CENTROIDAL VORONOI TESSELLATIONS

or

color printers, mailboxes, Homer Simpson, fish, etc.

Max Gunzburger
Florida State University
gunzburg@csit.fsu.edu

Collaborators:
John Burkardt, Qiang Du, Vance Faber, Hyung-Chun Lee, Lili Ju,
Vincente Romero, Janet Peterson, Xiaoqiang Wang

http://public.lanl.gov/kmh/uncertainty/meetings

CENTROIDAL VORONOI TESSELLATIONNS

- Centroidal Voronoi tessellations (CVT's) are:
  - a way to select the location of points
  - a way to cluster data

- It turns out that these two tasks are of interest in lots of applications
  - it also turns out that in many of these applications, CVT's do a pretty good job

- So, let's start by defining CVT's

TESSELLATIONS

- Given a set S, divide it into K subsets $S_1, S_2, \ldots, S_K$ such that
  - no member of a set S_i is a member of another set S_j
  - every member of S belongs to one of the sets S_i

- We call the set of subsets $\{S_1, S_2, \ldots, S_K\}$ a *tessellation of S*

VORONOI TESSELLATIONS

- Given a set S
- Given K elements z_i, $i = 1, 2, \ldots, K$
- Given a distance function $d(z, w)$ for elements z, w
- Then, the Voronoi subset V_j is the set of all elements belonging to S that are closer to z_j than to any of the other elements z_i, that is,

$$V_j = \left\{ w \in S \mid d(w, z_j) < d(w, z_i),\ i = 1, \ldots, K,\ i \neq j \right\}$$

- We call the set of Voronoi subsets $\{V_1, V_2, \ldots, V_K\}$ a Voronoi tessellation of S or a Voronoi diagram of S
- We call the set of points z_i, $i = 1, 2, \ldots, K$ the generators of the Voronoi tessellation
- The Voronoi subsets V_j are also called Voronoi cells and Voronoi regions

- Voronoi sets are known by many other names, depending on the application

  Voronoi sets =

  Dirichlet regions =

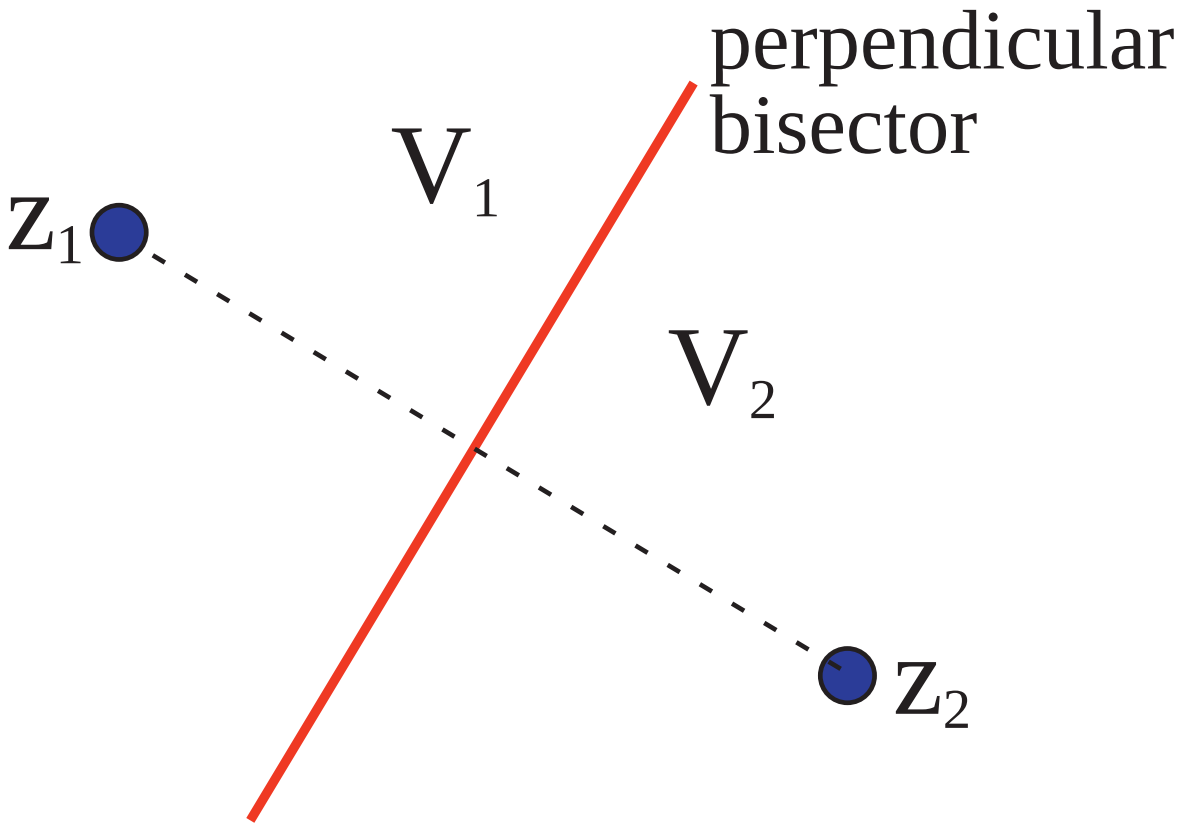
  Meijering cells =

  S-mosaics =

  Thiessen polygons =

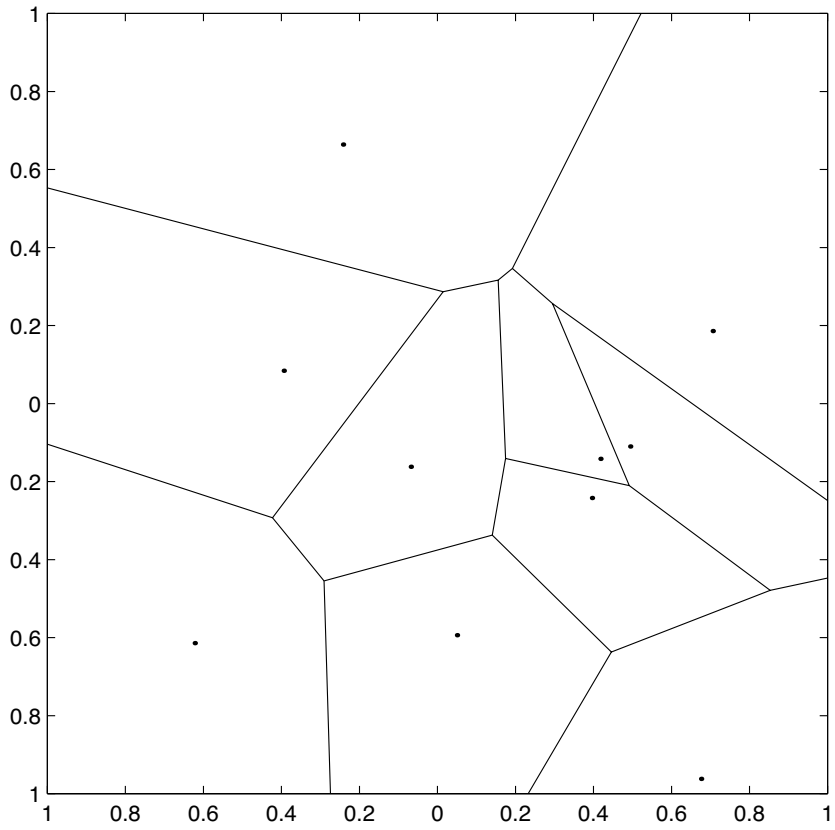
  area of influence polygons =

  etc., etc., etc.



$S =$ the plane; $d(z, w) =$ Euclidean distance; $K = 2$

The Voronoi regions for two points z_1 and z_2 in the plane are the two regions on either side of the perpendicular bisector of the line segment joining z_1 and z_2



$S =$ a square; $\quad d(z, w) =$ Euclidean distance; $\quad K = 10$

Voronoi tessellation for 10 randomly selected points in a square

CENTER OF MASS

- Centers of mass (or mass centroids) can be defined for general sets

- For example, given a region V in $\mathbb{R}^n$ and a density function $\rho(w)$ defined for $w \in V$, the center of mass z^* of V is given by

$$z^* = \frac{\displaystyle\int_V w\rho(w)\,dw}{\displaystyle\int_V \rho(w)\,dw}$$

- Or, given a discrete set of points $W = \{w_j\}_{j=1}^M$ in $\mathbb{R}^n$ and a density function $\rho(w_j)$, $j = 1, \ldots, M$, the center of mass z^* of W is given by

$$z^* = \frac{\displaystyle\sum_{j=1}^{M} w_j\rho(w_j)}{\displaystyle\sum_{j=1}^{M} \rho(w_j)}$$

CENTROIDAL VORONOI TESSELLATIONS

- We have now defined two different notions:

