# Centroidal power diagrams with capacity constraints

Shi-Qing Xin, Bruno Lévy, Zhonggui Chen, Lei Chu, Yaohui Yu, Changhe Tu

# Centroidal Power Diagrams with Capacity Constraints: Computation, Applications, and Extension

Shi-Qing Xin
Ningbo University
The University of Hong Kong

Bruno Lévy
Inria-Nancy Grand-Est
Rue du Jardin Botanique

Zhonggui Chen*
Department of Computer Science
Xiamen University

Lei Chu
Department of Computer Science
The University of Hong Kong

Yaohui Yu
Department of Computer Science
The University of Hong Kong

Changhe Tu
School of Computer Science and Technology
Shandong University

Wenping Wang
Department of Computer Science
The University of Hong Kong

## Abstract

This article presents a new method to optimally partition a geometric domain with capacity constraints on the partitioned regions. It is an important problem in many fields, ranging from engineering to economics. It is known that a capacity-constrained partition can be obtained as a *power diagram* with the squared L2 metric. We present a method with super-linear convergence for computing optimal partition with capacity constraints that outperforms the state-of-the-art in an order of magnitude. We demonstrate the efficiency of our method in the context of three different applications in computer graphics and geometric processing: displacement interpolation of function distribution, blue-noise point sampling, and optimal convex decomposition of 2D domains. Furthermore, the proposed method is extended to capacity-constrained optimal partition with respect to general cost functions beyond the squared Euclidean distance.

**Keywords:** centroidal power diagram, displacement interpolation, convex decomposition, blue noise

**Concepts:** •**Computing methodologies** → **Computer graphics;**

## 1 Introduction

The central theme of the present paper is to apply and extend the notion of power diagram to computing optimal partition of a geometric domain with capacity constraints. A partition of a geometric domain is a collection of disjoint constituent parts whose union is equal to the whole domain. A partition is said to be optimal if all its partitioned regions, also called *cells*, have compact shapes with respect to some specified cost *kernel* that defines the cost function. Here "compactness" can be roughly measured by the isoperimetric quotient. Capacity constraints on a partition mean that each cell of the partition has a specified size, measured in volume (or area in 2D) or mass. Hence, an optimal partition with capacity constraints is a partition such that its cells have compact shapes as well as sizes equal to some specified quantities, unlike the optimal mass transport [Santambrogio 2015] that concentrates on capacity constraints regardless of shape compactness or the centroidal Voronoi tessellation [Du et al. 1999; Liu et al. 2009] that considers the opposite.

Finding an optimal partition by some criteria is a central issue in many applications. Optimal partitions, with or without capacity constraints, are used in computer graphics and geometric computing for point sampling [de Goes et al. 2012], information visualization [Balzer et al. 2005], shape segmentation [Lu et al. 2007], shape matching [Su et al. 2015], area-preservation mapping [Zhao et al. 2013] and mesh generation [Du et al. 1999; Liu et al. 2009]. The optimal partitioning problem with capacity constraints is also studied in engineering and economics for optimal resource location or provision of services. For example, it appears as the optimal coverage problem in robotics and sensor networks [Cortés 2010; Patel et al. 2014], the pattern formation problem in material science [Bourne et al. 2014], and the land consolidation problem in agriculture [Borgwardt et al. 2014].

The contribution of this paper is twofold.

1. An L-BFGS method with empirically superlinear convergence is developed for computing capacity-constrained centroidal power diagram, assuming the squared Euclidean distance as the cost kernel. We show its superiority to previous method and validate it in two applications: displacement interpolation of function distribution and blue-noise point sampling. Furthermore, the method is extended to computing optimal capacity-constrained partition with a general cost function;

2. Based on the centroidal power diagram, an effective method is developed for decomposing a 2D polygonal region into convex and compact cells with capacity constraints.

## 2 Related Work

**Voronoi Diagram & Power Diagram** The Voronoi diagram is a partition of a given domain $\Omega \subset E^d$ into regions based on the straight-line distance to a set of points $\{x_i \in \Omega\}_{i=1}^n$ (called *sites* or *generators*), where $x_i$ dominates the subregion

$$\Omega_i : \{x \in \Omega \mid \|x - x_i\| \leq \|x - x_j\|, j \neq i\},$$

which is called the *Voronoi cell* of the site $x_i$. A power diagram is defined as an extension of Voronoi diagrams, where each site $x_i$ has an associated coefficient (called a "weight") and dominates the subregion

$$\Omega_i : \{x \in \Omega \mid \|x - x_i\|^2 - w_i \leq \|x - x_j\|^2 - w_j, j \neq i\}.$$

*Corresponding author (chenzhonggui@xmu.edu.cn)



SA '16 Technical Papers, December 05-08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2982428

$\Omega_i$ is called the *power cell* of the site $x_i$. The power diagram reduces to the Voronoi diagram when all the weights are equal [Aurenhammer 1987; Okabe et al. 1992].

**Centroidal Voronoi Tessellation (CVT)** A centroidal Voronoi tessellation (CVT) of a set of sites $X = \{x_i\}_{i=1}^n$ is a special Voronoi diagram for which each site $x_i$ coincides with the centroid (i.e. the center of mass) of its Voronoi cell $\Omega_i$ [Du et al. 1999]. Alternatively, a CVT can be also defined as a stationary point of the following objective function:

$$Q(X) = \sum_{i=1}^{n} \int_{\Omega_i} ||x - x_i||^2 \rho(x) \mathrm{d}\sigma, \tag{1}$$

where $\rho(x)$ is a $C^1$-smooth density function on the whole domain $\Omega$. Domain partition using CVT yields convex polyhedral cells (barring those abutting boundaries) that have compact shapes and similar sizes, assuming that $\rho(\cdot)$ is uniform. Due to these properties, CVT is widely used for computing optimal partitions in various applications [Lu et al. 2014; Zhou et al. 2015], and in some cases even for optimal partition with capacity constraints [Balzer and Heck 2008; Chen et al. 2012; Li et al. 2010]. However, it is known that the class of partitions given by the Voronoi diagram is too restrictive to properly accommodate capacity constraints [Zhang et al. 2016]. A prevailing method for computing CVT is Lloyd's method, which behaves as a gradient-descent method and thus converges linearly. A quasi-Newton method for computing CVT is proposed in [Liu et al. 2009] that is based on the limited-memory BFGS technique and is shown to demonstrate super-linear convergence in practice.

**Optimal Mass Transport (OMT)** The optimal mass transport (OMT) was formalized by the French mathematician Gaspard Monge [1781] who studied the most economical way of moving soil from one area to the other and introduced the theory in terms of minimizing the $L_1$ norm of the distance transported. The problem of minimizing $L_2$ cost whose kernel is the squared distance (note that the square root of the minimum $L_2$ cost is called the 2-Wasserstein metric), however, takes on additional structure and can be expressed as the fully nonlinear degenerate elliptic Monge-Ampère equation. In 1949, Kantorovich proved the existence and uniqueness of the optimal transport plan and showed how Monge's problem is connected to linear programming [Kantorovich and Gavurin 1949]. The Monge-Kantorovich optimization technique has been studied in a wide range of research fields in the past two decades, including imaging, adaptive meshing, geophysical fluid dynamics, and cosmology.

It is shown in [Pogorelov 1994], [Gangbo and Cann 1996] (example 1.6) and [Aurenhammer et al. 1998] (or as a direct consequence of [Brenier 1991]) that for the semi-discrete OMT (from a discrete probability distribution function to a continuous one), the class of partitions parameterized by the power diagram provides the proper representation to accommodate capacity constraints in the following sense. Given a set of fixed sites in a domain, there exists a unique set of weights (up to an additive constant) such that the power cells, which are determined by the given sites and the associated weights, satisfy the specified capacity constraints and minimize the sum of least squares costs of all the cells. Here, the least squares cost of the cell $\Omega_i$ associated with a site $x_i$ is defined to be $\int_{\Omega_i} \|x - x_i\|^2 \rho(x) \mathrm{d}\sigma$. This means that we just need to look for the capacity-constrained partition within the set of power diagrams with different weighting schemes, rather than having to search in the set of arbitrary partitions of the domain, which obviously cannot be parameterized by a finite number of parameters.

**Centroidal Power Diagram (CPD)** In this paper, we are interested in computing the optimal partition of a geometric domain with specified capacity constraints. Similar to the conventional OMT problem, the solution to such a capacity-constrained optimal partition problem with the least squares cost function can be represented by a power diagram [Balzer et al. 2009]. More specially, the solution is given by a centroidal power diagram (CPD) which is a special power diagram in which each site coincides with the centroid of its power cell. Interestingly, the weights of the power diagram exactly correspond to the degrees of freedom required to account for capacity constraints. However, one should notice that the relation between the weights and the capacities of the cells is non-trivial (in this context, the term "weight" is misleading, they are completely different from the masses of the cells).

CPD was used for blue-noise sampling in computer graphics. In numerical implementation, Balzer et al. [2009] proposed several iterative schemes to compute capacity-constrained optimal partitions, including the method based on pixel-based clustering and the methods that enforce capacity constraints on the Voronoi diagram as well as the power diagram. These methods are easy to implement. But because they update only one single pixel or site in each iteration, the methods have very slow convergence and low efficiency. Later, a Lloyd-like method that repeatedly moves each site to the mass center of its cell is proposed to compute CPD for blue-noise sampling [de Goes et al. 2012], which has only linear convergence.

The above discussions naturally lead to two questions. First, is it possible to develop an efficient method that outperforms these existing methods? Second, remarking that the least squares cost function is in fact the squared $L_2$ metric, we want to know whether the new numerical method is able to deal with the CPD problem with a general cost. We will give positive answers to both questions in this paper.

