Surface reconstruction from point clouds using optimal transportation

Guillaume Matheron

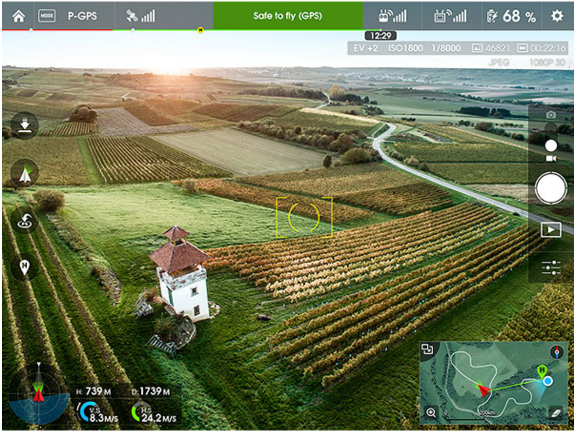
June-August 2015

Internship supervised by Pierre Alliez and David Cohen-Steiner

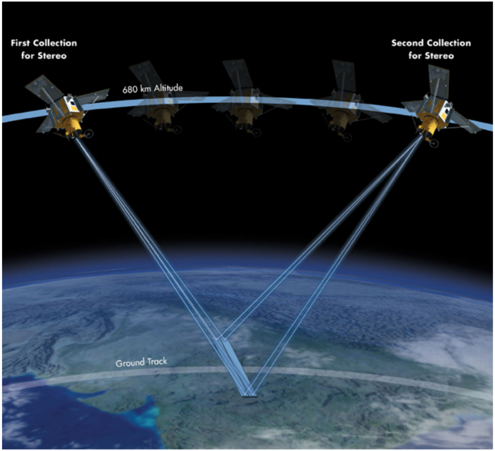
Input

Multiple and increasing variety of data sources

Drones


P-GPS
Safe to fly (GPS)
68 %
12:29
EV +2 ISO1800 1/8000 46821 00:22:16
JPEG 1080P 30
H: 739 M
D: 1739 M
V.S 8.3 M/S
H.S 24.2 M/S

Satellites


First Collection
for Stereo
Second Collection
for Stereo
680 km Altitude
Ground Track

Cars

Community data

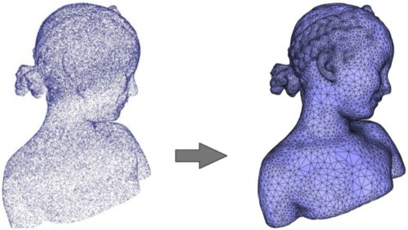
Snavely [2009]

Acquisition

- Stereography / Photogrametry
- Depth cameras

Challenge : Noise and outliers

Surface reconstruction



Set of points $\rightarrow$ surface representation
Suited for simulation, storage, visualization...

Problem statement

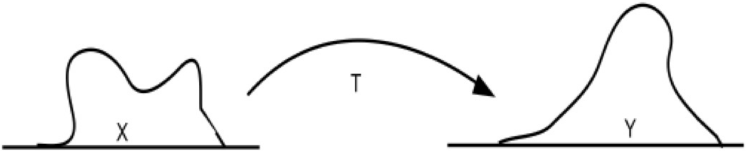
Discrete set of points $\rightarrow$ 2-manifold surface

Ill-stated problem

Idea : Interpret both as *mass distributions*.

Error metric for mass distributions : *Wasserstein distance*

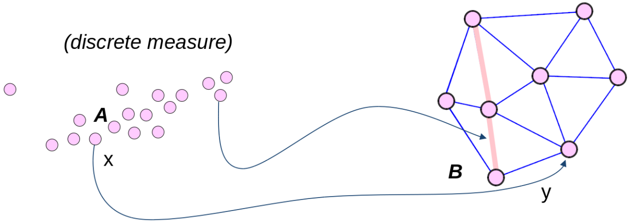
Optimal transportation



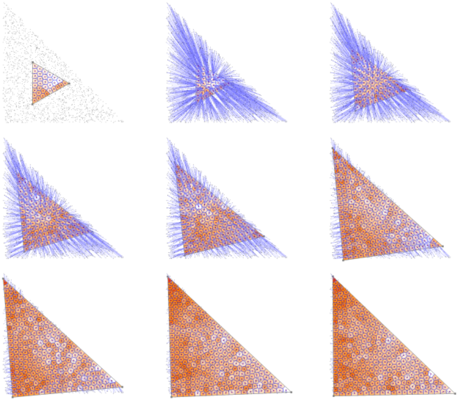
Wasserstein distance :

$$W_k(\lambda,\mu)^k = \inf \left\{ \int d(x-y)^k \mathrm{d}\pi(x,y) \mid \pi \in \Pi(\lambda,\mu) \right\}$$