Voronoi tessellations

and

centers of mass

- Let's bring them together

- Given K points z_i, $i = 1, \ldots, K$,
  - we can define the associated Voronoi sets

$$V_i\,, \quad i = 1, \ldots, K$$

- Given the Voronoi sets V_i, $i = 1, \ldots, K$,
  - we can define the associated mass centroids

$$z_i^*\,, \quad i = 1, \ldots, K$$

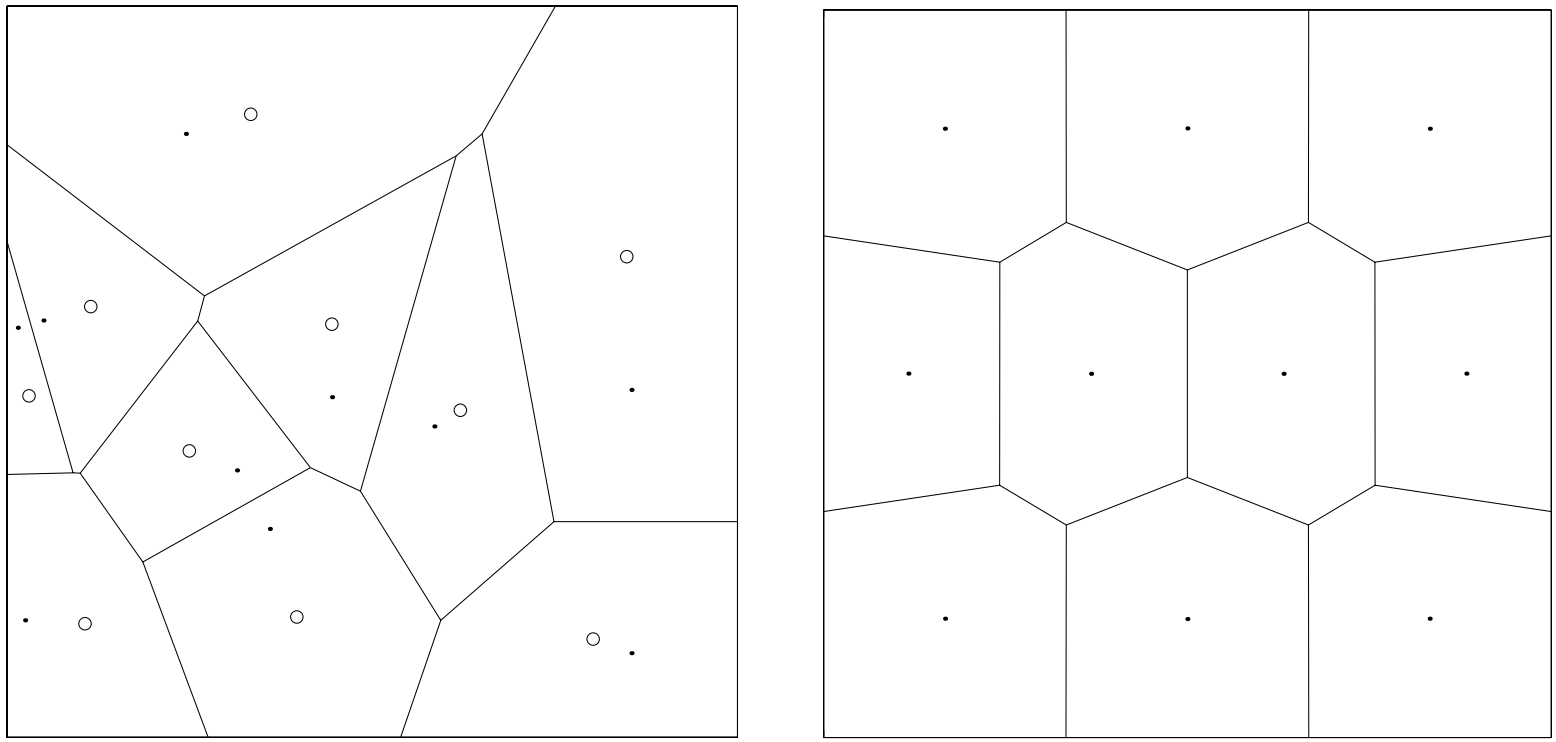
- In general, the centers of mass of the Voronoi sets do not coincide with the generators of the Voronoi sets

$$z_i \neq z_i^*\,, \quad i = 1, \ldots, K$$

- We are interested in the ery special cases for which the generators and centers of mass do actually coincide

$$z_i = z_i^* \,, \quad i = 1, \ldots, K$$

$\Longrightarrow$ **CENTROIDAL VORONOI TESSELLATIONS**



Left: Voronoi regions and their centers of mass for 10 randomly selected points in a square; the generators and the centers of mass do not coincide

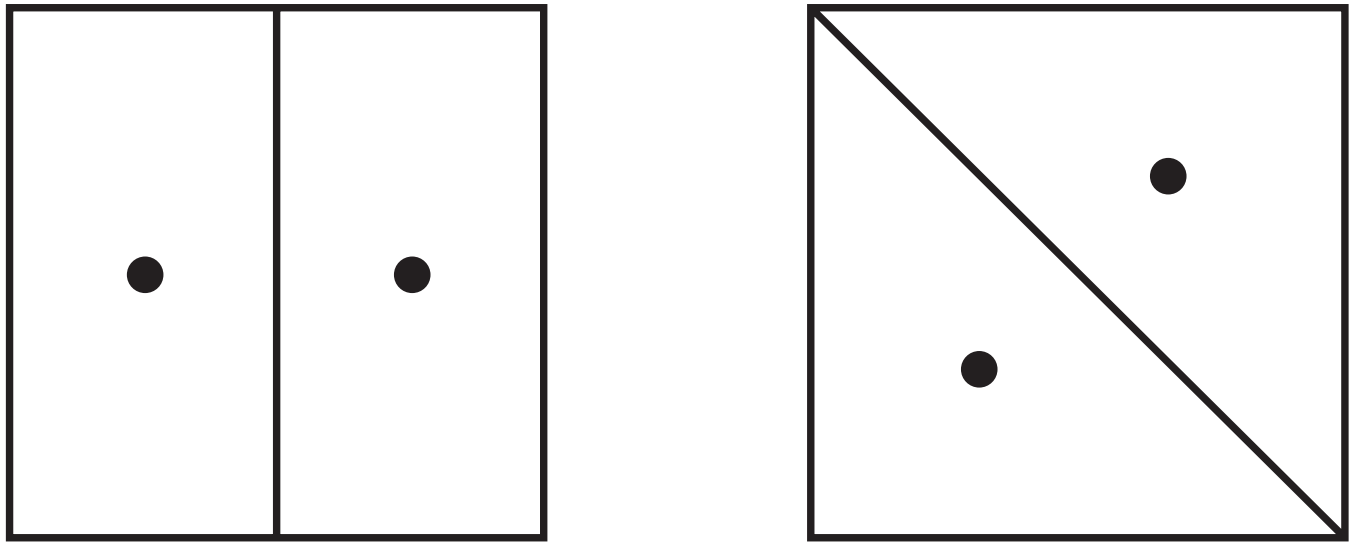
Right: Voronoi regions and their centers of mass for a 10-point centroidal Voronoi tessellation in a square; the generators and the centers of mass coincide

- Centroidal Voronoi tessellations (CVTs) do not happen by accident
  - in fact, the probability is zero that a set of points $\{z_i\}_{i=1}^{K}$ has the CVT property, that is, that they are at the same time the generators of a Voronoi tessellation and are the centers of mass of the Vornoi regions
  - therefore, CVTs must be constructed by some method

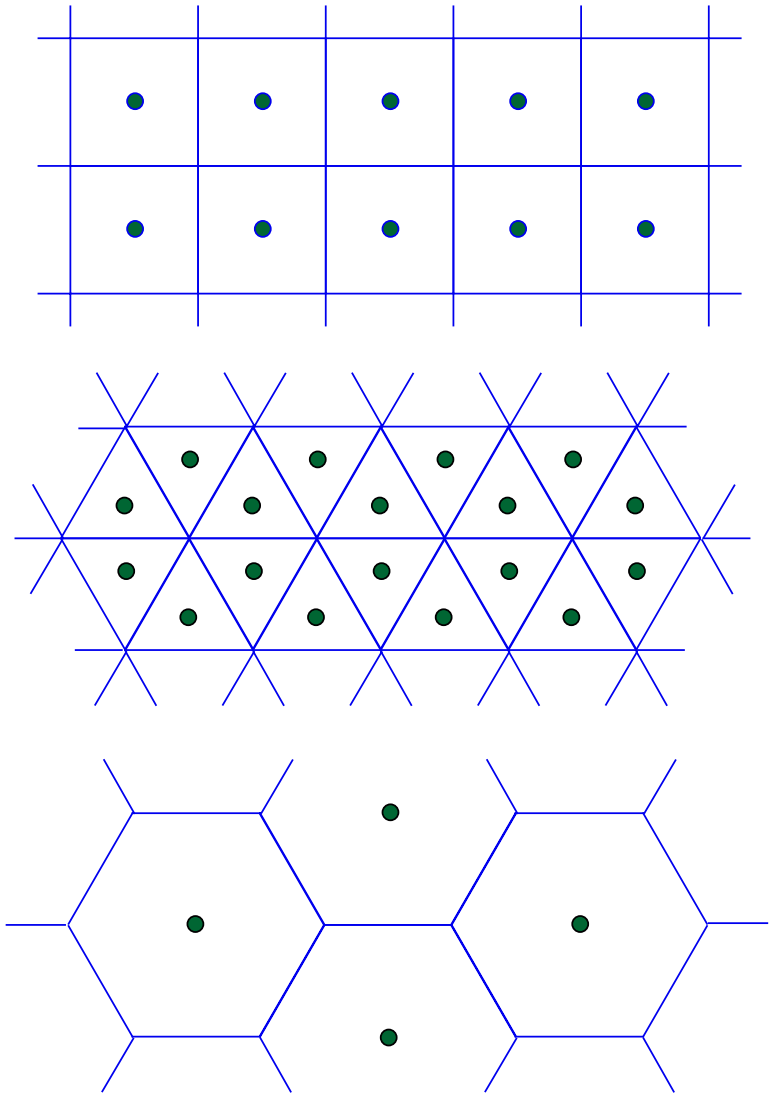
THE CONSTRUCTION PROBLEM FOR CENTROIDAL VORONOI TESSELLATIONS

- Given
  - a region Ω, an integer $K > 1$, and a density function $\rho(w)$
- we are interested in finding
  - K points $\{z_i\}_{i=1}^K$ and K regions $\{V_i\}_{i=1}^K$
- such that
  - $z_i \in \overline{\Omega}$, $i = 1, \ldots, K$, $\{V_i\}_{i=1}^K$ tessellates Ω
- and simultaneously
  - the regions $\{V_i\}_{i=1}^K$ are Voronoi regions for the generators $\{z_i\}_{i=1}^K$
  - the points $\{z_i\}_{i=1}^K$ are the centers of mass of the regions $\{V_i\}_{i=1}^K$