## 3 Capacity-Constrained Optimal Partition

### 3.1 Problem Formulation

We first recall the notions used for a power diagram in $E^d$, $d$-dimensional Euclidean space. Let $\Omega \subset E^d$ be a convex, closed, bounded and connected domain. Let $X = \{x_i\}_{i=1}^n \subset \Omega$ be a set of points, called *sites*, with its associated *weights* $\{w_i\}_{i=1}^n \subset R$. Then the power diagram induced by the weighted points $(x_i, w_i)$ is the collection of convex polytopes, or *cells*, $PD = \{PD_i\}_{i=1}^n$, where $PD_i = \{x | \|x - x_i\|^2 - w_i \leq \|x - x_j\|^2 - w_j, \forall j, j \neq i\}$. The restriction of the power diagram $PD$ to the domain $\Omega$ induces a partition of $\Omega$, denoted by $PD_\Omega = \{PD_i \cap \Omega\}_{i=1}^n$.

Now consider partitioning a domain $\Omega$ in Euclidean space $E^d$ into a set of $n$ regions $R = \{R_i\}_{i=1}^n$ in a capacity-constrained manner. Let $X = \{x_i\}_{i=1}^n \subset \Omega$ denote $n$ given points, also called *sites*, with associated capacity constraints $c_i > 0$, $i = 1, 2, \ldots, n$. We define the least squares cost of each region $R_i$, given by $x_i$, to be $\int_{R_i} \|x - x_i\|^2 \rho(x) \mathrm{d}\sigma$, where $\rho(x)$ is a $C^1$-smooth density function on $\Omega$. The capacity constraint here requires that $|R_i| \equiv \int_{R_i} \rho(x) \mathrm{d}\sigma = c_i$, $i = 1, 2, ..., n$, that is, the capacity (i.e. area or volume) of each region $R_i$ matches the constraint $c_i$. (Of course, $\sum_i c_i$ is assumed to be equal to $|\Omega| \equiv \int_\Omega \rho(x) \mathrm{d}\sigma$.) Then the problem of *capacity-constrained optimal partition* is to find the partitions $R_i$ *and* the sites $x_i$ such that the total cost

$$Q(X, R) = \sum_{i=1}^{n} \int_{R_i} ||x - x_i||^2 \rho(x) \mathrm{d}\sigma \tag{2}$$

is minimized, subject to the constraints $|R_i| = c_i$, $i = 1, 2, ..., n$.

In the case of the L2 cost, it follows from [Aurenhammer et al. 1998] that the solution $(X^*, R^*) = \arg\min Q(X, R)$ exists and is given by a power diagram; See [Pogorelov 1994] and [Gangbo and Cann 1996] for a proof, as well as the derivations in [Gu et al. 2013]. Aurenhammer [1998] further showed that the optimal power diagram can be found by extremizing

$$F(X, W) = \sum_{i=1}^{n} \int_{PD(W)_i} ||x - x_i||^2 \rho(x) \mathrm{d}\sigma - \sum_{i=1}^{n} w_i(|PD(W)_i| - c_i), \tag{3}$$

where $PD(W)_i$ (or $PD_i$) is a power cell dominated by $x_i$. A key observation made in [Aurenhammer et al. 1998] is that, when the sites $X$ in $F(X, W)$ are fixed, their weights can be computed as a maximizer of a concave function to make the cells of the resulting power diagram meet the capacity constraints $|PD_i| = c_i$, $i = 1, 2, ..., n$. This observation leads to a Lloyd-like method [Cortés 2010; de Goes et al. 2012] for minimizing the objective function $F(X, W)$ in Eqn. (3), i.e., alternatively optimizing the weights to meet the given capacity constraints and moving the sites to their respective mass centers. We will discuss these numerical techniques in detail in the next section, and introduce a new method that better exploits the derivatives of the objective function, resulting in a faster convergence.

### 3.2 Partition with General Cost Functions

Given a general distance cost kernel $d(x, y)$, where $x, y \in \Omega \subset E^d$, a *generalized power diagram* (also called a Laguerre diagram) of $\Omega$ with the weighted sites $\{(x_i, w_i)\}_{i=1}^n$ is defined to be the collection of cells $GD = \{GD_i\}_{i=1}^n$, where $GD_i = \{x | d(x, x_i) - w_i \leq d(x, x_j) - w_j, \forall j, j \neq i\}$. When $d(x, y)$ is the squared distance, the generalized power diagram reduces to the traditional power diagram whose cells are convex polytopes. However, with a general cost, the cells $GD_i$ are not necessarily convex and their boundaries are curves or curved surfaces rather than straight lines or planes.

Let $\mu$ denote a measure with non-vanishing density $\rho(.)$, supported by the domain $\Omega$. Suppose that each site $x_i$ has capacity constraint $c_i > 0$, such that $\sum_i^n c_i = |\Omega| \equiv \int_\Omega \mathrm{d}\mu$, which is the total mass of the measure $\mu$. Again, let $d(x, y)$ denote a distance cost between any two points $x$ and $y$ in $\Omega$. Let $R = \{R_i\}_{i=1}^n$ denote a partition of $\Omega$, that is, $\cup R_i = \Omega$ and $R_i \cap R_j = \emptyset, \forall i \neq j$. Define the cost of the partition $R$ with respect to the sites $X = \{x_i\}_{i=1}^n$ as

$$Q(X, R) = \sum_{i=1}^{n} \int_{R_i} d(x, x_i) \mathrm{d}\mu. \tag{4}$$

In what follows, we support that the distance $d(x, y)$ satisfies the twist condition, i.e. $\nabla_x d(x_0, y)$ is one-to-one $\forall x_0$, which ensures the existence and uniqueness of the optimal transport plan (see e.g. [McCann and Guillen 2010] p. 10 and [Santambrogio 2015]). Then the capacity-constrained optimal partition problem is to find the sites $X^*$ and partition $R^*$ that minimize the function $Q(X, R)$ subject to the *capacity constraints* $c_i = |R_i| \equiv \int_{R_i} \mathrm{d}\mu$, $i = 1, 2, \ldots, n$. The difference between this definition of the general setting and the previous simple setting of the squared $L_2$ cost is that here $\Omega$ may be a curved surface and that $d(x, y)$ is a general cost kernel.

Next we shall show that the optimal partition $R^*$ with capacity constraints assuming a general cost function $Q(X, R))$ can be found by only considering the candidates of extended power diagrams where each region (possibly with curved boundaries) is associated with a weight. This is a natural extension from [Aurenhammer et al. 1998; Santambrogio 2015]. Its proof is given in the appendix.

**Theorem 1**: *For any fixed sites $X = \{x_i\}_{i=1}^n$, the capacity-constrained partition $R = \{R_i\}_{i=1}^n$ of the domain $\Omega$ that minimizes the cost function in Eqn. (4) is the one such that the boundary of any two adjacent regions $R_a$ and $R_b$, which are associated with the sites $a$ and $b$ in $X$ respectively, is defined by the equation $d(x, a) - w_a = d(x, b) - w_a$ for some constants $w_a$ and $w_b$.*

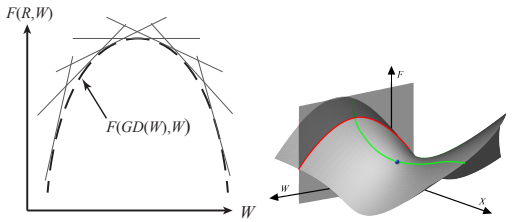
Similar to the treatment in [Aurenhammer et al. 1998] for the case of the least squares cost function, we build a new objective function to accommodate the optimal partition according to the Lagrangian Multiplier Theorem:

$$F(X, W) = \sum_{i=1}^{n} \int_{GD(W)_i} d(x, x_i) \mathrm{d}\mu - \sum_{i=1}^{n} w_i(|GD(W)_i| - c_i). \tag{5}$$

It can be shown that $F(X, W)$ is concave w.r.t. $W$ and therefore has a unique maximizer (up to an additional constant), denoted $W^* = \{w_i^*\}_{i=1}^n$. When the sites $X = \{x_i\}_{i=1}^n$ are fixed, the optimal partition exists uniquely and is given by the generalized Voronoi diagram $GD(W^*)$. When the capacity constraints are met, $F(X, W)$ is exactly equal to the cost function in Eqn. (4). Furthermore, the gradient of $F(X, W)$ w.r.t. $W$ can be shown to be [Aurenhammer et al. 1998; Santambrogio 2015]

$$\nabla_W F(X, W) = -(|GD(W)_1| - c_1, \ldots, |GD(W)_n| - c_n), \tag{6}$$

which is a normal vector to the tangent plane of $F(X, W)$. Hence, to seek the maximizer $W^*$ of $F(X, W)$, we have $\nabla_W F(X, W^*) = 0$, which means that all the capacity constraints are satisfied.




(a) (b)

**Figure 1:** *An optimal partition $(X^*, W^*(X^*))$ is a saddle point of the cost function $F(X, W)$ in Eqn. (5).*

To summarize, we optimize over all the sites $X$ and weights $W$ to find the generalized weight Voronoi diagram that gives an optimal partition with capacity constraints. Because there is a unique set of weights $W$ (up to an additional constant) for every set of sites $X$ to make the resulting weighted Voronoi diagram satisfy the capacity constraints, all the capacity-constrained partitions can be represented by a surface $W = W^*(X)$, which will be called the *constraint surface* (see the green curve in Fig. 1(b)). For each point $(X_a, W^*(X_a))$ on the constraint surface, the restriction $F(X_a, W)$ of $F(X, W)$ (Eqn. (5)) to the subspace $X = X_a$ attains its unique maximum at the point $(X_a, W^*(X_a))$ among all the weighting schemes $W$ (see the red curve in Fig. 1(*b*)). Meanwhile, the restriction $F(X, W^*(X))$ of $F(X, W)$ to the constraint surface attains its local minimum at a point $(X^*, W^*(X^*))$ that gives an optimal capacity-constrained partition, which is marked with the blue ball on the green curve in Fig. 1(*b*). Hence, an optimal partition $(X^*, W^*(X^*))$ is a saddle point of the cost function $F(X, W)$ in Eqn. (5).