Transportation as error metric



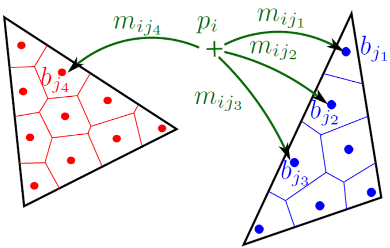
Digne et al. [2013]



Digne et al. [2013]

Finding the transport plan

- "Easy" for discrete distributions using Linear Programming (LP), but computationally intensive
- Continuous distributions (faces) split into "bins"



$\rightarrow$ (Digne et al. [2013])

Kantorovich-Rubinstein's dual formulation

$$W_1(\lambda, \mu) = \inf \left\{ \int d(x-y) \mathrm{d}\pi(x,y) \mid \pi \in \Pi(\lambda, \mu) \right\}$$

$$W_1(\lambda, \mu) = \sup \left\{ \int f(\lambda - \mu) \mid f \text{ continuous 1-Lipschitz} \right\}$$

$\rightarrow$ Quickly find an approximation of the best function f.

Previous work

We have :

- μ : discrete distribution (samples)
- λ : piecewise-constant distribution (faces)

We use an analog of the Wasserstein metric :

$$E = \frac{1}{N} \sum_{j=1}^{N} \left(\int f_j d\mu - \sum_{i=1}^{M} w_i \int f_j d\lambda_i \right)^2$$

Where :

- f_j is a set of chosen functions
- We solve for w_i

$$E = \frac{1}{N}\sum_{j=1}^{N}\left(\int f_j d\mu - \sum_{i=1}^{M} w_i \int f_j d\lambda_i\right)^2$$

rewritten as :

$$E = \sum_{j=1}^{N}\left(b_j - \sum_{i=1}^{M} w_i a_{j,i}\right)^2$$

Minimum reached when :

$$\forall k, 0 = \sum_{j=1}^{N} a_{j,k}\left(b_j - \sum_{i=1}^{M} w_i a_{j,i}\right)$$

$\rightarrow$ Solution of linear system

- Choice of f_j : bounded radial functions.
- $a_{j,i} = \int f_j d\lambda_i$ intensive to compute $\rightarrow$ Integrable in closed form on triangles (Hubert [2012])

Open questions :

- Placement of f_j : very heuristic
- Problem of validation

Proposal

$$W_1(\lambda, \mu) = \sup \left\{ \int f d(\lambda - \mu) \mid f \text{ continuous 1-Lipschitz} \right\}$$

LP program in 1D (discrete) :

$$\left\{ \begin{array}{ll} \text{minimize} & f(1)(\lambda(1) - \mu(1)) + \cdots + f(N)(\lambda(N) - \mu(N)) \\ \text{with respect to} & f(1), \cdots f(N) \\ \text{such as} & |f(2) - f(1)| \leq 1 \\ & \vdots \\ & |f(N) - f(N-1)| \leq 1 \end{array} \right. \tag{1}$$

Test case in 1D using Maple

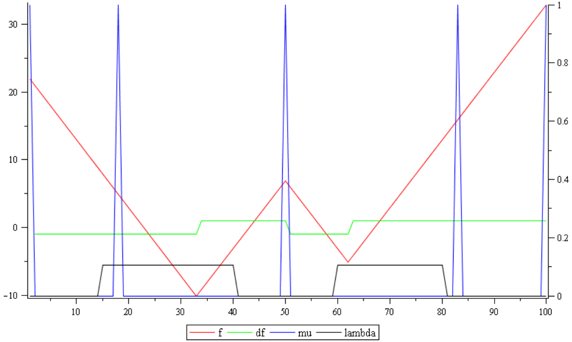
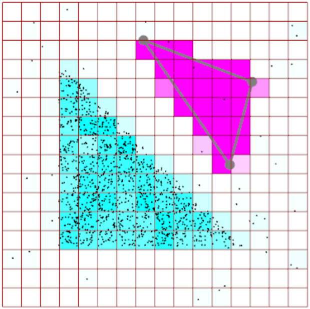


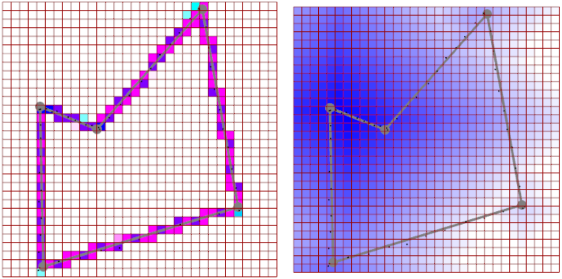
Figure: The f function always has a slope of 1 (Lipschitz condition attained)