- Note that, in general, one does not have uniqueness



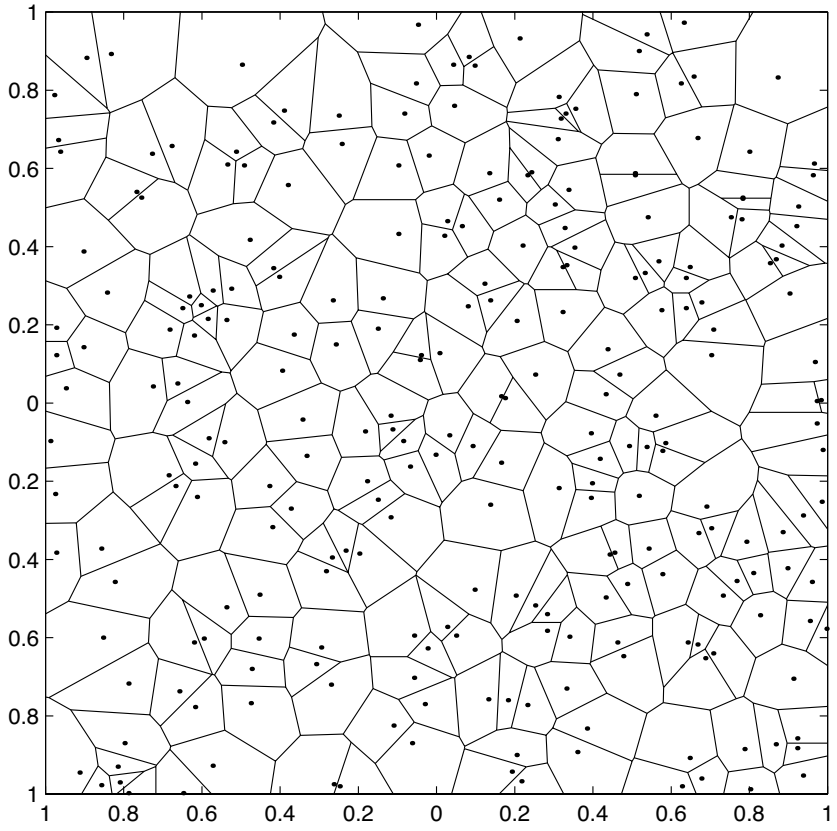
Two two-point centroidal Voronoi tessellations of a square



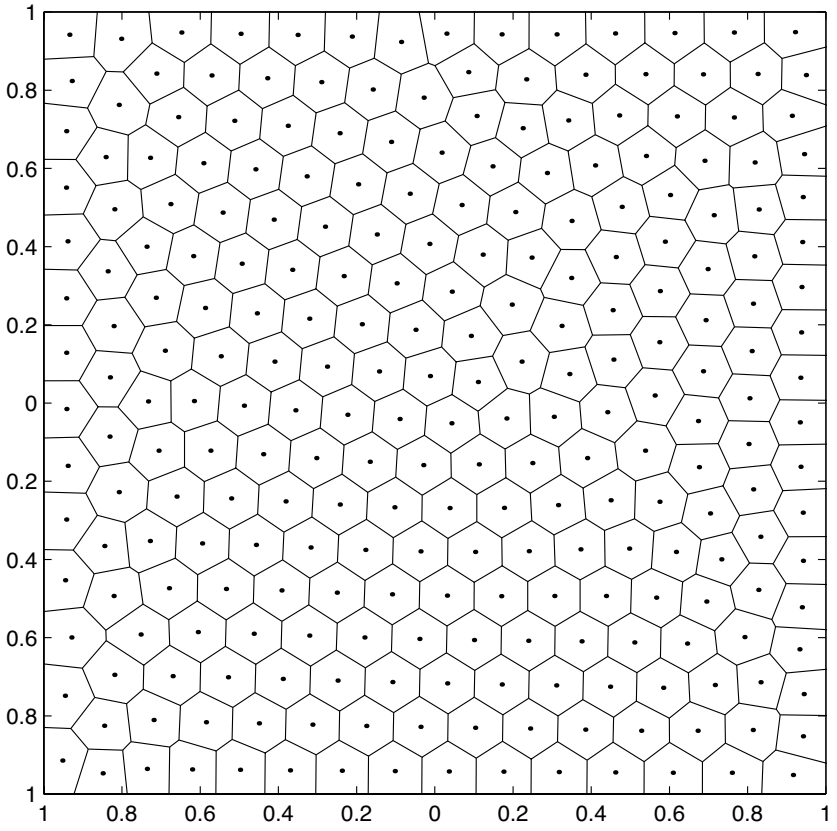
Three regular tessellations of the plane

