi \text{ should match A...}} + \underbrace{\tau^2 \, \mathrm{KL}\big(\textstyle\sum_i \pi_{i,j} \,||\, \beta_j\big)}_{\text{...onto B.}} ,$$

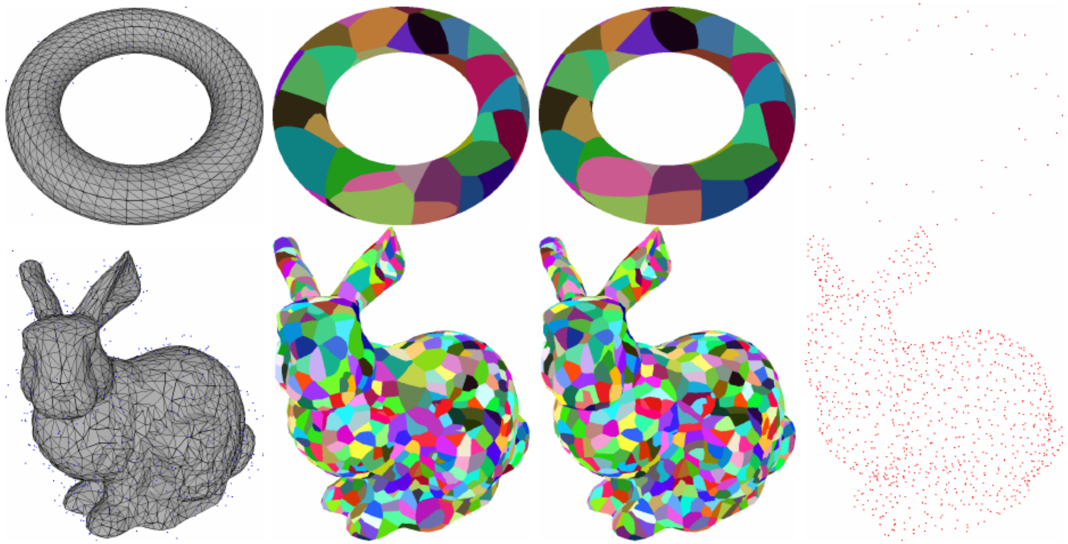


[Shen 21]

Applications to Geometry Processing

Shape registration (point cloud to mesh)

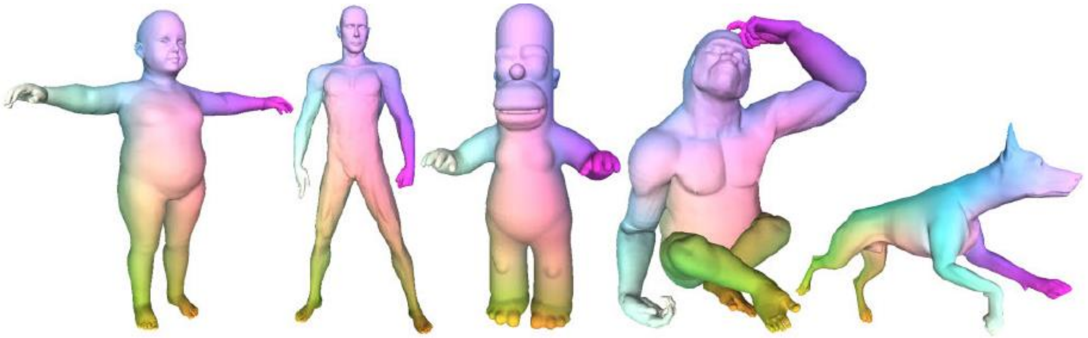
- Semi-discrete Optimal Transport [Mérigot 18]
  - Restricted power diagram on the mesh
  - Weights optimized by a damped Newton
- OT-ICP
  - Drops the assignment step and replace it by semi-discrete OT
  - Each point is assigned to the barycenter of its power cell
  - Faster empirical convergence/far away poses
- Also for remeshing:
  - Dual of the computed power diagram yields remeshing of the mesh.
  - Allows to adapt locally the mesh density