Generalization to 2D

Project samples and triangles on an $N \times N$ lattice



LP with N^2 variables



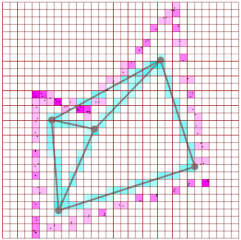
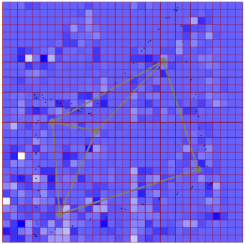
- Visualize f
- Accurate and fast
- Problem : Lipschitz constraints defined in only two directions
- Complexity lower than previous approach for similar resolutions

Wavelet Approach (Shirdhonkar and Jacobs [2008])

$$W_1 \approx \sum_i |T(\lambda)_i - T(\mu)_i| \, 2^{-2 \cdot j(i)}$$

$T(\lambda)_i$: i-th coefficient of the Discrete Wavelet Transform (DWT) of λ.

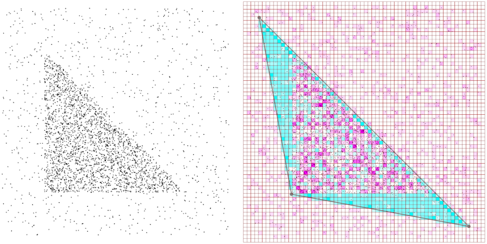
$\rightarrow$ Linear time in the number of cells ! $(O(N^2))$

 $\rightarrow$  $\rightarrow w = 0.35046$

$\lambda - \mu \rightarrow T(\lambda - \mu)$ (coefficients arranged in matrix) $\rightarrow$ Result

Experiments

Global minimum not where we want it to be



Problem : normalization (weight of triangle = weight of all samples $<$ weight of "good" samples)

Adjusting normalization

Problem : normalization (weight of triangle = weight of all samples < weight of "good" samples)

$$W_1 \approx \sum_i \left|\mathsf{k}T(\lambda)_i - T(\mu)_i\right| 2^{-2 \cdot j(i)}$$

Best k ?

Convex optimization

$$\Phi(k) = \sum_{i \in E} \left| kT(\lambda)_i - T(\mu)_i \right| 2^{-2j(i)}$$

Let $\Phi_i(k) = |kT(\lambda)_i - T(\mu)_i|$.

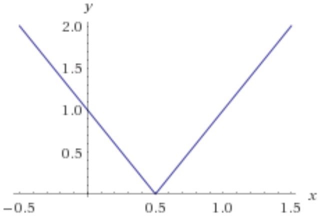


Figure: Φ_i

Convex optimization (cont.)

Φ is convex

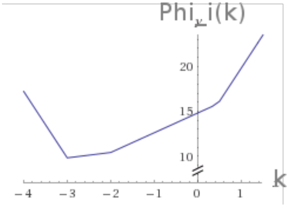


Figure: Φ

$O(\log_2(N^2))$ evaluations of Φ :
Complexity $O(N^2 \log_2 N)$

Explicit enumeration

$$\Phi(k) = \sum_{i \in E} |kT(\lambda)_i - T(\mu)_i| \, 2^{-2j(i)}$$

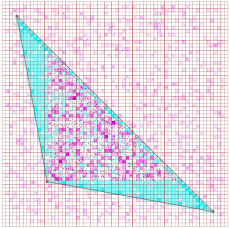
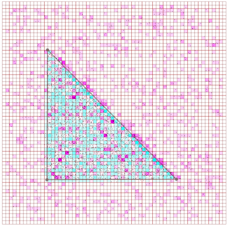
Let $k_l = \dfrac{T(\mu)_l}{T(\lambda)_l}$.

Set of singular points : $k \in \{k_l \mid l \in E\}$.

- Compute and sort all singular points (set of k_l)
- Find smallest $\Phi(k_l)$ via dichotomy (Φ is convex)

Complexity : $O(N^2 \log_2 N)$

Results

		
Fixed normalization	0.0146 (best)	0.0231
Adjusted normalization	0.0267	0.0067 (best)

Implementation

- Wavelet transform using GSL

```
gsl_wavelet2d_nstransform_matrix_forward(...);
```

- Multi-threading using OpenMP

```
#pragma omp parallel for
for (int i = 0; i < n; i++) {
    //(...)
    #pragma omp atomic
    totalSimplexWeight += w;
}
```

- LP solving using Coin|Or

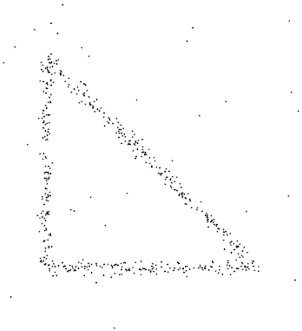
Reconstruction

Use metric to build surface from points

Methods :