# 4 Algorithm for Computing Centroidal Power Diagrams

In this section we propose an efficient method for computing CPDs, as well as the implementation details.

## 4.1 Overall Algorithm

As discussed above, if $(X^*, W^*)$ defines the optimal partition, the cost function $F(X, W)$ (see Eqn. (5)) is minimized by $X^*$ while maximized by $W^*$. In other words, finding $(X^*, W^*)$ is neither a pure maximization problem nor a minimization problem. de Goes et al. [2012] suggested updating $X$ and $W$ in an alternative style, where the two key operations are: (1) finding the optimal weighting scheme $W$ to meet the given capacity constraints using Newton's method while fixing $X$ [Benamou et al. 2014]; and (2) moving each site $x_i$ to the mass center of its corresponding power cell. However, this iterative scheme has only linear convergence due to the ineffectiveness of the latter operation. As de Goes et al. [2012] observed, developing an efficient CPD algorithm with super-linear convergence is non-trivial.

Based on the discussion in Section 3, the optimal weights $W^*$ maximizing the cost function in Eqn. (5) is uniquely determined (up to an additional constant) for a given site collection $X$. Therefore the cost function can be reconsidered as a function of $X$ :

$$\begin{aligned} \text{Minimize} \quad F(X, W^*(X)) = &\sum_{i=1}^{n} \int_{GD(W^*)_i} d(x, x_i) \mathrm{d}\mu \\ &- \sum_{i=1}^{n} w_i^* (|GD(W^*)_i| - c_i), \end{aligned} \tag{7}$$

where $W^*$ depends on $X$ so that the specified capacity constraints are met. Based on this formulation, we will use the combination of the reduced-gradient technique and the L-BFGS solver [Lasdon et al. 1974; Liu and Nocedal 1989] to move $(X, W)$ on the constraint surface $W = W^*(X)$ towards the minimum of $F(X, W^*(X))$.

Our method works as follows. We start from a feasible point $(X_0, W_0)$, i.e. a point on the constraint surface $W_0 = W^*(X_0)$. Without loss of generality, we assume that $(X_k, W_k)$, the position in the $k$-th iteration, is a feasible point as well. Next, we compute the reduced-gradient vector $\nabla_{x_i} F(X, W^*(X))$ and feed it to the LBFGS solver to update the sites $X_k$ to $X_{k+1}$. Note that the weights are computed according to the up-to-date sites before we evaluate the objective function $F$. Thus the $(k+1)$-th iteration yields the next position $(X_{k+1}, W_{k+1})$. Repeat the process until some stopping criteria is met.

We now consider the derivation of $\nabla_{x_i} F(X, W^*(X))$. Based on the envelope theorem [Milgrom and Segal 2002], we know

$$\nabla_{x_i} F(X, W^*(X)) = \nabla_{x_i} F(X, W). \tag{8}$$

According to Reynolds' transport theorem [Leal 2007] that recasts derivatives of integrated quantities, we can further show that when the cost kernel $d$ is differentiable everywhere in the domain, the gradients of $F(X, W^*(X))$ also have a simple form:

$$\nabla_{x_i} F(X, W^*(X)) = \int_{GD(W^*)_i} \nabla_{x_i} d(x, x_i) \mathrm{d}\mu. \tag{9}$$

The pseudocode of our method is given in Algorithm 1. It is worth noting that for general cost kernel that is different from the 2-Wasserstein metric whose kernel is the squared distance, instead of using Newton's method, we use the L-BFGS method to update $W$, because the Hessian matrix is much more complicated in this case than that in the case.

---

**Input:** domain $\mathcal{D}$, density $\rho$, cost kernel $d$, number of points $n$, capacity constraints $\{c_i\}$ and a threshold $\epsilon$ as the termination condition.
**Initialization:** set $k = 0$ and $X_0$ to be $n$ randomly generated sites.
**repeat** *//L-BFGS*
  **repeat** *//To meet the capacity constraints*
    Update $W_k$ by Newton's method or L-BFGS method
  **until** $||\nabla_W F(X, W)|| < 10^{-12}$
  Compute the gradients $\nabla_X F(X, W^*(X))$ in Eqn. (9)
  Estimate the step-size for updating $X$ by a line-search
  $X_{k+1} \leftarrow X_k$
  $k \leftarrow k + 1$
**until** $||\nabla_X F(X, W^*(X))|| < \epsilon$
**Output:** $(X_k, W_k)$

---

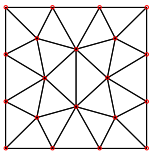
**Algorithm 1:** Computing CPDs using L-BFGS.

## 4.2 Optimization Options

The computation of $W^*$ is crucial to accurate computation of $\nabla_{x_i} F(X, W^*(X))$. As de Goes et al. [2012] observed, it usually takes 3 to 5 iterations to bring the residual of capacity constraints to be within an accuracy of $10^{-12}$. Therefore we require $||\nabla_W F(X, W)|| < 10^{-12}$ during the step of updating $W$. In implementation, rather than always resetting $W$ to 0 at the very beginning of Newton's method, it is better to use the previous values of $W$ to initialize Newton's method. This is helpful for reducing the computation cost especially when $||\nabla_W F(X, W)||$ becomes very small. We observed an improvement of about 10% due to this "trick" in our experiments. When empty cells occur, we reduce the step size by half repeatedly until each site dominates a non-empty cell, as in [de Goes et al. 2012; Zhao et al. 2013]. Note that the convergence of this Newton step-control scheme was recently proved [Kitagawa et al. 2016].

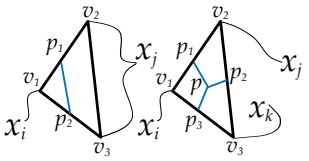
For the 2-Wasserstein metric, the optimal partition is given by power diagrams. In this case, the Hessian matrix of $F(X, W)$ w.r.t. $W$ has an elegant form [de Goes et al. 2012]. But for a general cost kernel different from the squared Euclidean distance, the coefficients of the Hessian matrix (gradient of the distance function integrated over the facets of the power diagram) are much more complicated, and in most cases not known in closed form. For this reason we use the L-BFGS method that only needs the first-order derivatives (that always correspond to the difference between the capacity constraints and the measures of the Laguerre cells).

## 4.3 Numerical Integration and Boundary Treatment



When the density function $\rho$ is constant and the cost kernel is the 2-Wasserstein metric, the integrand in Eqn. (7) has an explicit representation by using the Green formula [Dobrovolsky 1972] to transform the computation of the cost function, as well as its gradients, into contour integration. But in general cases, the cost function cannot be reduced to an explicit representation and thus discretization of the domain, typically a sufficiently refined triangulation, is required. Therefore, at the very beginning of the algorithm, we randomly distribute a set of points inside the domain and optimize their positions [Liu et al. 2009]. After that, we project the outer-ring points (whose Voronoi cells intersect the boundary) onto the boundary and re-optimize the other

points. Finally we compute the constrained Delaunay triangulation w.r.t. these points as discretization of the domain; See the inset figure.




Besides, the optimal partition of the domain for a general cost kernel $d(\cdot,\cdot)$ may not be the power diagram. We have to develop an effective technique to build the partition with the help of the base triangulation. First of all, we take the sites in $X$ as sources and compute the weighted discrete distance field using the fast sweeping technique so that each vertex $v$ of the base triangulation finds its nearest site $x_i$ in $X$, i.e. $d(v,x_i)-w_i \le d(v,x_j)-w_j, \forall j \neq i$. After that, we compute the decomposition of each triangle depending on the following cases:

- ◇ If the site $x_i$ gives the minimum weighted distances to $v_1, v_2, v_3$ at the same time, we simply assign the triangle $f$ to $x_i$.
- ◇ If the site $x_i$ gives the minimum weighted distance to $v_1$ and the site $x_j$ gives the minimum weighted distances to $v_2, v_3$, we come to find two dividing points $p_1 \in \overline{v_1v_2}$ and $p_2 \in \overline{v_1v_3}$ such that $\overline{p_1p_2}$ can be taken as a segment delimiting $x_i$'s cell and $x_j$'s cell. The dividing point $p_1$ can be found from the equation $d(x_i,v_1) - w_i + (x - v_1)\cdot\nabla_x d(x,x_i)\ |_{v_1} = d(x_j,v_2) - w_j + (x-v_2)\cdot\nabla_x d(x,x_j)\ |_{v_2}$ . The other dividing point $p_2$ can be computed likewise. See the inset figure.
- ◇ If the minimum weighted distances to $v_1, v_2, v_3$ are respectively given by $x_i, x_j, x_k$, then we find three dividing points $p_1 \in \overline{v_1v_2}, p_2 \in \overline{v_2v_3}, p_3 \in \overline{v_3v_1}$ as mentioned above. We denote the center point by $p = (p_1 + p_2 + p_3)/3$. Finally, we assign the subcell $\widehat{v_1p_1pp_3}$ to $x_i$, $\widehat{v_2p_2pp_1}$ to $x_j$, and $\widehat{v_3p_3pp_2}$ to $x_k$, as the inset figure shows.