[Mérigot 2018]

Applications to Geometry Processing

Shape registration (mesh to mesh)



[Mandad17]

- Variance minimizing transport plan [Mandad17]
  - Surface discretized as point sets
  - Transport plan that minimizes the variance between the neighborhoods.
- Deep Shells [Eisenberger20b]
  - Replaces matching step of Smooth Shells with entropy-regularized OT
  - All steps become differentiable

Source

Smooth Shells

Deep Shells

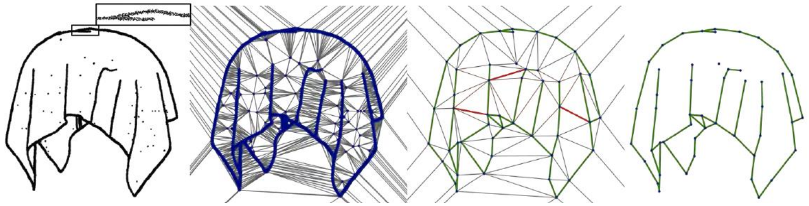
Optimal transport for shape registration: needs further constraints or only used as a metric

Applications to Geometry Processing

Shape reconstruction

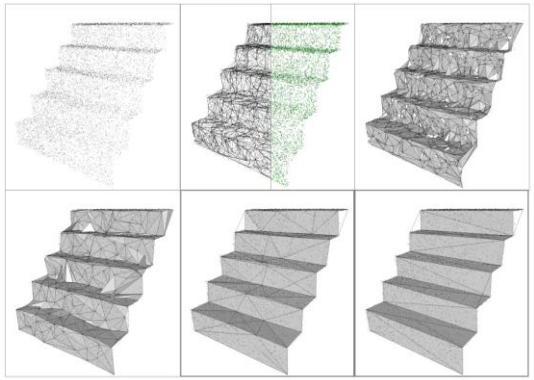
- Idea: measure the distance between a mesh and a input point cloud, optimize the mesh to lower the OT cost between mesh and point cloud.

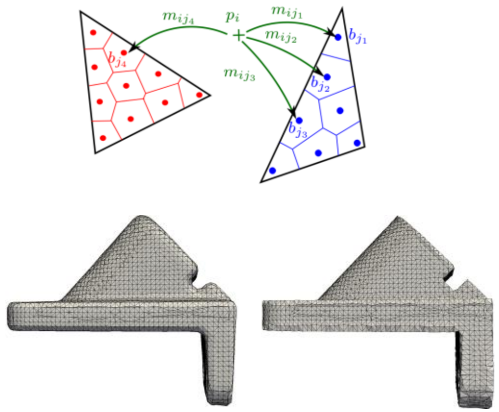
- 2D sketches [De Goes 11]
  - Assignment to nearest mesh simplex
  - OT cost computed in closed form on edges



[De Goes 11]

- 3D surfaces [Digne 14]
  - Local OT solve to approximate the global solve
  - Mass can split
  - Application also to sharp feature recovery





[Digne 14]

Applications to Geometry Processing

Shape Parameterization

- Parameterization = Unfolding a mesh on a planar/spherical domain

- Conformal mapping preserves angles but large area distortion
- Area distortion of this parameterization = density μ
- OT between μ and an uniform density

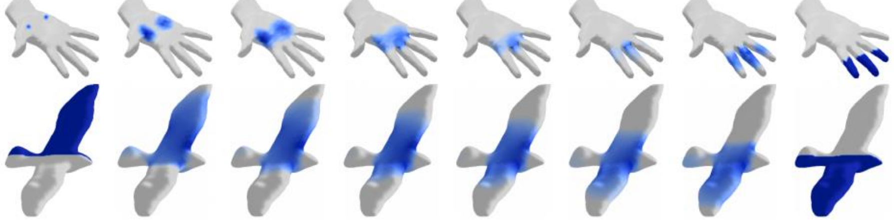
- Combination of conformal mapping + OT map = area preservation mapping
  - [Dominitz 09]: solved using a gradient flow
  - [Zhao 13]: solved using semi-discrete OT