CVTS IN THE SQUARE

uniform density 256 generators

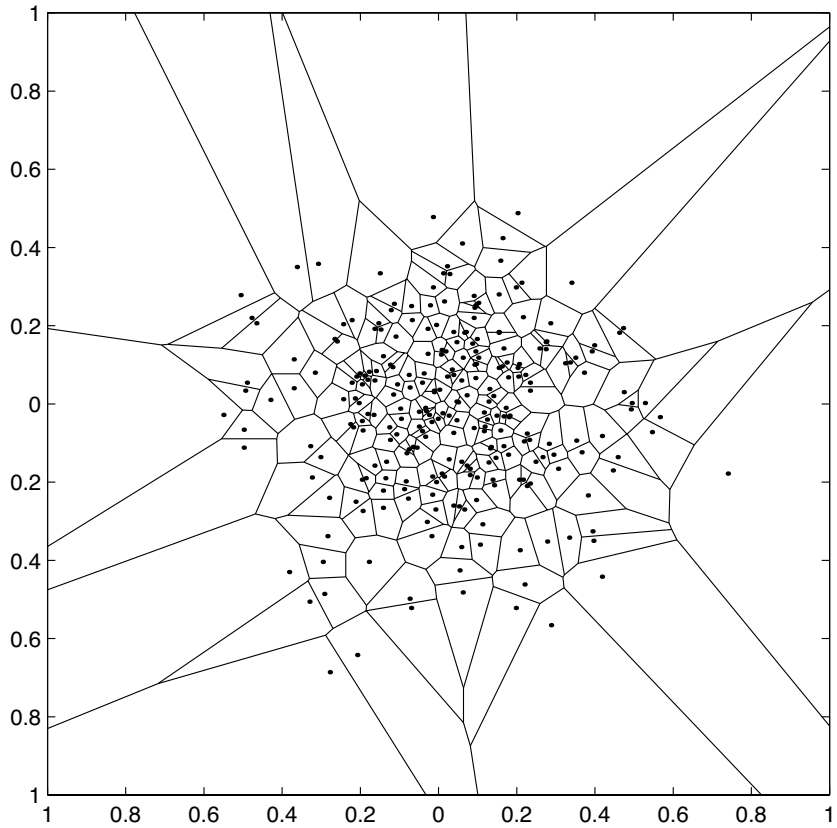


Random sampling

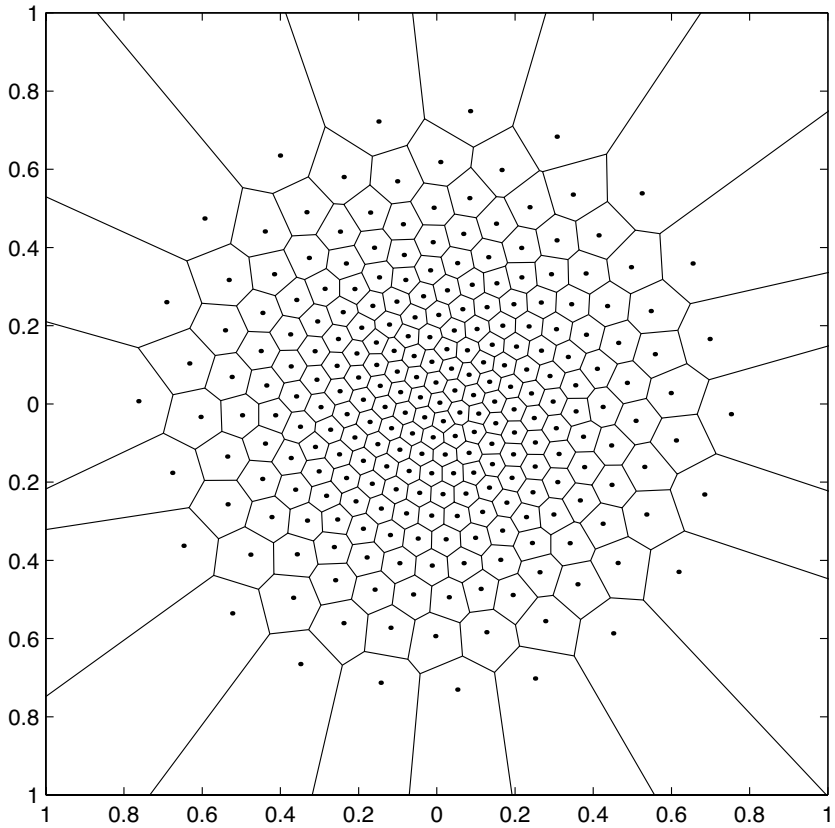


Centroidal Voronoi

density with peak in middle 256 generators

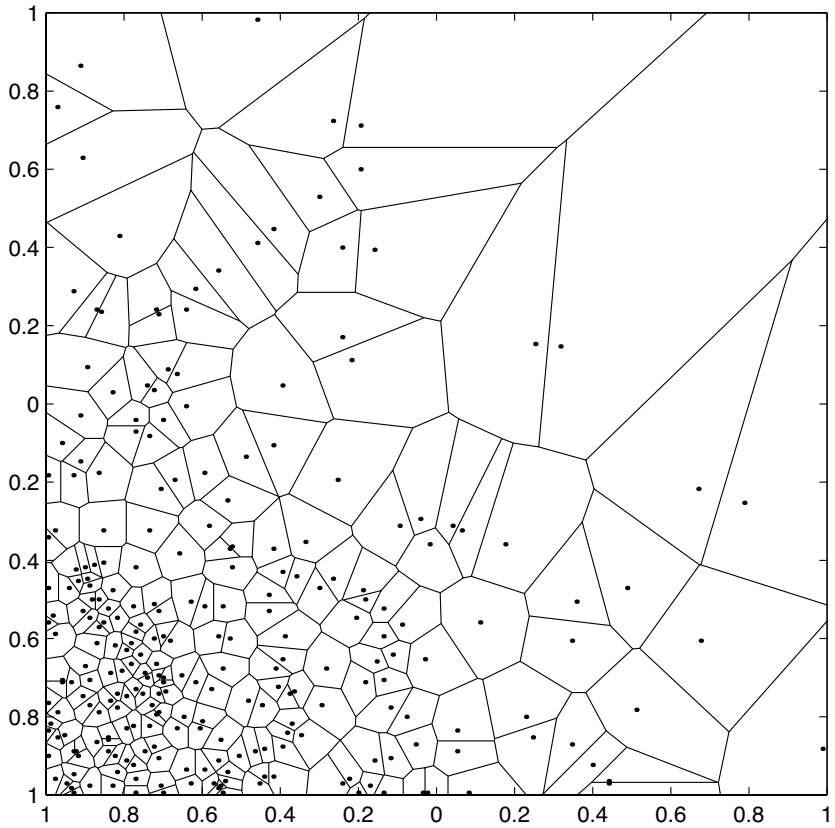


Random sampling

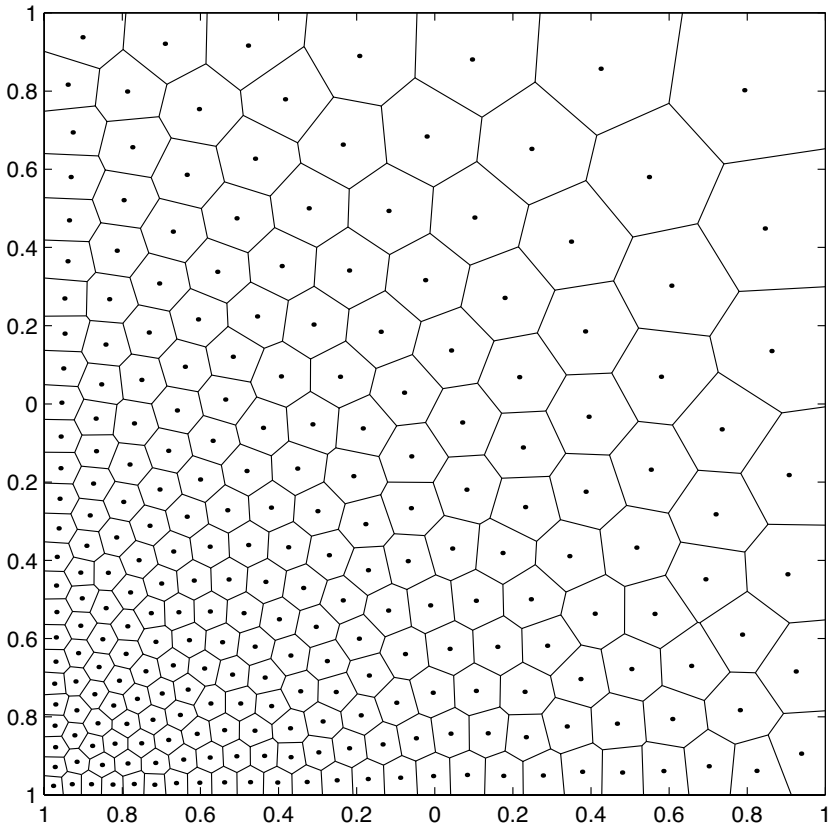


Centroidal Voronoi

density with peak at a corner 256 generators



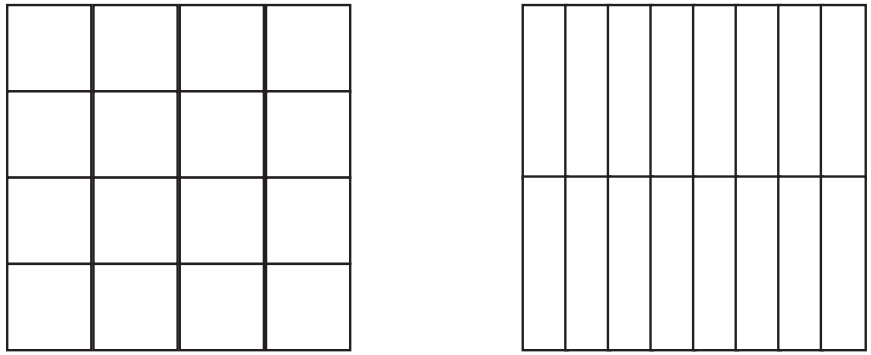
Random sampling

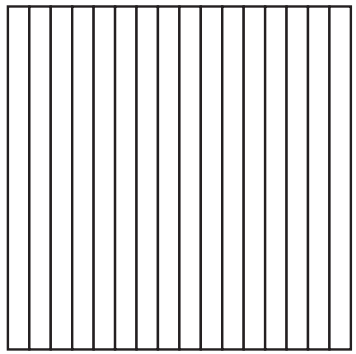


Centroidal Voronoi

- Clearly, there are many other CVTs of a square

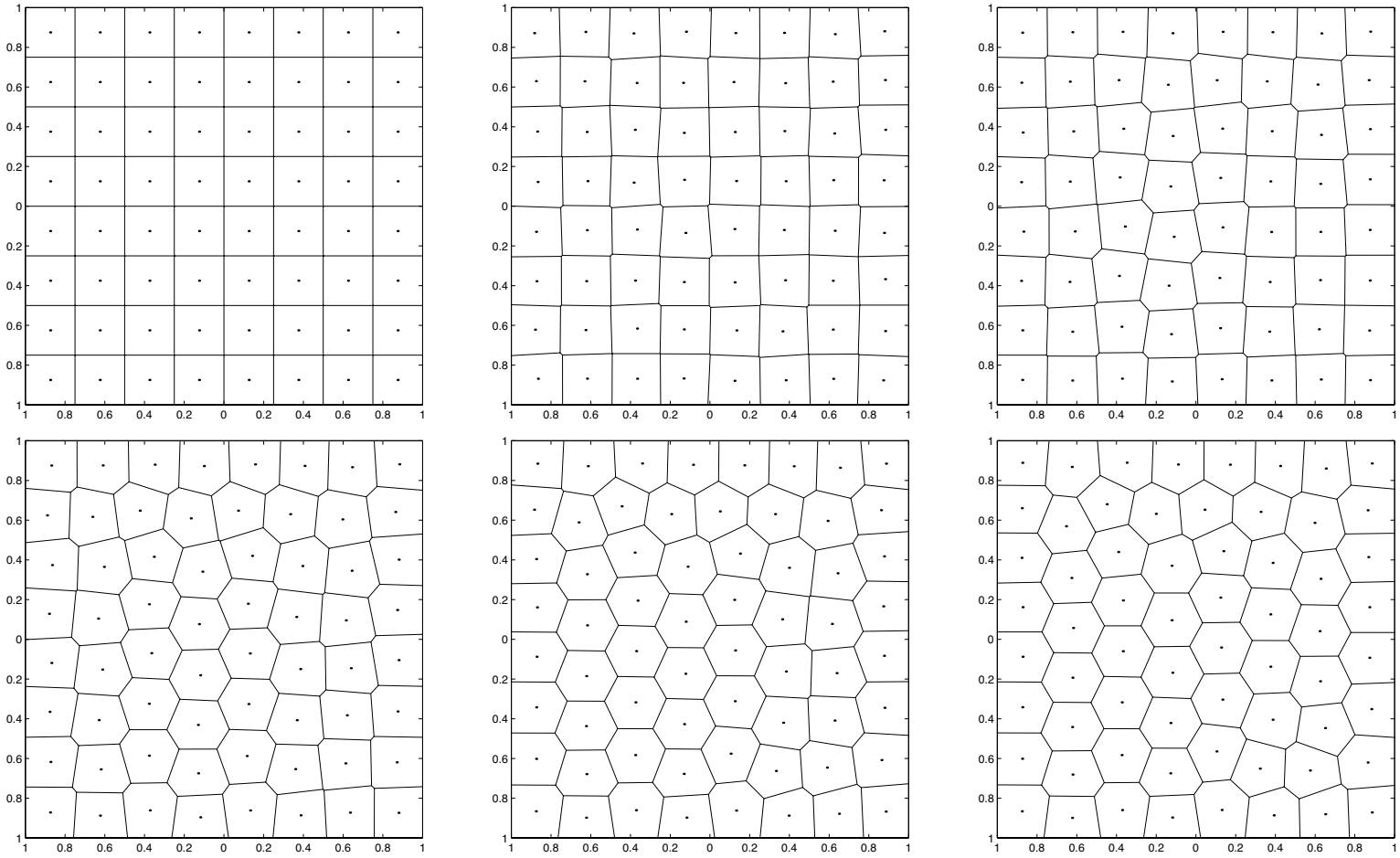
- For example, here are three CVTs of the square with 16 cells





- Which CVT will a computer give you?

- Starting from any CVT, you will end up with a CVT that is as hexagonal as possible



- Why hexagons? Because they possess a certain optimality property

CVT'S FOR DISCRETE SETS

- Sets of discrete data, i.e., vectors in $\mathbb{R}^n$, can also be subdivided into a CVT
  - the previous discussions (and more) holds for CVT's of discrete data sets
  - it may desirable to use more general definitions for the center of mass of discrete objects in defining CVT's for discrete data sets

- For discrete data sets, we can view CVT's as a clustering algorithm,
  - i.e., a method for subdividing a set into subsets, each of which contains elements with a closely related attribute

- In its simplest form, CVT clustering reduces to the very well-known k-means clustering algorithm
  - thus, CVT can be viewed as a generalization of k-means clustering

- There are. of course, many applications where clustering is important; we will discuss some of these

SOME PROPERTIES OF CVT'S

Centroidal Voronoi tessellations as minimizers

Given

- $\Omega \subset \mathbb{R}^N$
- a positive integer K
- a density function $\rho(\cdot)$ defined on $\overline{\Omega}$.

Let

- $\{\mathbf{z}_i\}_{i=1}^K$ denote any set of K points belonging to $\overline{\Omega}$
- $\{V_i\}_{i=1}^K$ denote any tessellation of Ω into K regions.

Let

$$\mathcal{F}\big((\mathbf{z}_i, V_i), i = 1, \ldots, K\big) = \sum_{i=1}^{K} \int_{\mathbf{y} \in V_i} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2 \, d\mathbf{y} \, .$$

Then, a necessary condition for $\mathcal{F}$ to be minimized is that $\{V_i\}_{i=1}^K$ and $\{\mathbf{z}_i\}_{i=1}^K$ form a centroidal Voronoi tessellation of Ω.

- If $\Omega \in \mathbb{R}^N$ is bounded, then $\mathcal{F}$ has a global minimizer.