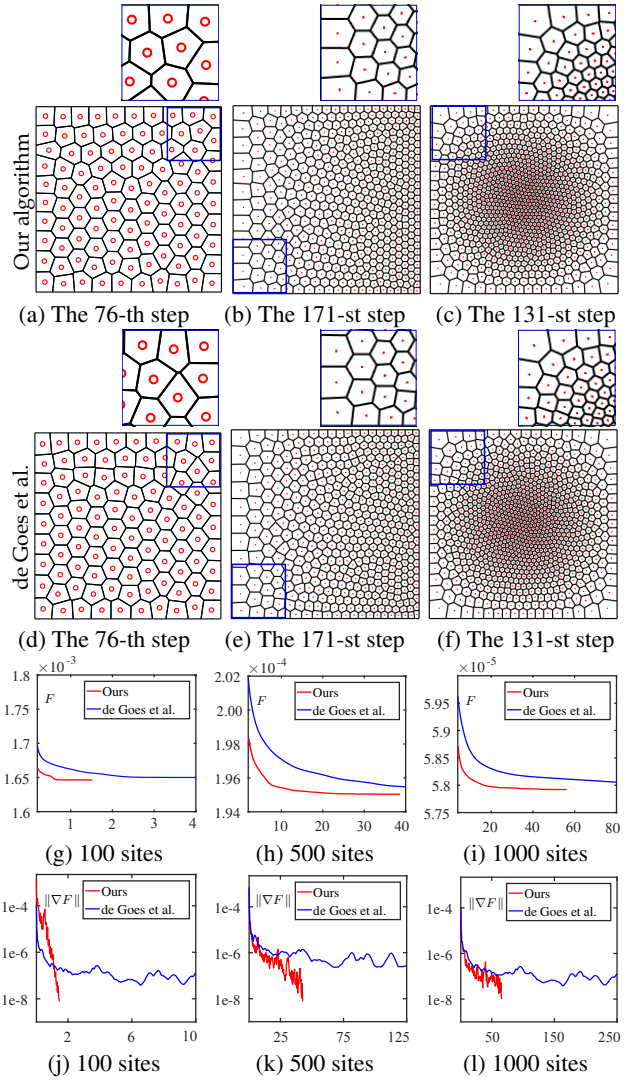
Another implementation detail we need to mention is boundary treatment. During the optimization process, some sites may jump out of the domain $\Omega$ and we need to pull them back to $\partial\Omega$ or the interior side of $\Omega$. Suppose that in the $j$-th iteration, the $i$-th site $x_i^{(j)}$ is outside $\Omega$ and we wish to pull $x_i^{(j)}$ back. We find the intersection point $p$ between $\partial\Omega$ and the connecting line segment of $x_i^{(j)}$ and its previous position $x_i^{(j-1)}$, and take $p$ as the corrected position of $x_i^{(j)}$.

### 4.4 Performance Evaluation

**Table 1:** *Statistics of the number of operations of building power cells for the three examples shown in Fig. 2. Note that we take* $\|\nabla F\| \le 10^{-8}$ *as the termination condition.*

| Examples | # of building power cells | | Timing (s) | |
|---|---|---|---|---|
| | [de Goes et al. 2012] | Ours | [de Goes et al. 2012] | Ours |
| 100 sites | 4091 | 279 | 20.8 | 1.6 |
| 500 sites | 2944 | 464 | 246.7 | 39.5 |
| 1000 sites | 4307 | 471 | 556.0 | 58.4 |

We implemented and experimented with our CPD algorithm on a computer with a 64-bit version of Win7 system, a 3.07 GHz Intel(R) Core(TM) i7 CPU and 6 GB memory. The coding language is C++. Now we shall compare our CPD algorithm with the state of the art [de Goes et al. 2012]. For the examples shown in Fig. 2, we take $\|\nabla F\| \le 10^{-8}$ as the termination condition. For the example of $n = 100$ sites (see the left column), we assume that the domain $[0,1]\times[0,1]$ is associated with a uniform density. For the example of $n = 500$ sites (see the middle column), the density function at $x = (x^{(1)}, x^{(2)})$ is set to $\rho(x) = 0.1 + x^{(1)}$. For the example of $n = 1,000$ sites (see the right column), the density is set to $\rho(x) = e^{-8(x^{(1)}-0.5)^2 - 8(x^{(2)}-0.5)^2}$. Fig. 2(j-l) shows that our algorithm exhibits a faster convergence. In detail, we observe that (1) the resultant cells by our algorithm are of more compact shapes than [de Goes et al. 2012] after the same number of iterations; (2) the cost by our algorithm reaches the stable state earlier than [de Goes et al. 2012]; and (3) the gradient of the cost function decreases more sharply. Note that the horizontal axis of Fig. 2(g-l) denotes the timing.




(a) The 76-th step (b) The 171-st step (c) The 131-st step

(d) The 76-th step (e) The 171-st step (f) The 131-st step

(g) 100 sites (h) 500 sites (i) 1000 sites

(j) 100 sites (k) 500 sites (l) 1000 sites

**Figure 2:** *Performance comparison with [de Goes et al., 2012]. The first row: snapshots of our algorithm. The second row: snapshots of [de Goes et al., 2012]. The third row: the plots of the cost function. The last row: the plots of the gradient of the cost function. Note that the horizontal axis in (g-l) denotes the timing (s).*

Since the operation of building power cells is the most time-consuming part, Table 2 shows the number of building power cells to achieve the same specified threshold $10^{-8}$, as well as the total computation time. From the statistics we can see that our reduced gradient method has empirical super-linear convergence and significantly outperforms [de Goes et al. 2012].

### 4.5 General Cost Functions

Many real-life application occasions can be formulated as a CPD problem. Consider deploying some service centers (e.g. postal offices or medical centers) to serve the residents of a city, where each center has a capacity constraint. It's better to take the Euclidean distance, rather than the squared distance, as the service cost kernel. Another example is when the domain is a curved surface where we often take geodesic distances as the cost kernel. As pointed out in Section 3, our algorithmic framework supports general kernels. Here we use six typical kernels (see Fig. 3) to exhibit the extensibility of our algorithm, including

- $\diamond\, d(x, x_i) = (x^{(1)} - x_i^{(1)})^2 + (x^{(2)} - x_i^{(2)})^2$ in $\mathbb{R}^2$;
- $\diamond\, d(x, x_i) = \sqrt{(x^{(1)} - x_i^{(1)})^2 + (x^{(2)} - x_i^{(2)})^2}$ in $\mathbb{R}^2$;
- $\diamond\, d(x, x_i) = (x^{(1)} - x_i^{(1)})^4 + (x^{(2)} - x_i^{(2)})^4$ in $\mathbb{R}^2$;
- $\diamond$ Geodesic distance on mesh surfaces;
- $\diamond$ Squared geodesic distance on mesh surfaces.
- $\diamond$ Cubic geodesic distance on mesh surfaces.

In Fig. 3, the first row shows that CPD has freedom degrees to support user-specified capacity constraints while CVT cannot control cell sizes. The second row gives three CPDs with different cost kernels assuming the same uniform density. The third row shows the case of a Gaussian density function. The last row exhibits three CPDs on curved surfaces. When handling mesh surfaces, each site $x_i$ is constrained to move along the tangent plane and projected onto the surface before evaluating the cost function. Furthermore, to find the optimal CPD partition, we have to frequently query the geodesic distances [Surazhsky and Surazhsky 2005; Xin and Wang 2009] between two points, which is computationally expensive. Due to this limitation, each of the input meshes in Fig. 3(i-k) contains about 1K triangle faces in our experiments. With the termination condition $\|\nabla_W F(X)\|$ being less than $10^{-4}$, we need about 15 iterations and 5 minutes to deal with each example. (Note that the partition of the surface is approximate and the evaluation of the cost function is approximate as well. That's why we set the tolerance to be $10^{-4}$, rather than $10^{-8}$.) However, this can be greatly accelerated using several recent methods for geodesic computation [Crane et al. 2013; Ying et al. 2013b; Ying et al. 2013a]. With a faster geodesic computation querier, we believe that it has a great potential for many geometry processing tasks.

## 5 Applications

In this section, we shall use it for three problems: displacement interpolation of function distributions, blue-noise point sampling, and optimal convex decomposition of 2D domains.

### 5.1 Displacement Interpolation

The traditional Wasserstein barycenter problem, i.e. displacement interpolation, provides a generic method for interpolating between distributions or functions based on advection instead of blending [Bonneel et al. 2011]. It has been found very useful in many applications such as shape interpolation [Solomon et al. 2015; Oberman and Ruan 2015] and texture synthesis [Bonneel et al. 2015]. The displacement interpolation is to find a target probability density $\rho$ defined by

$$\min_{\rho} \sum_{k=1}^{m} \lambda_k \mathbf{W}_2^2(\rho_k, \rho), \tag{10}$$

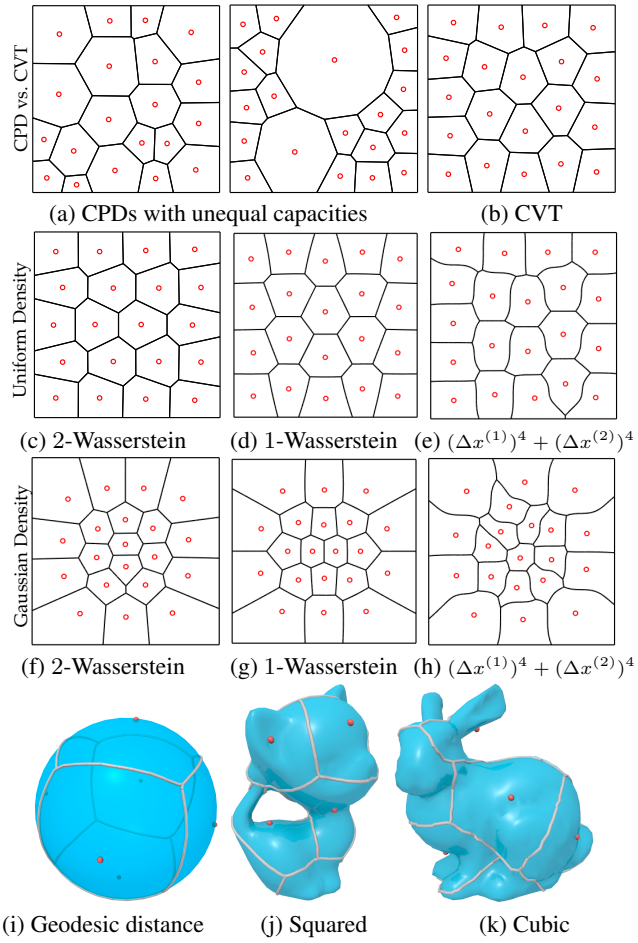



**Figure 3:** *Our algorithm can be extended to general cost kernels. The first row shows that CPD can generate compact cells with user-specified capacity constraints while CVT cannot. For the other examples, the capacity constraints divide the domain equally.*