[Zhao 13]

- Used for OT-based shape retrieval [Lipman 09, 11, Su 15]: W2 between mapped domain to compare surfaces

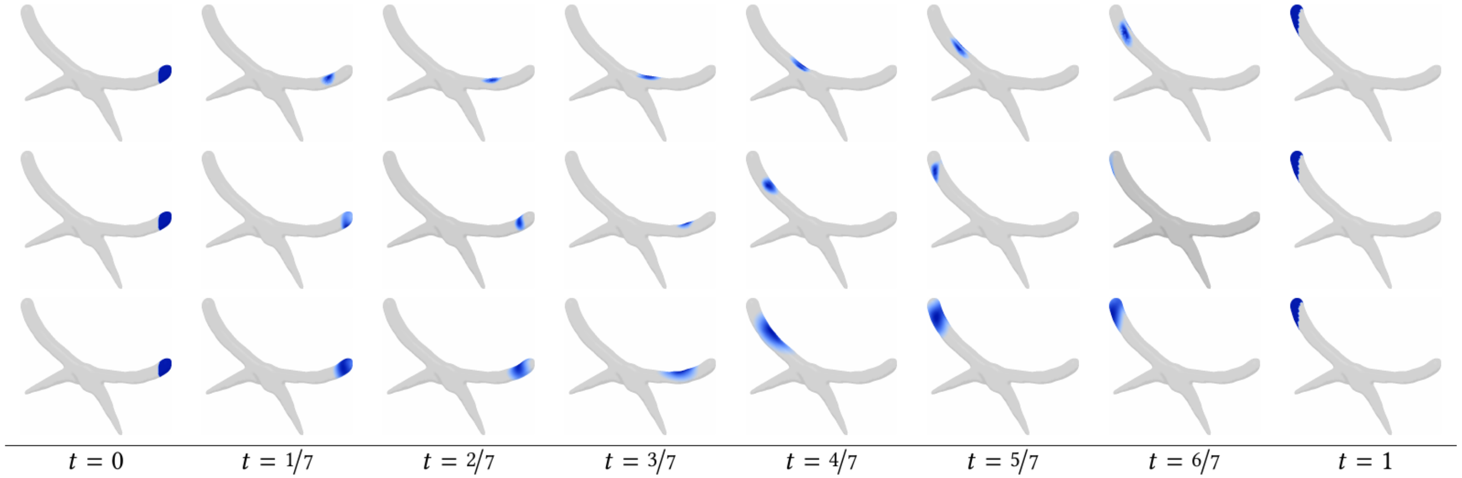
Applications to Geometry Processing

Transport on surfaces



[Lavenant 18]

- Transportation of densities defined on non-euclidean domain
- [Solomon 14] Variational formulation = look for a tangential vector field J
  - Flow lines of J are geodesics on the surface
  - Projection on a spectral basis -> family of geodesic distances
- Alternative: entropic regularization [Solomon 15]
- Better: Dynamic formulation for quadratic costs [Lavenant 18]