- Assume that $\rho(\cdot)$ is positive except on a set of measure zero in Ω
  - then $\mathbf{z}_i \neq \mathbf{z}_j$ for $i \neq j$

- For general metrics, existence is provided by the compactness of the Voronoi regions; uniqueness can also be attained under some assumptions, e.g., convexity, on the Voronoi regions and the metric

- There are many additional results available for the discrete case
  - many of these are in the nature of limiting results as the sample size increases

Gersho's conjecture (proven in 2D under assumptions)

- For any density function, as the number of points increases, the distribution of CVT points becomes locally uniform
  - this means that if one looks at small enough patch and a large enough number of points, the CVT point distribution will look uniform, regardless of how nonuniform it is globally

- In 2D, CVT Voronoi regions are always locally congruent regular hexagons
  - that is, CVT points are always locally vertices of congruent equilateral triangles
  - the dual Delaunay CVT grid is always locally congruent equilateral triangles

- In higher dimensions, the basic cell of a CVT grid is not known

- Gersho's conjecture is a key observation that helps explain the effectiveness of CVT's

ALGORITHMS FOR CONSTRUCTING CVT'S

Lloyd's method

0. Start with some initial set of K points $\{z_i\}_{i=1}^K$

1. Construct the Voronoi tessellation $\{V_i\}_{i=1}^K$ of Ω associated with the points $\{z_i\}_{i=1}^K$

2. Construct the centers of mass of the Voronoi regions $\{V_i\}_{i=1}^K$ found in Step 1; these centroids are the new set of points $\{z_i\}_{i=1}^K$

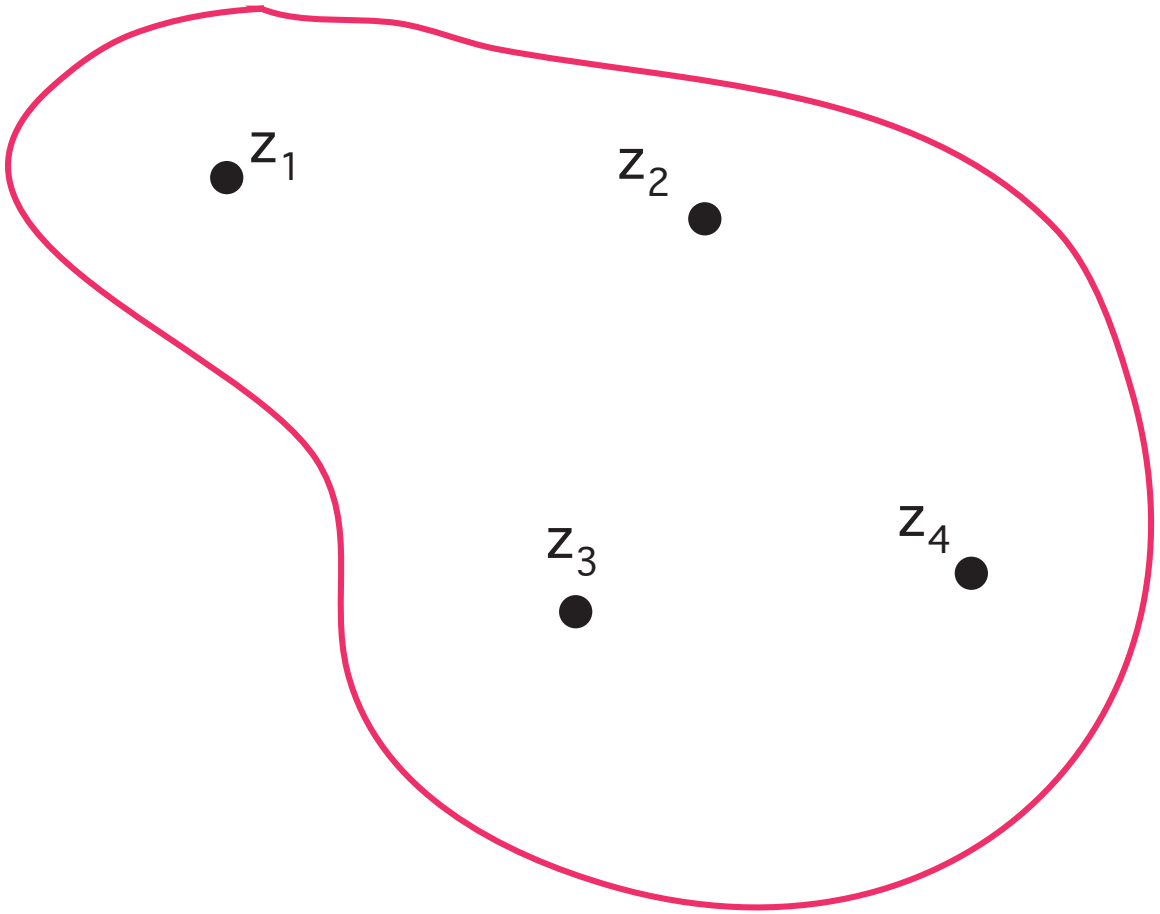
3. Go back to Step 1, or, if you are happy with what you have, quit

---

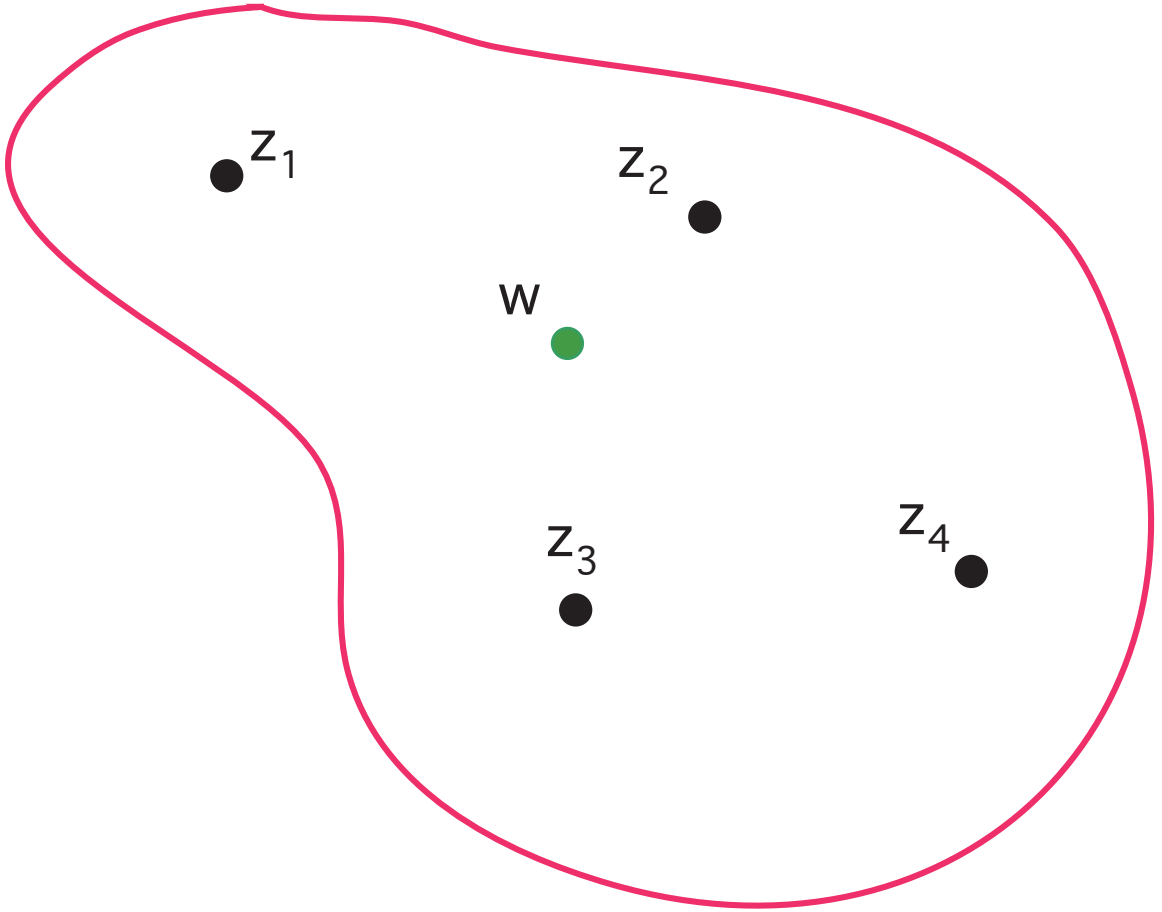
- Steps 1 and 2 can both be costly to effect

McQueen's method (random sampling and averaging)