where $\{\rho_k\}_{k=1}^m$ are a collection of given probability densities and $\{\lambda_k\}_{k=1}^m$ are user-specified weights for computing the average probability density with regard to the 2-Wasserstein distance $\mathbf{W}(\cdot,\cdot)$. Recently many research works [Cuturi and Doucet 2013; Benamou et al. 2015; Solomon et al. 2015] suggested an energy term with an additional entropy regularization, bringing convexity to the problem and make it easier to solve. However, such an entropy term would result in an over-smoothed result in exchange for better efficiency. Besides, although lots of works proposed to compute the traditional Wasserstein barycenter distributions, few works aim at the semi-discrete setting. Here we propose a framework that supports multiple input distributions, each of which is represented by a collection of sites associated with capacities or a continuous probability density function. Our algorithm produces adaptively distributed samples to encode the final distribution.

Consider the following semi-discrete minimization problem, denoting $R_k = \{R_k^i\}_{i=1}^n : \Omega \to \{x_i\}_{i=1}^n$ as an arbitrary assignment between the support of $\rho_k$ and $X$

$$\begin{cases} \min\limits_{X} \sum\limits_{k=1}^{m} \lambda_k \mathbf{W}_2^2(\rho_k, X, R_k), \\ s.t. \quad \int_{R_k^i} \rho_k \, \mathrm{d}x = c_i, \text{ for all } i, k. \end{cases} \tag{11}$$

The constraint term in (11) forces the mass transported to $x_i$ under $\rho_k$ to be equal to a given $c_i$. Based on the conclusion of [Aurenhammer et al. 1998], each assignment $R_k$ minimizing the 2-Wasserstein cost under the capacity constraint should be a power diagram, and thus Eqn. (11) can be further reduced to the following minimization problem [de Goes et al. 2012]:

$$\underset{X}{\text{Minimize}}\ F_b(X) = \sum_{k=1}^{m} \lambda_k \left( \sum_{i=1}^{n} \int_{GD(W)_k^i} \rho_k ||x - x_i||^2 \, \mathrm{d}x - \sum_{i=1}^{n} w_i (|GD(W)_k^i| - c_i) \right), \tag{12}$$

with

$$\nabla_{x_i} F_b(x_i) = \sum_{k=1}^{m} 2 c_i \lambda_k (x_i - b_k^i), \tag{13}$$

where $b_k^i$ is the centroid of the $i$-th power cell under the $k$-th distribution. In implementation, the optimization procedure can follow a similar procedure to computing CPDs, presented in Algorithm 2.

---

**Input:** densities $\{\rho_k\}_{k=1}^m$, weights $\{\lambda_k\}_{k=1}^m$, capacities $\{c_i\}_{i=1}^n$, and a threshold $\epsilon$.
**Output:** $\{x_i\}_{i=1}^n$.
**repeat** *//L-BFGS*
  **for** $k = 1, 2, 3, \cdots, m$
    *//for the k-th distribution*
    Optimize $W_k$ until each power cell amounts to the given capacity
    Compute the centroids of the power diagram under $k$-th distribution
  **end**
  Compute the cost function $F_b(X)$ in Eqn. (12)
  Compute the gradient w.r.t. $X$; See Eqn. (13)
  Update $X$ by step search
**until** $||\nabla_X F_b(X)|| < \epsilon$

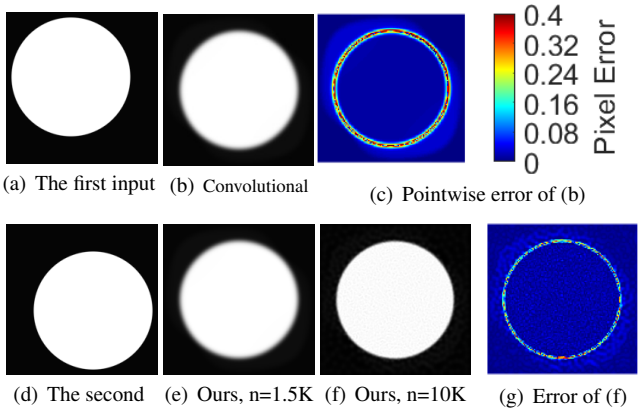
---

**Algorithm 2:** Capacity Constrained Wasserstein Barycenter with LBFGS.

Fig. 4 shows the result from two identical circular spots. The ideal distribution when the weights of these two inputs are both 0.5 should be of the same circular shape at the middle position. We compare our result with that by [Solomon et al. 2015] and visualize the color-coded errors in Fig. 4(c&g). The resolutions of both input images are $200 \times 200$ in this example. We can see that when $n = 1.5$K, we achieve an error of RMSE=0.078 in about 30 seconds, which is at the same efficiency/accuracy level with [Solomon et al. 2015]; But when $n$ increases to 10K, the error of RMSE decreases to 0.0441 at the cost of more computation time - about 30 minutes for this example (the inefficiency is due to the density setting where lots of density values are very close to 0). Note that for comparison purpose, Fig. 4(e-f) are produced by an adaptive kernel estimation as the postprocessing [Abramson 1982]. Fig. 5 shows the results of 2-Wasserstein barycenters with multiple inputs.

### 5.2 Blue Noise Sampling

Stochastic point distributions with blue noise properties are widely used for various applications in computer graphics [Yan et al. 2015]. de Goes et al. [2012] presented a fast algorithm to generate high-quality blue noise point distributions by finding a stationary point of the objective function defined in Eqn. (5). However, extremizing




(a) The first input (b) Convolutional (c) Pointwise error of (b)

(d) The second (e) Ours, n=1.5K (f) Ours, n=10K (g) Error of (f)

**Figure 4:** *Displacement interpolation of two identical circular spots. (a)&(d): the two inputs of circular spots; (b)&(c): interpolated distribution by [Solomon et al. 2015] and the corresponding color-coded point-wise error (RMSE=0.0794), generated in 30 seconds (using the Matlab code provided by the authors of [Solomon et al. 2015]). (e-g) our results for $n$=1.5K and $n$=10K. When $n = 1.5$K, we achieve RMSE=0.078 in about 30 seconds; When $n$ increases to 10K, the error of RMSE decreases to 0.0441 at the cost of increased computation time - about 30 minutes for this example.*

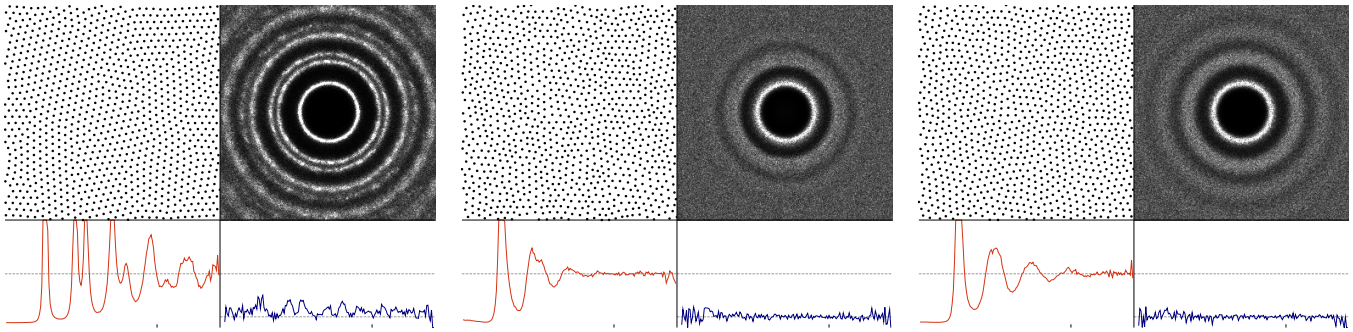
the objective function does not prevent the regular patterns from emerging. In fact, the hexagonal grid patterns are the solutions to the extremization problem. de Goes et al. [2012] propose to jitter the points and run the optimization again when local regularities are detected. It is expected that a "shallower" extremum of the objective function which contains no regular patterns is obtained by a local searching algorithm. However, the optimization method of [de Goes et al. 2012] used a very loose stopping criteria, which usually terminates in 10 iterations, well before convergence. A more faithful local extremum can be efficiently achieved by our reduced gradient method that optimizes the same function. Experimental results show that "extreme" optimization of the objective function always results in structured point distributions even starting from a random point distribution (see Fig. 6(a)).

It is worth noting that all point samples are enforced to have the same capacity $c$ in the existing methods based on capacity-constrained Voronoi tessellation [Balzer et al. 2009; Li et al. 2010; Chen et al. 2012; de Goes et al. 2012]. To prevent regular patterns in the point distributions, Zhang et al. [2016] suggested using non-uniform capacity constraints and dynamically adjust them during optimization. Inspired by this, we propose to jitter the capacity constraints for the points before optimization. A random noise with a maximum magnitude $\tau$ is used to jitter the capacity constraint for each point. The new capacity constraint $c_i^*$ for each point is now specified by the following equation:

$$c_i = (1 + \tau \cdot rand(-1, 1))c, \quad c_i^* = \frac{c_i}{\sum c_i} nc,$$

where $n$ is the sample number, and the second formula is to normalize the constraints so that they all sum to the total capacity of the domain.