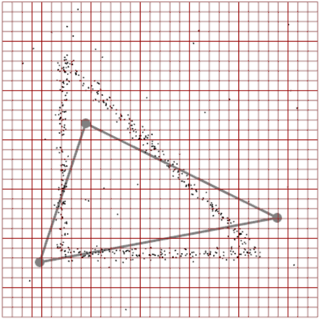
- Vertex relocation
- Coarse-to-fine (refinement, subdivision)
- Fine-to-coarse (edge collapse, decimation)

Vertex relocation



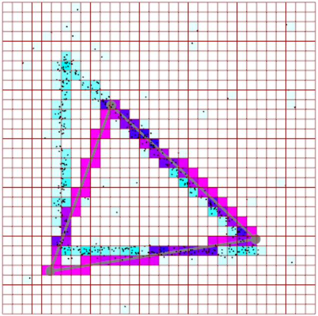
Input points

Vertex relocation



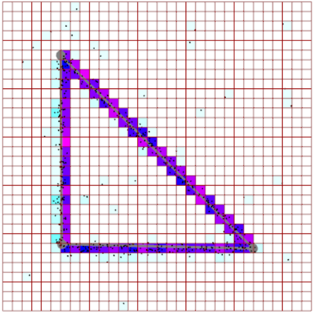
Initialization

Vertex relocation



Relocating vertices along gradient of W_1

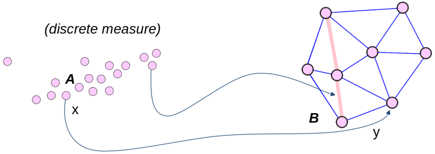
Vertex relocation

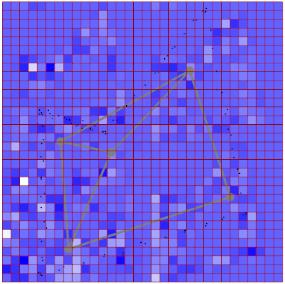


After relocation

Coarse-to-fine approach

Needs guided refinement operators.
Problem : we don't have the transport plan

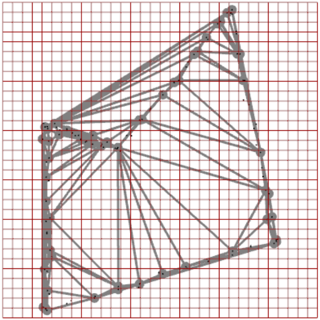




Fine-to-coarse approach

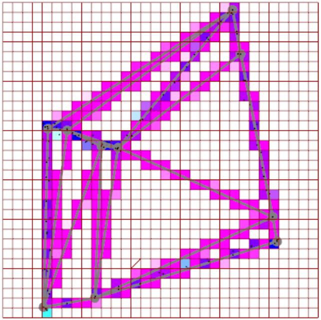
Input points

Fine-to-coarse approach



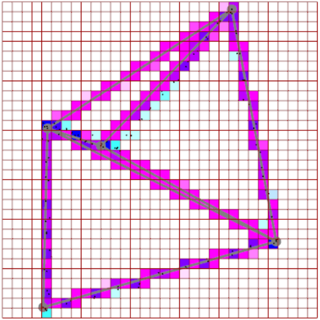
Delaunay triangulation

Fine-to-coarse approach



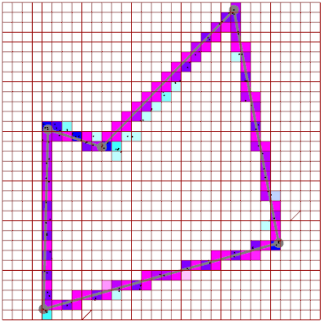
Removing vertices while minimizing ΔW_1

Fine-to-coarse approach



After decimation (5 vertices left)

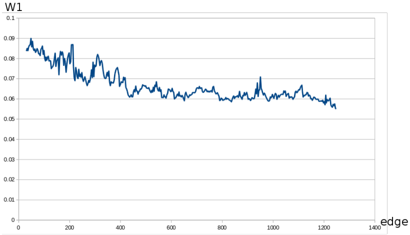
Fine-to-coarse approach



Remove three edges, minimizing ΔW_1

Limitations

- Remove edges
  $\rightarrow$ Greedy and operator-guided for now
  $\rightarrow$ LP at each computation of W_1 ?
- Stopping criteria ?



Conclusion

- Computational aspects of optimal transportation
- Wavelets are fast
  $\rightarrow$ scalable in 3D

Future work

- Local re-approximation of W_1
- Complexity-distortion tradeoff

Bibliography

J. Digne, D. Cohen-Steiner, P. Alliez, F. de Goes, and M. Desbrun. Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets. *Journal of Mathematical Imaging and Vision*, 2013.

E. Hubert. Convolution Surfaces based on Polygons for Infinite and Compact Support Kernels. *Graphical Models*, 2012.

S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. 2008.

K. N. Snavely. *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD thesis, 2009.