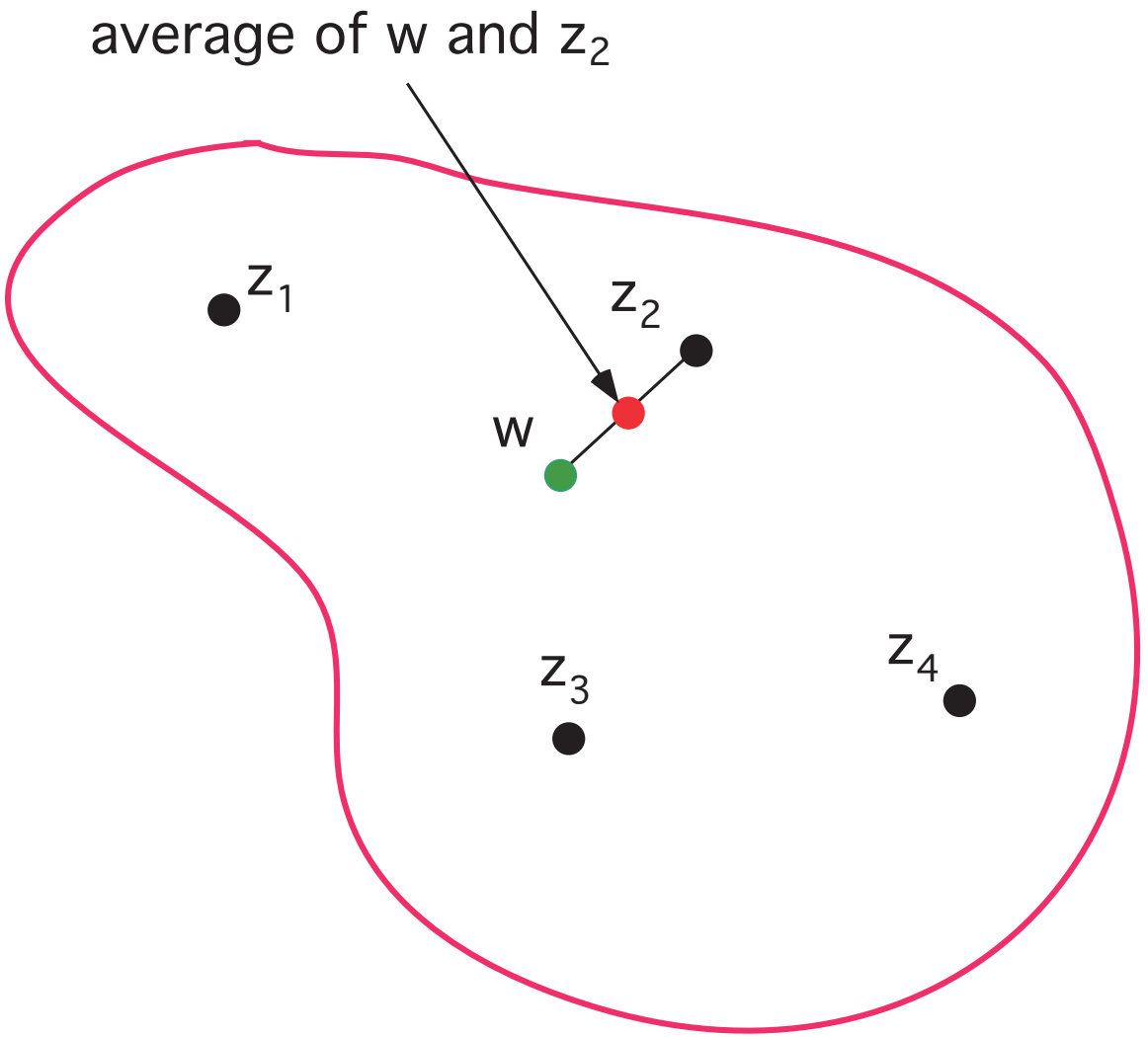
- Start with some initial set of K points $\{z_i\}_{i=1}^{K}$ ($K = 4$ in the sketch)



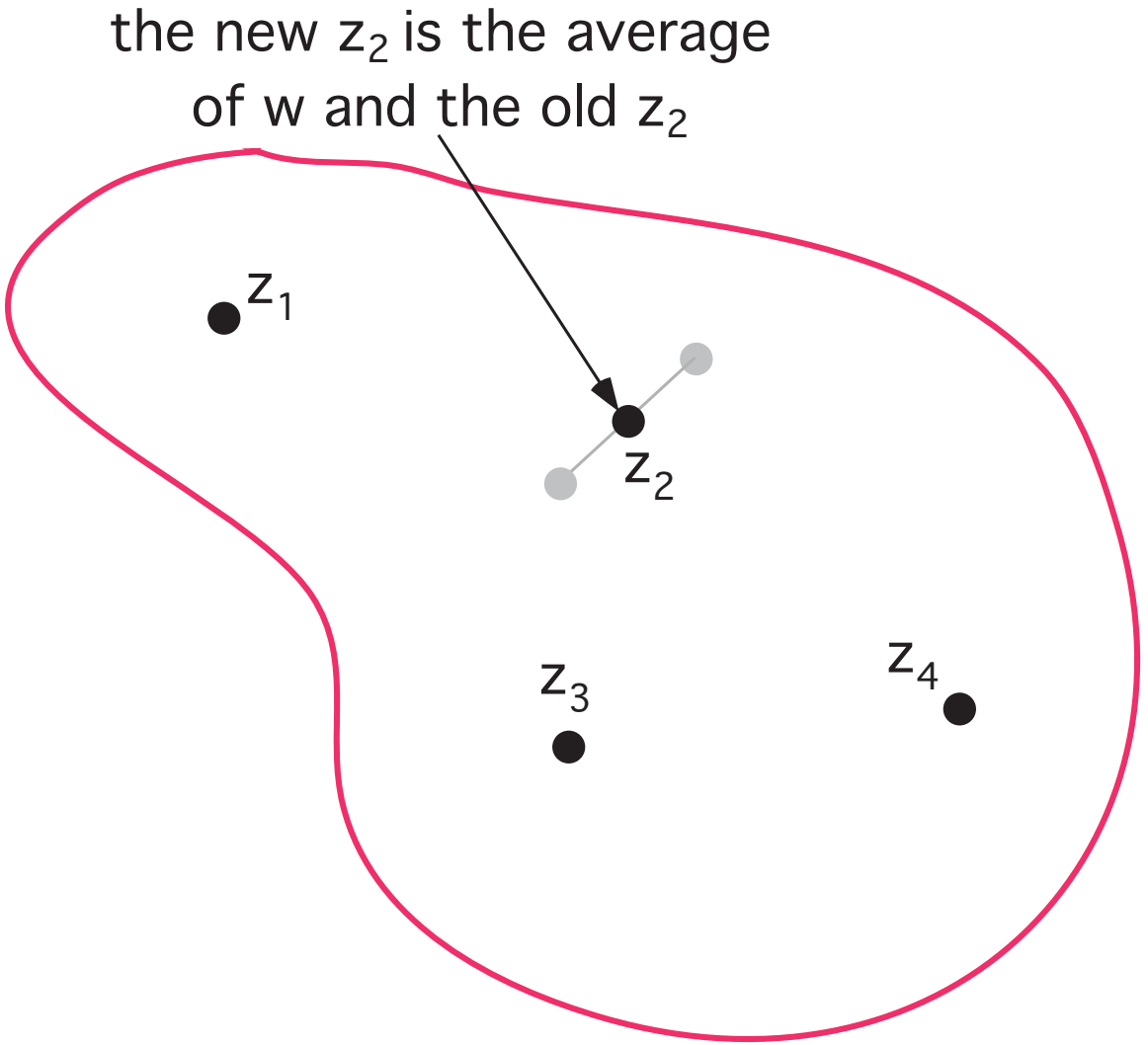
- Sample another point w
- Determine which of the z_i's is closest to to w (it is z_2 in the sketch)



- Find the average of w and the z_i closest to it



- Replace the z_i used in the averaging by the average point

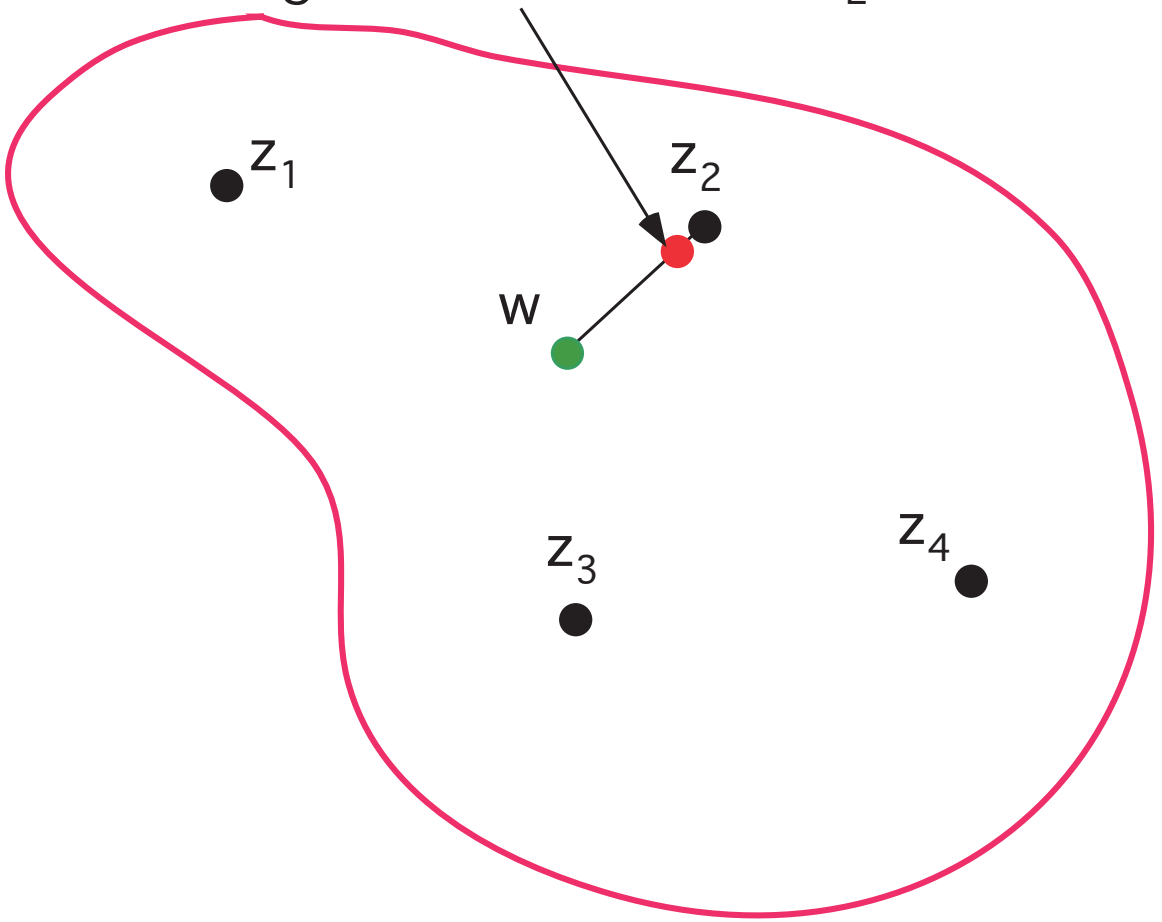


- Continue the process, that is,
  - sample points w
  - find the closest z_i
  - average w and that z_i
  - replace that z_i by the average

  except
  - we keep track of how many times a points has been previously updated
  - when we do the averaging, we weight the old point according the number of times it has been previously updated

- For example, suppose z_2 had already been updated 12 times (counting the intial positions as the first update); then
  - instead of the new $z_2 \leftarrow \dfrac{w + z_2}{2}$
  - we have the new $z_2 \leftarrow \dfrac{w + 12z_2}{13}$

since z_2 had previously updated,
the new z_2 is the weighted
average of w and the old z_2