Fig. 6(a) and 6(c) show point sets generated by extremizing the objective function with uniform capacity constraints and the jittered capacity constraints, respectively. A strict convergence threshold ($\|\nabla_X F\| \leq 10^{-7}$) is used for both cases. We can see that the jittered capacity constraints avoid regularity in the resultant point



(a) $\tau = 0$, $\|\nabla_X F\| < 10^{-7}$ (b) $\tau = 0.4$, $\|\nabla_X F\| < 10^{-7}$ (c) $\tau = 0.2$, $\|\nabla_X F\| < 10^{-5}$

**Figure 6:** *Sampling results with different $\tau$'s over a periodic domain with constant density. The blue noise properties can be analyzed through comparing the averaged periodograms, radial mean power (red curve), and anisotropy (blue curve).*

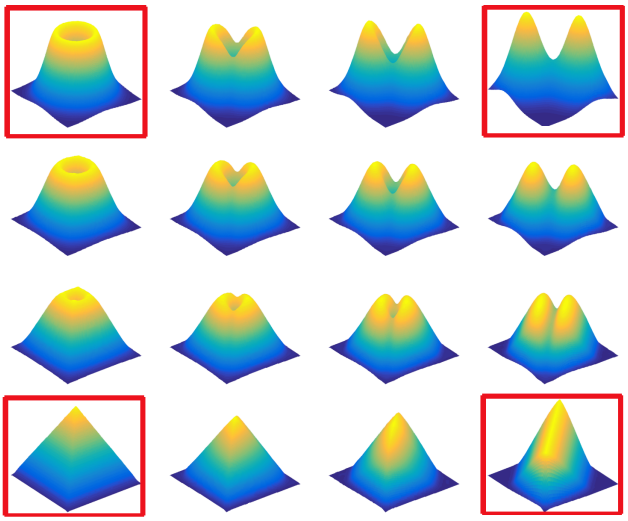


**Figure 5:** *The interpolation by our method of the four inputs (500×500 in size) at the four corners of the array (highlighted in red). Each output frame is generated by 10K sites in about 15 minutes.*

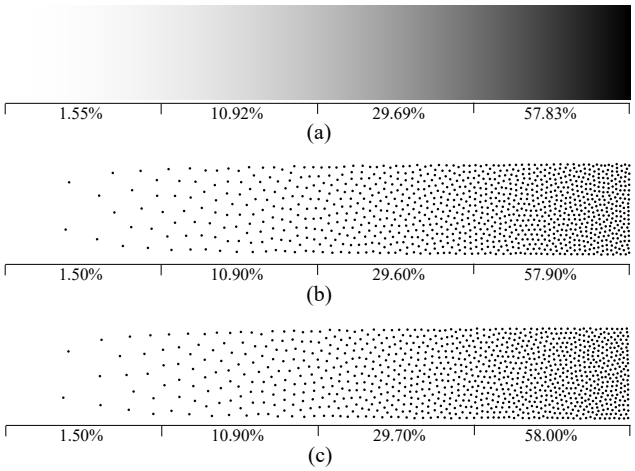



**Figure 7:** *Sampling of a quadratic density function with 1000 points, the percentages indicate point density in each quarter: (a) input quadratic density function; (b) sampling result by [de Goes et al. 2012]; (c) our result.*

distribution, producing high-quality blue noise. For the sake of fast computation, we set $\tau$ as a small number 0.2 and use a loose stopping condition with $\|\nabla_X F\| \leq 0.05\sqrt{nm^3}$ in practice. Under this setting, our optimization method is able to converge after a dozen of iterations, and generate high-quality blue noise samplings, as shown in Fig. 6(c).

As we jitter the capacity constraints for sampling points, an immediate concern is that the point distribution may not adapt to the given density function precisely. Fig. 7 shows a sampling result generated by our method for an intensity ramp. We count the point density in each quarter of the ramp, and compare it to the reference density of the ramp. We can see that jittering the capacity constraints break local structured patterns while returns good adaption to the given density function.

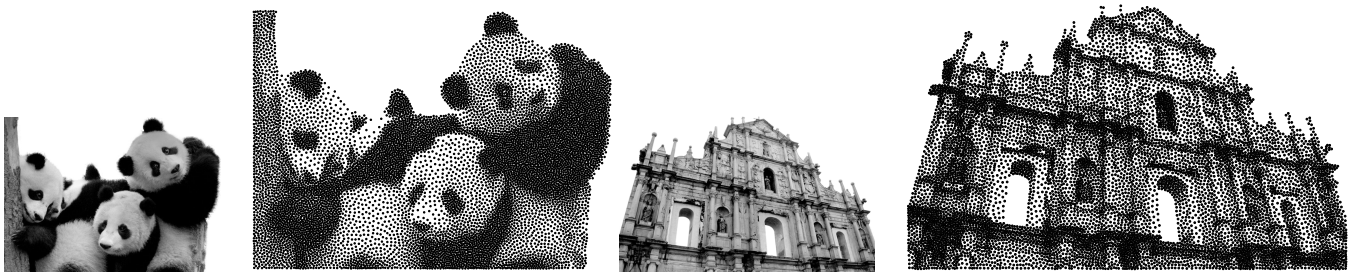
Fig. 8 shows two image stippling results generated by our optimization framework. The sampling points $\{x_i\}_{i=1}^n$ are initialized randomly according to the piecewise-constant density function defined by the gray scale of the input image. Integrations involved in our formulation are exactly computed by constructing the pixel-cell intersections, as done in [de Goes et al. 2012].

### 5.3 Centroidal Convex Decomposition

Another important problem in geometric computing, especially in mesh generation, is the convex decomposition of a geometric domain [Greene 1983; Ghosh et al. 2013], that is, decomposing a general non-convex domain into the union of a number of convex cells. The problem is relevant to numerous applications in pattern recognition, Minkowski sum computation, motion planning, computer graphics, and origami folding [Ghosh et al. 2013]. In this section we use the optimal capacity-constrained partition to provide a novel and effective solution to the convex decomposition of a 2D planar region.

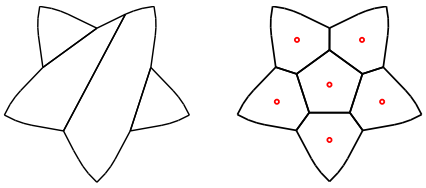
Existing algorithms for convex decomposition usually minimize the number of convex components; See Fig. 9(a). In this paper, we propose a different type of convex decomposition, called the *optimal convex decomposition* that meets the following criteria: (1) each decomposed cell is convex; (2) the decomposed cells have specified areas; and (3) the decomposed cells are of compact shapes. Fig. 9(b) shows such a convex decomposition. Note that the term "optimal" is used to emphasize shape regularity rather than the number of decomposed cells.

Suppose that the boundary of the input 2D domain to be decom-



(a) (b) (c) (d)

**Figure 8:** *Image stippling. (a)&(c) Input images; (b)&(d) stippling results with 10K points generated by our optimization framework in 129 seconds and 146 seconds, respectively.*



(a) (b)

**Figure 9:** *Convex decomposition of a star shape. The optimal convex decomposition (b) is much different than the traditional convex decomposition (a).*

posed is defined by a polygonal curve. It is observed that simply running a CPD algorithm would cause a concave vertex on the boundary to be located in the interior of a single CPD cell, resulting in a non-convex cell (see Fig. 10(a)). Our key idea is to define all concave boundary vertices as *virtual sites* that are fixed and associated with a sufficiently small capacity constraint. Thus, the new cost function becomes

$$F(X,W)=\sum_{i=1}^{n+k}\int_{R_i^W}\rho(x)\|x-x_i\|^2\mathrm{d}x \tag{14}$$

subject to

$$c_i=\int_{R_i^W}\rho(x)\mathrm{d}x(\triangleq m_i), 1\le i\le n+k, \tag{15}$$

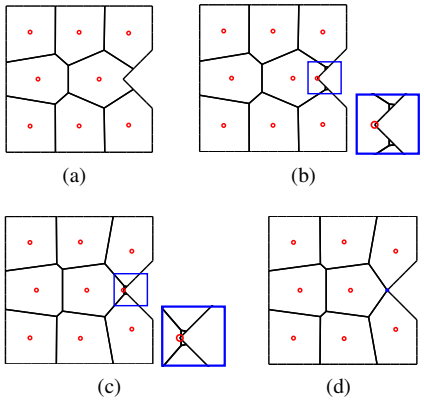
where $n$ is the number of variable sites and $k$ is the number of virtual sites on the boundary. Here $\{m_i, 1\le i\le n\}$ is a set of capacity constraints for the moveable sites, and the number of such sites, $n$, is usually specified by the user. Let $m_{\text{average}}$ be the average value of $\{m_i, 1\le i\le n\}$. The capacity constraint for the virtual sites, say, $m_{\text{virtual}}$, is given by

$$m_{\text{virtual}}=\epsilon\times m_{\text{average}}, \tag{16}$$

where $\epsilon$ is set to be $10^{-3}$ in our experiments.