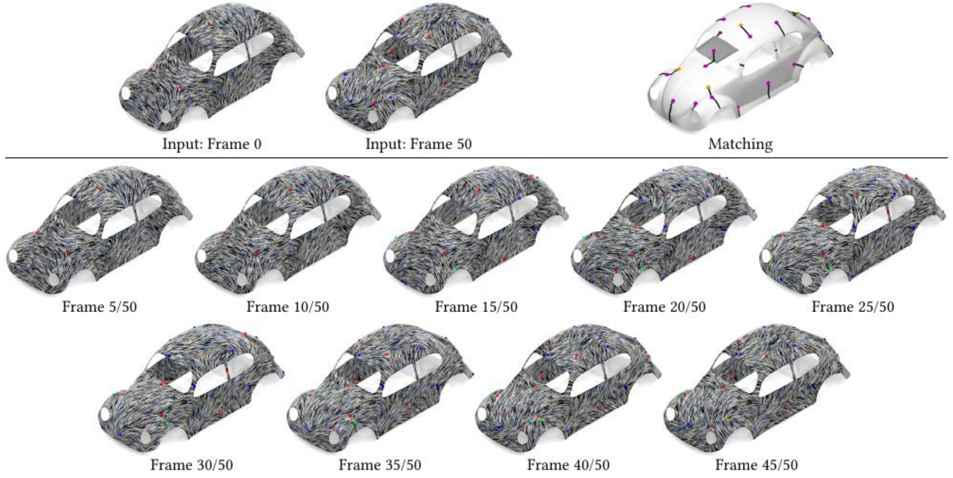
[Lavenant 18]

Two variants of [Solomon 15]

Applications to Geometry Processing

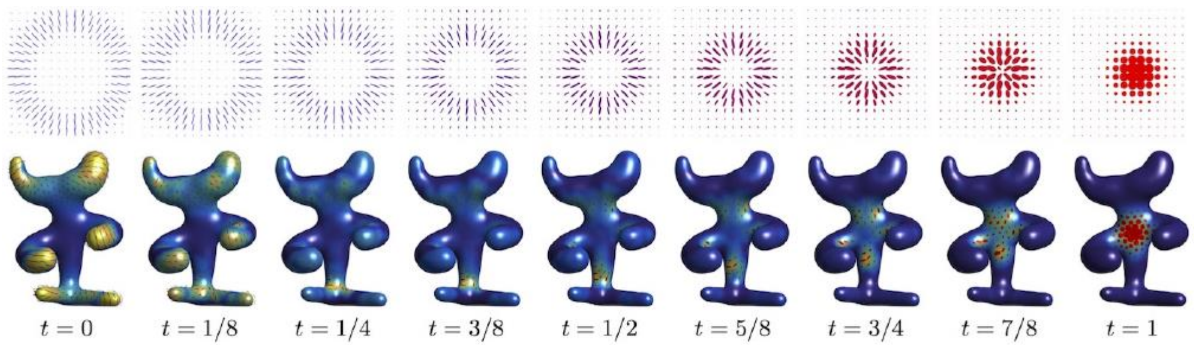
Transport on surfaces

- Interpolation of Directional field [Solomon 19]
  - OT matches the vector field singularities
  - Masses = classification of singularities
  - Allows for negative mass
  - LP solver