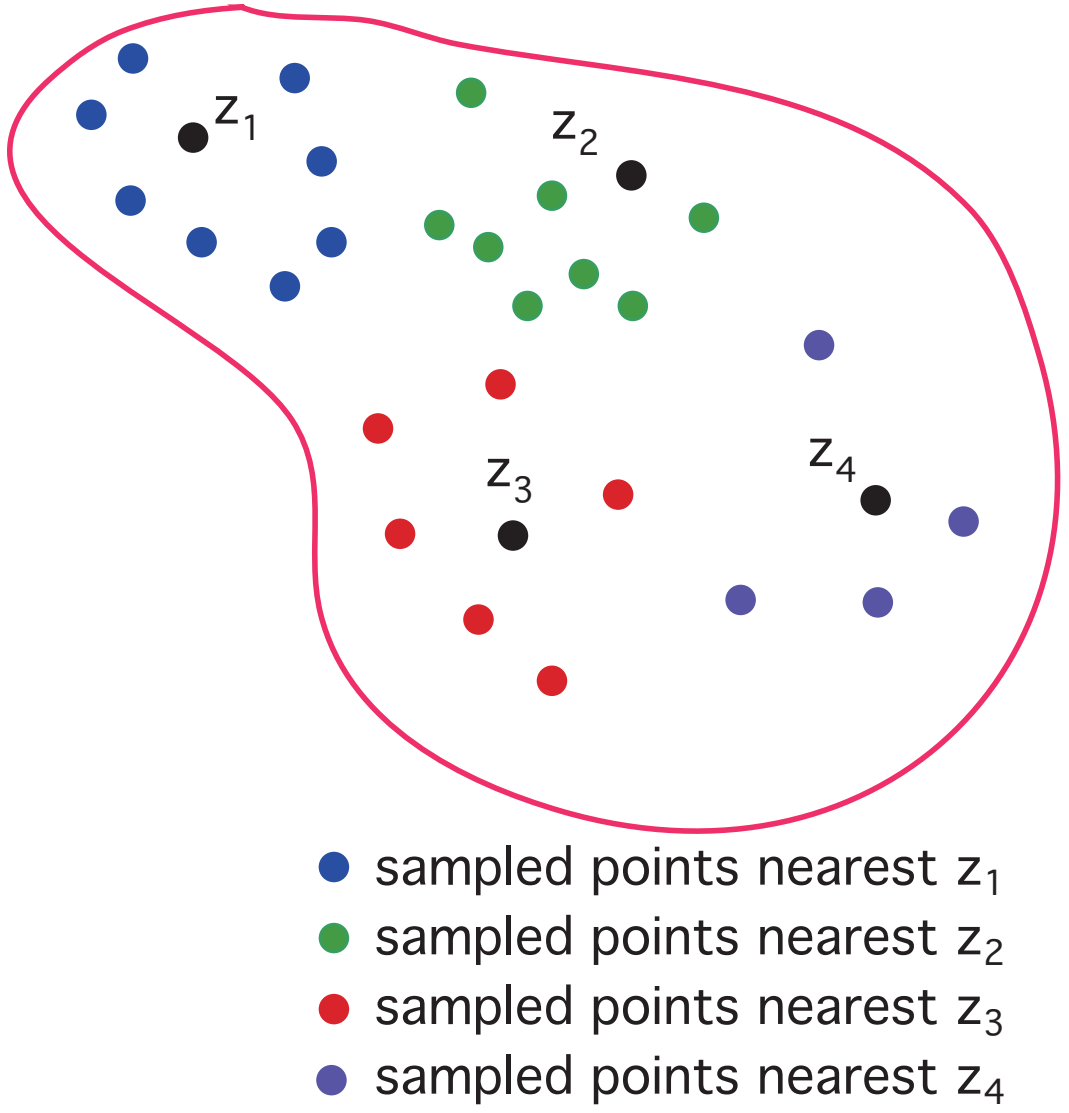


- McQueen's method doesn't require the construction of Voronoi sets or centers of mass
  - despite this, the points produced by McQueen's method converge to the generators of a centroidal Voronoi tessellation

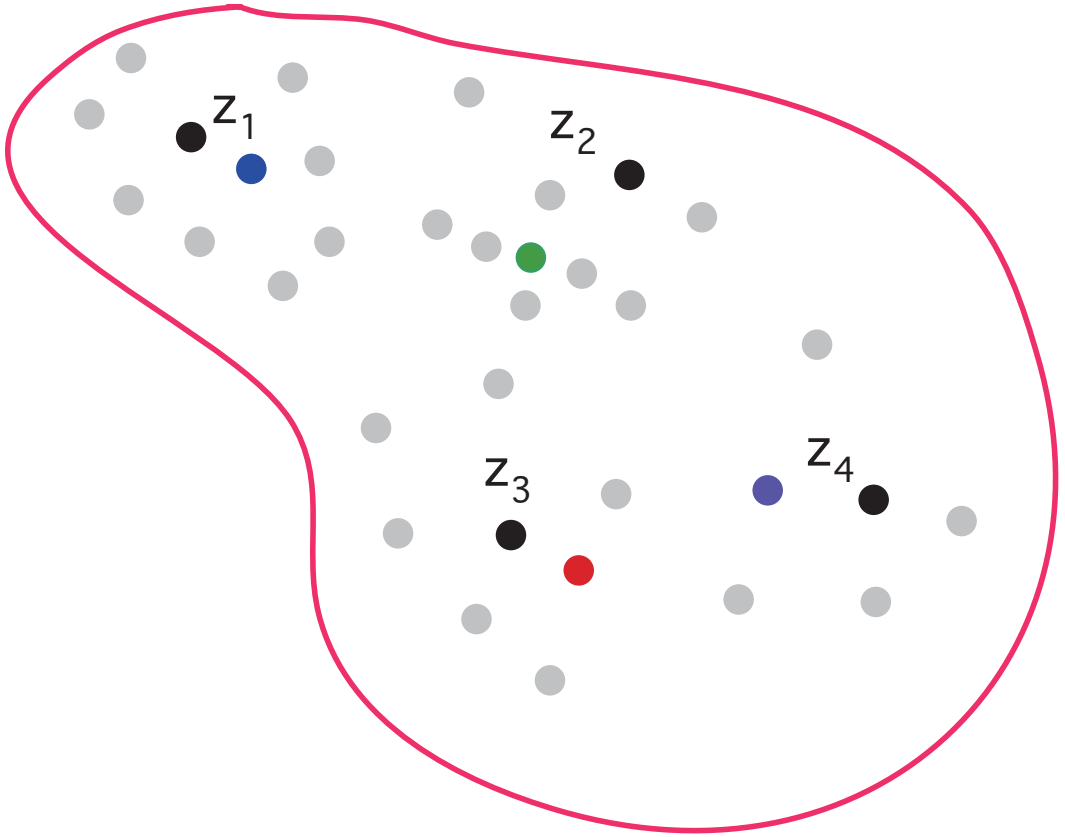
- The convergence of McQueen's method is very slow, that is, it takes many steps (millions) to obtain a CVT set of points from an initial set of points
  - the problem with McQueen's method is that it samples only one point before it averages

- New method
  - sample a lot of points before averaging
  - very good for parallel computers

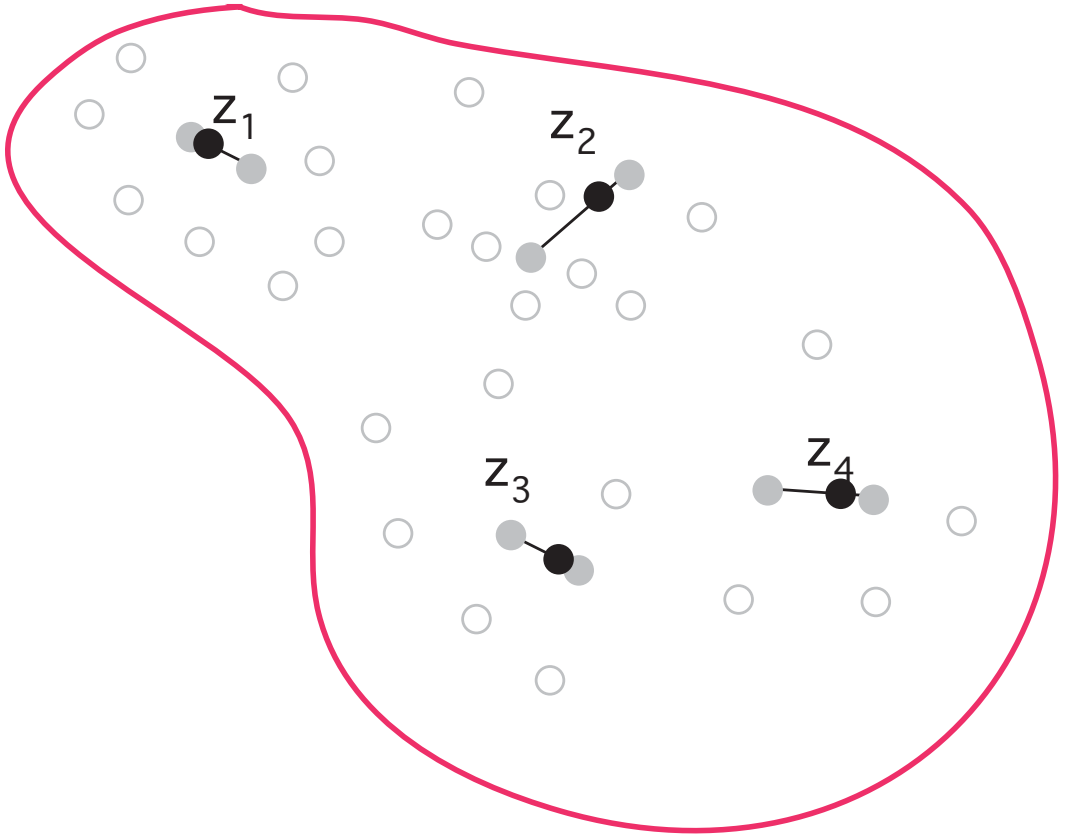
- Sample lots of points (thousands) and group (cluster) them according to which is the nearest z_i