Furthermore, we assign a large multiplicative penalty to the cost terms defined by the virtual sites such that those small-sized power cells can be made compact and stay around the virtual sites (see Fig. 10(b)). The new cost function is as follows.

$$\begin{aligned}F(X,W)=&\sum_{i=1}^{n+k}\int_{R_i^W}\rho(x)\|x-x_i\|^2\mathrm{d}x\\&+\lambda\sum_{i=n+1}^{n+k}\int_{R_i^W}\rho(x)\|x-x_i\|^2\mathrm{d}x\end{aligned} \tag{17}$$



(a) (b) (c) (d)

**Figure 10:** *Computation of the optimal convex decomposition: (a) Generally the concave points may be covered totally by a single CPD cell, giving rise to a non-convex cell. (b) Even if we enforce a small capacity for the virtual site at the concave point, the power cell stay non-convex after optimization. (c) After adding a large penalty coefficient $\lambda$ for the virtual sites, the minimization makes all the normal power cells almost convex. (d) By removing the virtual sites from the results in (c), we obtain a convex decomposition of the domain.*

Empirically we take the penalty coefficient to be

$$\lambda=\frac{m_{\text{normal}}}{m_{\text{average}}}. \tag{18}$$

After the above cost function is minimized (see Fig. 10(c)), we remove those virtual sites to get a clean convex decomposition in which each concave point is located on the common boundary of two adjacent convex power cells (see Fig. 10(d)). Fig. 11 demonstrates three examples of convex decomposition produced by our method based on CPDs with various density settings. We note that the density value should be larger in narrow regions and around concave points, generally implying a denser site distribution, to achieve a successful decomposition.

## 6 Conclusions and Future Works

In this paper, a super-linear convergent method is developed for computing the capacity-constrained centroidal power diagram. Experimental results show that our algorithm outperforms the existing method. Due to its effectiveness, we apply our fast CPD algorithm to three problems in computer graphics and geometric processing: displacement interpolation, blue-noise point sampling, and optimal

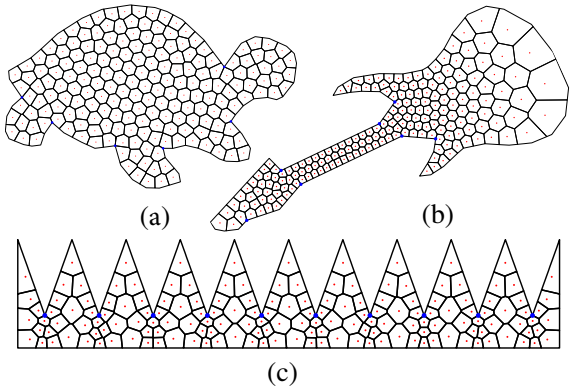



**Figure 11:** *Convex decomposition of complex 2D shapes. The blue dots denote the user-specific concave points on the given boundary. For the Tortoise shape, we set a uniform density distribution. For the Guitar shape (b) and the Gear-Teeth shape (c), the density is higher in narrow regions or around the user-specified concave points.*

convex decomposition of 2D domains. Furthermore, we show that the proposed algorithm can be extended to the capacity-constrained optimal partition with respect to general cost functions.

Our algorithm, in its current form, has several limitations. First, the capacity constraints are required to be met throughout the optimization process, which may be a cause of inefficiency. It is therefore valuable to develop an optimization technique that can accommodate an infeasible start. Second, the optimal mass transport problem in 3D space has been studied [Su et al. 2016; Lévy 2014]. We are therefore interested in extending our method to 3D or spaces of higher dimensions. Finally, we have only studied optimal convex decomposition of a 2D domain. The optimal convex decomposition of 3D domain is more relevant problem in practice and is much more challenging that the 2D version, so would need further study.

## Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments. This work is supported by NSF of China (61332015, 61300168, 61472332, 61100105, 61572021, 61272019), GRF of RGC (17263316), National 973 Program (2015CB352501) and SZ-fund (SIRI0404201431). It is also partially supported by ANR grants BECASIM and MAGA.

## 7 Appendix

**Theorem 1**: *For any fixed sites $X = \{x_i\}$, the capacity-constrained partition $R = \{R_i\}_{i=1}^n$ of the domain $\Omega$ that minimizes the cost function in Eqn. (4) is the one such that the boundary of any two adjacent regions $R_a$ and $R_b$, which are associated with the sites $a$ and $b$ in $X$ respectively, is define by the equation $d(x,a) - w_a = d(x,b) - w_b$ for some constants $w_a$ and $w_b$.*

PROOF: We shall prove the theorem by contradiction. Let us denote partition that minimizes the cost function in Eqn. (4). Assume that the theorem is not true. Then there exist two neighboring sites $a$ and $b$ with the shared boundary $\Gamma$ of $R_a$ and $R_b$ such that there exist two distinct points $x_0, y_0 \in \Gamma$ for which

$$d(x_0,a) - d(x_0,b) \neq d(y_0,a) - d(y_0,b).$$

Without loss of generality, suppose that

$$d(x_0,a) - d(x_0,b) > d(y_0,a) - d(y_0,b)$$

Let $B(x,\delta)$ denote the ball centered at $x$ with radius $\delta$. Since $d(x,y)$ is continuous with respect to $x$ and $y$, there exist two balls $B(x_0,\delta_0)$ and $B(y_0,\delta_0)$ with sufficiently small $\delta_0 > 0$ such that

$$d(u,a) - d(u,b) > d(v,a) - d(v,b)$$

for all $u$ in $B(x_0,\delta_0)$ and all $v$ in $B(y_0,\delta_0)$.

Because the boundary $\Gamma$ goes through the centers of $B(x_0,\delta_0)$ and $B(y_0,\delta_0)$, there exist two balls $B_x \equiv B(x_1,\delta_1)$ and $B_y \equiv B(y_1,\delta_2)$ with sufficiently small $\delta_1, \delta_2 > 0$ such that $B_x$ and $B_y$ have equal capacities (i.e. $\int_{u\in B_x} \rho(u)\mathrm{d}\sigma = \int_{u\in B_y} \rho(v)\mathrm{d}\sigma$) and that $B_x \subset R_a \cap B(x_0,\delta_0)$ and $B_y \subset R_b \cap B(y_0,\delta_0)$.

Since $B_x \subset B(x_0,\delta_0)$ and $B_y \subset B(y_0,\delta_0)$, we have

$$d(u,a) - d(u,b) > d(v,a) - d(v,b)$$

for all $u$ in $B_x$ and all $v$ in $B_y$. It follows that

$$\int_{u\in B_x} \rho(u)(d(u,a) - d(u,b))\mathrm{d}\sigma > \int_{v\in B_y} \rho(v)(d(v,a) - d(v,b))\mathrm{d}\sigma,$$

which, after rearranging, yields

$$\begin{aligned} &\int_{u\in B_x} \rho(u)d(u,a)\mathrm{d}\sigma + \int_{v\in B_y} \rho(v)d(v,b)\mathrm{d}\sigma \\ > &\int_{u\in B_x} \rho(u)d(u,b)\mathrm{d}\sigma + \int_{v\in B_y} \rho(v)d(v,a)\mathrm{d}\sigma \end{aligned} \quad (19)$$

On the other hand, it follows from the optimality of $R$, in particular, the optimality of $R_a$ and $R_b$ that swapping $B_x$ and $B_y$ between $R_a$ and $R_b$, while maintaining the capacities of $R_a$ and $R_b$, does not reduce the total cost. Therefore

$$\begin{aligned} &\int_{u\in B_x} \rho(u)d(u,a)\mathrm{d}\sigma + \int_{v\in B_y} \rho(v)d(v,b)\mathrm{d}\sigma \\ \leq &\int_{u\in B_x} \rho(u)d(u,b)\mathrm{d}\sigma + \int_{v\in B_y} \rho(v)d(v,a)\mathrm{d}\sigma \end{aligned} \quad (20)$$

Clearly, Eqn. 20 and Eqn. 19 contradict. Therefore we conclude that

$$d(x_0,a) - d(x_0,b) = d(y_0,a) - d(y_0,b),$$

for any two arbitrary points $x_0$ and $y_0$ on the boundary $\Gamma$. Hence, we have proved that the boundary curve $\Gamma$ separating $R_a$ and $R_b$ is defined by an equation of the form

$$d(x,a) - w_a = d(x,b) - w_b \quad (21)$$

for some constants $w_a$ and $w_b$. This completes the proof.

## References

ABRAMSON, I. S. 1982. On bandwidth variation in kernel estimates-a square root law. *Annals of Statistics 10*, 4, 1217–1223.

AGUEH, M., AND CARLIER, G. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis 43*, 2, 904–924.

ATKINSON, K. E. 2008. *An introduction to numerical analysis*. John Wiley & Sons.

AURENHAMMER, F., AND KLEIN, R. 2000. Voronoi diagrams. *Handbook of computational geometry 5*, 201–290.

AURENHAMMER, F., HOFFMANN, F., AND ARONOV, B. 1998. Minkowski-type theorems and least-squares clustering. *Algorithmica 20*, 1, 61–76.

AURENHAMMER, F. 1987. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing 16*, 1, 78–96.

BALZER, M., AND HECK, D. 2008. Capacity-constrained Voronoi diagrams in finite spaces. In *Voronoi Diagrams in Science and Engineering*.

BALZER, M., DEUSSEN, O., AND LEWERENTZ, C. 2005. Voronoi treemaps for the visualization of software metrics. In *Proceedings of the 2005 ACM symposium on Software visualization*, 165–172.

BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graph. 28*, 3, 86:1–8.

BENAMOU, J., FROESE, B. D., AND OBERMAN, A. M. 2014. Numerical solution of the optimal transportation problem using the Monge-Ampère equation. *J. Comput. Physics 260*, 107–126.

BENAMOU, J.-D., CARLIER, G., CUTURI, M., NENNA, L., AND PEYRÉ, G. 2015. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing 37*, 2, A1111–A1138.

BONNEEL, N., VAN DE PANNE, M., PARIS, S., AND HEIDRICH, W. 2011. Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph. 30*, 6, 158.

BONNEEL, N., RABIN, J., PEYRÉ, G., AND PFISTER, H. 2015. Sliced and radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision 51*, 1, 22–45.