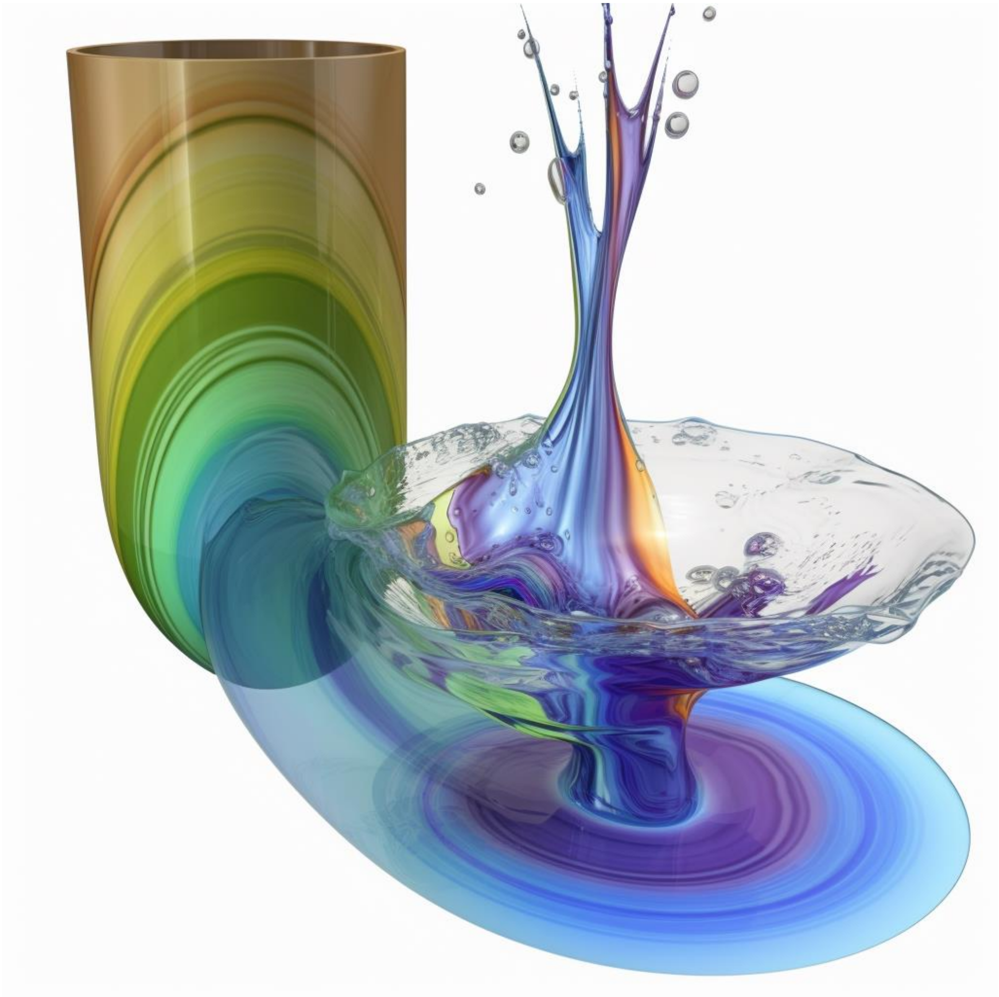
[Solomon 19]

- Tensor-valued distributions [Peyré 19]
  - Quantum regularized optimization
  - Modified Sinkhorn algorithm
  - Interpolation of orientation field



[Peyré 19]

Application to simulation and animation

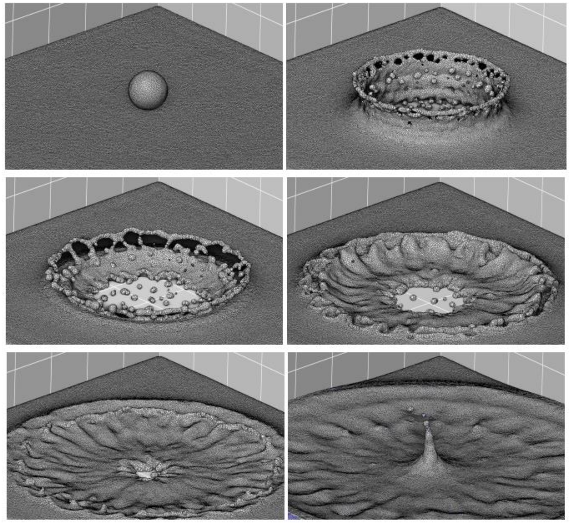


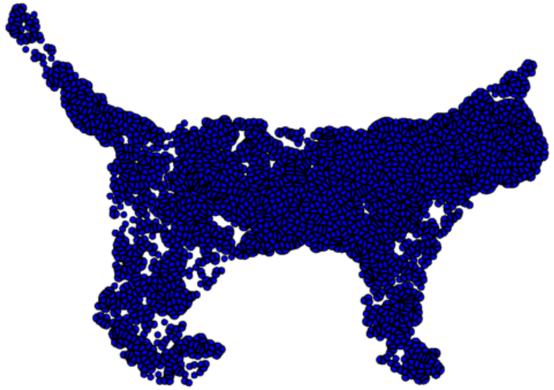
"a realistic illustration of the mathematical theory of Monge-Kantorovich's optimal mass transport applied to fluid simulation"