BORGWARDT, A., BRIEDEN, A., AND GRITZMANN, P. 2014. Geometric clustering for the consolidation of farmland and woodland. *Mathematical Intelligencer 36*, 2, 37–44.

BOURNE, D., AND ROPER, S. 2014. Centroidal power diagrams, Lloyd's algorithm and applications to optimal location problems. *arXiv preprint arXiv:1409.2786*.

BOURNE, D., PELETIER, M., AND ROPER, S. 2014. Hexagonal patterns in a simplified model for block copolymers. *SIAM Journal on Applied Mathematics 74*, 5, 1315–1337.

BRENIER, Y. 1991. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics 44*, 4, 375–417.

CHEN, Z., YUAN, Z., CHOI, Y.-K., LIU, L., AND WANG, W. 2012. Variational blue noise sampling. *IEEE Transactions on Visualization and Computer Graphics 18*, 10, 1784–1796.

CORTÉS, J. 2010. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control 55*, 3, 749–754.

CRANE, K., WEISCHEDEL, C., AND WARDETZKY, M. 2013. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5, 13–15.

CUTURI, M., AND DOUCET, A. 2013. Fast computation of Wasserstein barycenters. *arXiv preprint arXiv:1310.4375*.

DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. 2012. Blue noise through optimal transport. *ACM Trans. Graph. 31*, 6, 171.

DOBROVOLSKY, V. A. 1972. Sur l'histoire de la classification des points singuliers des équations différentielles. *Revue d'histoire des sciences 25*, 1, 3–11.

DU, Q., FABER, V., AND GUNZBURGER, M. 1999. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Rev. 41*, 4, 637–676.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of SIGGRAPH 2001*, 15–22.

GANGBO, W., AND CANN, R. M. 1996. The geometry of optimal transportation. *Acta Mathematica*.

GHOSH, M., AMATO, N. M., LU, Y., AND LIEN, J. M. 2013. Fast approximate convex decomposition using relative concavity. *Computer-Aided Design 45*, 2, 494–504.

GREENE, D. H. 1983. The decomposition of polygons into convex parts. *Computational Geometry 1*, 235–259.

GU, X., LUO, F., SUN, J., AND YAU, S. T. 2013. Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampère equations. *Mathematical Methods in Solid State & Superfluid Theory 61*, 5, 1–45.

JOBSON, D. J., RAHMAN, Z.-U., AND WOODELL, G. A. 1996. Retinex image processing: Improved fidelity to direct visual observation. In *Color and Imaging Conference*, Society for Imaging Science and Technology, 124–125.

KANTOROVICH, L., AND GAVURIN, M. 1949. Application of mathematical methods to problems of analysis of freight flows. *Problems of raising the efficiency of transport performance*, 110–138.

KARTCH, D. 2000. *Efficient Rendering and Compression for Full-Parallax Computer-Generated Holographic Stereograms*. PhD thesis, Cornell University.

KITAGAWA, J., MÉRIGOT, Q., AND THIBERT, B. 2016. A Newton algorithm for semi-discrete optimal transport. *CoRR abs/1603.05579*.

LANDIS, H., 2002. Global illumination in production. ACM SIGGRAPH 2002 Course #16 Notes.

LASDON, L. S., FOX, R. L., AND RATNER, M. W. 1974. Nonlinear optimization using the generalized reduced gradient method. *RAIRO - Operations Research - Recherche Opérationnelle 8*, 3, 73–103.

LEAL, L. G. 2007. Advanced transport phenomena: fluid mechanics and convective transport processes. *Scitech Book News*, 3.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project. In *Proceedings of SIGGRAPH 2000*, 131–144.

LÉVY, B. 2014. A numerical algorithm for $l_2$ semi-discrete optimal transport in 3D. *Esaim Mathematical Modelling & Numerical Analysis 49*, 6.

LI, H., NEHAB, D., WEI, L.-Y., SANDER, P. V., AND FU, C.-W. 2010. Fast capacity constrained Voronoi tessellation. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, 13:1–13:1.

LIU, D. C., AND NOCEDAL, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming: Series A and B 45*, 3, 503–528.

LIU, Y., WANG, W., LÉVY, B., SUN, F., YAN, D.-M., LU, L., AND YANG, C. 2009. On centroidal Voronoi tessellation: energy

smoothness and fast computation. *ACM Trans. Graph. 28*, 4, 101:1–101:17.

LLOYD, S. 2006. Least squares quantization in PCM. *IEEE Trans. Inf. Theor. 28*, 2, 129–137.

LU, L., CHOI, Y.-K., WANG, W., AND KIM, M.-S. 2007. Variational 3D shape segmentation for bounding volume computation. *Computer Graphics Forum 26*, 3, 329–338.

LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. 2014. Build-to-last: strength to weight 3D printed objects. *ACM Trans. Graph. 33*, 4, 70–79.

MA, X.-N., TRUDINGER, S. N., AND WANG, X.-J. 2005. Regularity of potential functions of the optimal transportation problem. *Archive for Rational Mechanics and Analysis 177*, 2, 151–183.

MCCANN, R., AND GUILLEN, N. 2010. Five lectures on optimal transportation: Geometry, regularity and applications. In *Summer School: New Vistas in Image Processing and Partial Differential Equations*.

MÉRIGOT, Q. 2011. A multiscale approach to optimal transport. *Computer Graphics Forum 30*, 5, 1583–1592.

MILGROM, P., AND SEGAL, I. 2002. Envelope Theorems for Arbitrary Choice Sets. *Econometrica 70*, 2, 583–601.

MONGE, G. 1781. *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale.

MORÉ, J. J., AND THUENTE, D. J. 1994. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software 20*, 3, 286–307.

OBERMAN, A., AND RUAN, Y. 2015. An efficient linear programming method for optimal transportation. *arXiv:1509.03668*.

OKABE, A., BOOTS, B., AND SUGIHARA, K. 1992. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Inc., New York, NY, USA.

PARK, S. W., LINSEN, L., KREYLOS, O., OWENS, J. D., AND HAMANN, B. 2006. Discrete Sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics 12*, 2, 243–253.

PARKE, F. I., AND WATERS, K. 1996. *Computer Facial Animation*. A. K. Peters.

PATEL, R., FRASCA, P., AND BULLO, F. 2014. Centroidal area-constrained partitioning for robotic networks. *Journal of Dynamic Systems, Measurement, and Control 136*, 3, 031024.

PELLACINI, F., VIDIMČE, K., LEFOHN, A., MOHR, A., LEONE, M., AND WARREN, J. 2005. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Trans. Graph. 24*, 3, 464–470.

POGORELOV, A. V. 1994. Generalized solutions of Monge-Ampère equations of elliptic type. In *A tribute to Ilya Bakelman, volume 3 of Discrourses Math. Appl*, 47–50.

RABIN, J., PEYRÉ, G., DELON, J., AND BERNOT, M. 2011. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision*. Springer, 435–446.

SAKO, Y., AND FUJIMURA, K. 2000. Shape similarity by homotropic deformation. *The Visual Computer 16*, 1, 47–61.

SANTAMBROGIO, F. 2015. *Optimal Transport for Applied Mathematicians*. Springer International Publishing.

SOLOMON, J., DE GOES, F., PEYRÉ, G., CUTURI, M., BUTSCHER, A., NGUYEN, A., DU, T., AND GUIBAS, L. 2015. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph. 34*, 4, 66.

SU, Z., WANG, Y., SHI, R., ZENG, W., SUN, J., LUO, F., AND GU, X. 2015. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis & Machine Intelligence 37*, 11, 2246–2259.

SU, K., CHEN, W., LEI, N., CUI, L., JIANG, J., AND GU, X. D. 2016. Measure controllable volumetric mesh parameterization. *Computer-Aided Design 78*, 188–198.

SURAZHSKY, V., AND SURAZHSKY, T. 2005. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph 24*, 553–560.

VILLANI, C. 2008. *Optimal transport: old and new*, vol. 338. Springer Science & Business Media.

XIN, S.-Q., AND WANG, G.-J. 2009. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM Trans. Graph. 28*, 4, 104.

XIN, S.-Q., WANG, X., XIA, J., MUELLER-WITTIG, W., WANG, G.-J., AND HE, Y. 2013. Parallel computing 2D Voronoi diagrams using untransformed sweepcircles. *Computer-Aided Design 45*, 2, 483–493.

YAN, D.-M., GUO, J.-W., WANG, B., ZHANG, X.-P., AND WONKA, P. 2015. A survey of blue-noise sampling and its applications. *Journal of Computer Science and Technology 30*, 3, 439–452.

YEE, Y. L. H. 2000. *Spatiotemporal sensistivity and visual attention for efficient rendering of dynamic environments*. Master's thesis, Cornell University.

YING, X., WANG, X., AND HE, Y. 2013. Saddle vertex graph (svg): A novel solution to the discrete geodesic problem. *ACM Trans. Graph. 32*, 6, 2504–2507.

YING, X., XIN, S. Q., AND HE, Y. 2013. Parallel Chen-Han (PCH) algorithm for discrete geodesics. *ACM Trans. Graph. 33*, 1, 57–76.

ZHANG, S., GUO, J., ZHANG, H., JIA, X., YAN, D. M., YONG, J., AND WONKA, P. 2016. Capacity constrained blue-noise sampling on surfaces. *Computers & Graphics 55*, C, 44–54.

ZHAO, X., SU, Z., GU, X. D., KAUFMAN, A., SUN, J., GAO, J., AND LUO, F. 2013. Area-preservation mapping using optimal mass transport. *IEEE Transactions on Visualization & Computer Graphics 19*, 12, 2838–47.

ZHOU, Y., JU, L., AND WANG, S. 2015. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. *IEEE Transactions on Image Processing 24*, 11, 3834–45.