Applications to simulation and animation

Fluid simulation

- Incompressible Euler equations = flow under a volume preserving map
  - Fluid at any state = warping of a uniform density under area preserving map
  - Recover incompressibility using optimal transport
  - Particle-based approach: apply forces, advect, projection with OT
    - Projection with OT = Semi-discrete OT between particles and uniform density [Gallouët and Mérigot 2018]
    - Partial semi-discrete OT for free boundary fluids [Lévy 2022]
    - Semi-discrete entropy-regularized transport [Qu 2022]
    - Move particles towards the barycenter of their cell





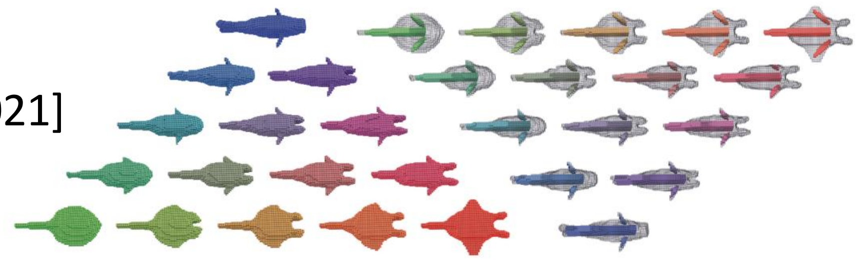
Applications to simulation and animation
Animation

- Wasserstein barycenters (with Sinkhorn divergence) to interpolate between animation keyframes [Zhang 2022]

- Linear program OT to interpolate between cloud keyframes [Webanck 2018]

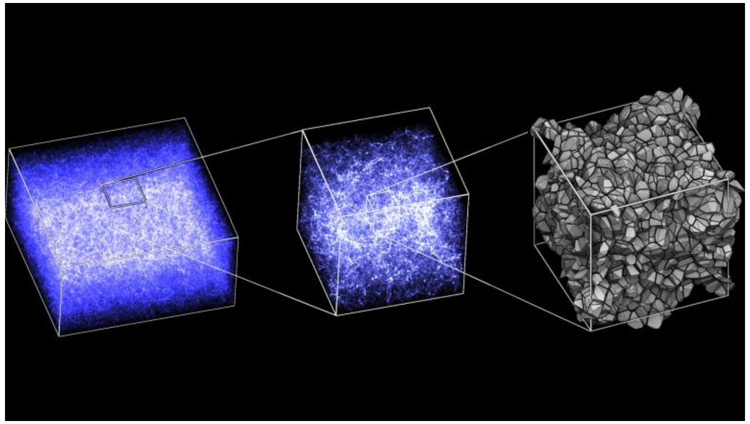
- Entropy-regularized barycenters to interpolate plant point clouds [Golla 2020]

- OT barycenters as a way to parameterized shapes for swimmers [Ma 2021]



Other popular applications

- Music interpretation
  - Recovering notes from a spectrum [Flamary 16]
  - Match pitches for smooth transition [Henderson 19]
- Cosmology
  - Accurate baryon variations through semi-discrete OT [von Hausegger 22]
- Text retrieval and analysis
  - Document distances (word distribution) using relaxed EMD [Kusner 15]
- Machine Learning
  - Semi-supervision [Solomon 14]/transfer learning [Courty 17]
- Genomics
  - Unbalanced OT between gene signatures [Schiebinger 14], Gene mover's distance [Bellazzi 21]



Discussion & Conclusion

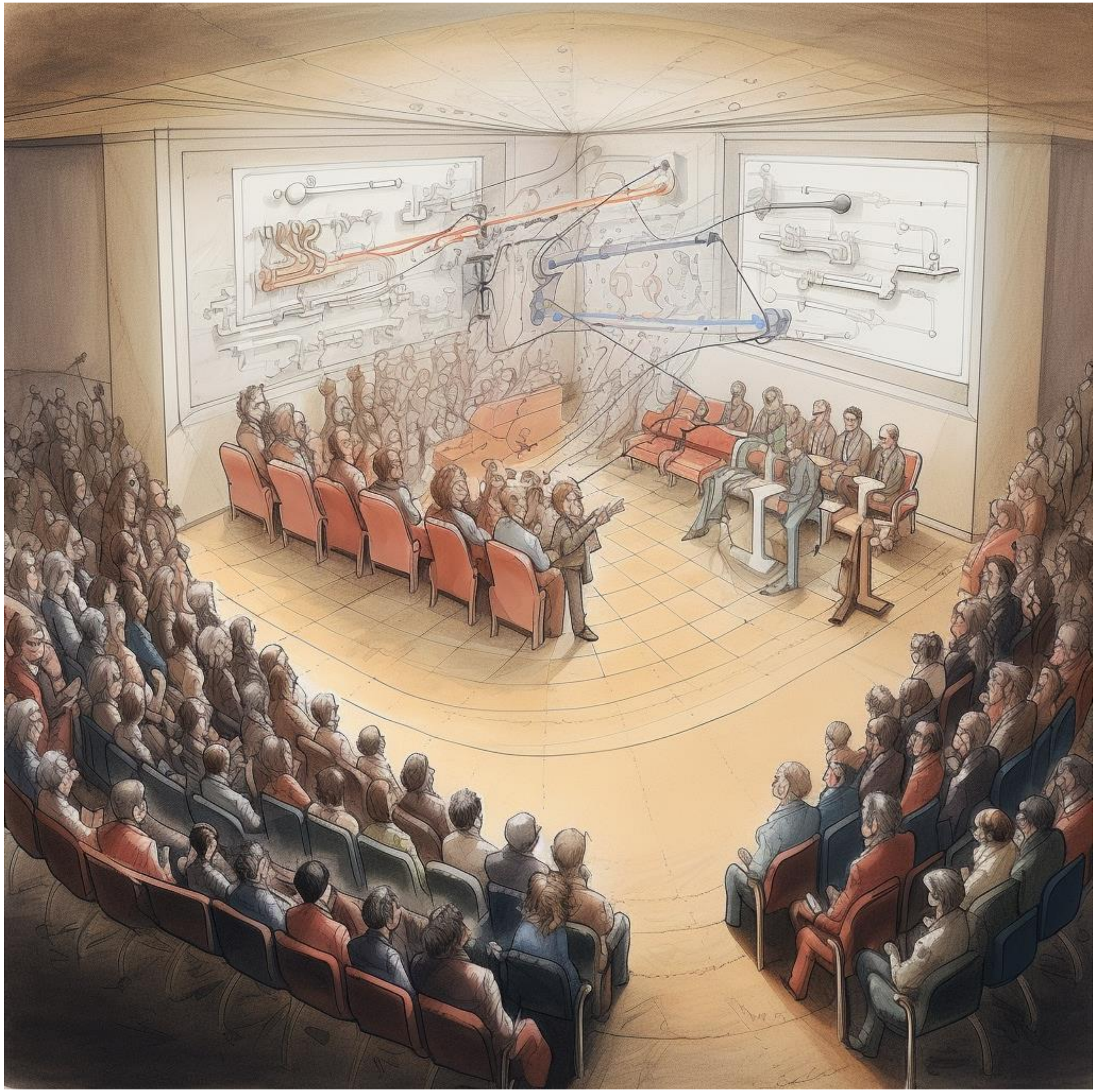
- Many applications of OT in particular for interpolation
    - May not preserve topology
    - Not always meaningful (e.g., for direct image warping)

- If input data ≠ probability distribution
    - How do we cast data as a probability distribution?

- Many results are superseded by Deep Learning
    - But the optimization tools are still useful even for … Deep Learning

- Remaining problem: scalability
    - Many fast algorithms but still slow for millions of variable

Questions?



a realistic illustration of the mathematical theory of Monge-Kantorovich's optimal mass transport applied to asking questions by an audience