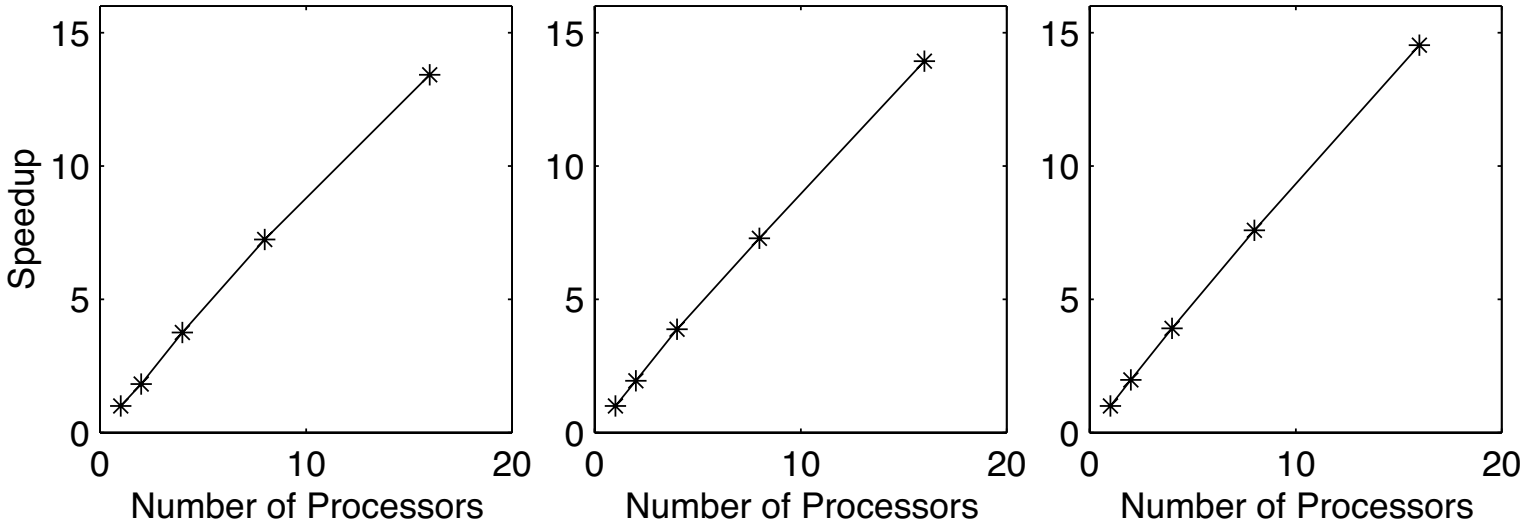
- Find the average of each of the clusters



- The new z_i's are a weighted average of the old z_i's and the corresponding cluster averages



- Parallel versions of the new method exhibit near-perfect scalability



Speedup of a parallel implementation of the new algorithm
for three different density functions

- We have, in fact, developed a two parameter family of effective, proabilistic methods for generating CVT's in general regions and with general point density distributions

GENERALIZATIONS OF CVT

- Constrained CVT for placing some of of all of the points on a surface
- Constrained CVT for fixing the position of some points
- Other metrics, e.g., for anisotropic point distributions one can use anisotropic metrics

APPLICATIONS OF CVT'S

- optimal quadrature rules
- covolume and finite difference methods for PDE's
- optimal representation, quantization, and clustering
- optimal placement of sensors and actuators
- optimal distribution of resources
- cell division
- finite volume methods for PDE's
- territorial behavior of animals
- data compression
- image segmentation
- meshfree methods
- grid generation
- point distributions and grid generation on surfaces
- hypercube point sampling
- reduced-order modeling

OPTIMAL DISTRIBUTION OF RESOURCES

What is the optimal placement of mailboxes in a given region?

- A user will use the mailbox nearest to their home
- The cost (to the user) of using a mailbox is proportional to the distance from the user's home to the mailbox
- The total cost to users as a whole is measured by the distance to the nearest mailbox averaged over all users in the region
- The optimal placement of mailboxes is defined to be the one that minimizes the total cost to the users

The optimal placement of the mail boxes is at the centroids of a centroidal Voronoi tessellation, using the population density as the density function in the center of mass definition

CELL DIVISION

- There are many examples of cells that are polygonal – often they can be identified with a Voronoi, indeed, a centroidal Voronoi tessellation
  - this is especially evident in monolayered or columnar cells, e.g., as in the early development of a starfish (*Asteria pectinifera*)

- **Cell division**
  - Start with a configuration of cells that, by observation, form a Voronoi tessellation (this is very commonly the case)
  - After the cells divide, what is the shape of the new cell arrangement?
  - It is observed that the new cell arrangement is closely approximated by a centroidal Voronoi tessellation

FINITE VOLUME METHODS FOR PDE'S

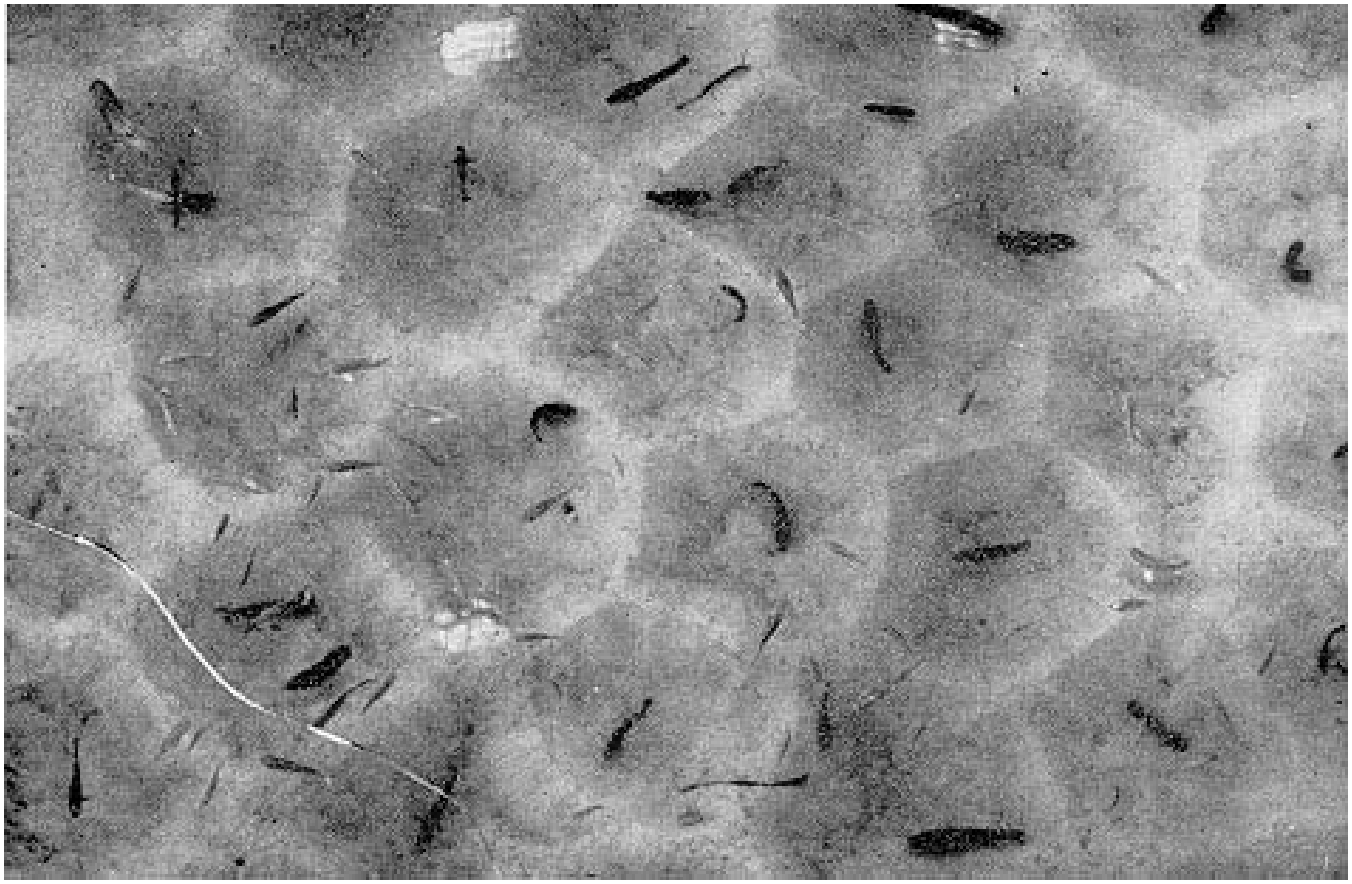
- Finite volume methods, including some based on Delaunay/Voronoi grids, are sometimes observed to be second-order accurate

- Recenlty, Q. Du and L. Ju have proved that a finite volume scheme based on CVT's and the its dual Delaunay grid is second-order accurate
  - their result holds for general, unstructured CVT grids
  - their result also holds for finite volume schemes on the sphere

TERRITORIAL BEHAVIOR OF ANIMALS

Male moutbreeder fish – *Tilapia mossambica*

- Fishes dig nesting pits in sandy bottoms
- They adjust the centers and boundaries of the pits so that the final configuration of territories is a centroidal Voronoi tessellation



A top view photograph, using a polarizing filter, of the territories of the male *Tilapia mossambica*

Photograph from: George Barlow; Hexagonal territories, *Animal Behavior* **22** 1974, pp. 876–878

DATA COMPRESSION (IMAGE PROCESSING)

- Each point in a picture has a specific color
- Each color is a combination of basic (primary, RGB, CMY) colors
- Let the components of a vector w represent a possible combination of the basic colors
- Let $\rho(w)$ denote the number of times the particular combination w occurs in the picture
- Let Ω denote the set of admissible color combinations
- There are zillions of different colors in a given picture
- One would like to approximate the picture using just a few combinations of the basic colors

- Questions:

  1. How to choose the few colors that are to be used to represent the picture?

  2. How to assign the colors in the picture to the few chosen colors?

     –one usually assigns all colors in the Voronoi region (in color space) associated with one of the few chosen colors to that chosen color

- Why data compression?

  – the number of pixels remains the same, but the information associated with each pixel is reduced

- "Obvious" method for choosing the reduced set of colors
  - using random sampling to determine the colors
  - even with 256 approximating colors, doesn't give good approximate pictures

- Better method for choosing the reduced set of colors
  - choose the colors by doing a CVT in color space
  - produces great approximate pictures
  - method used on some (early) HP color ink jet printers

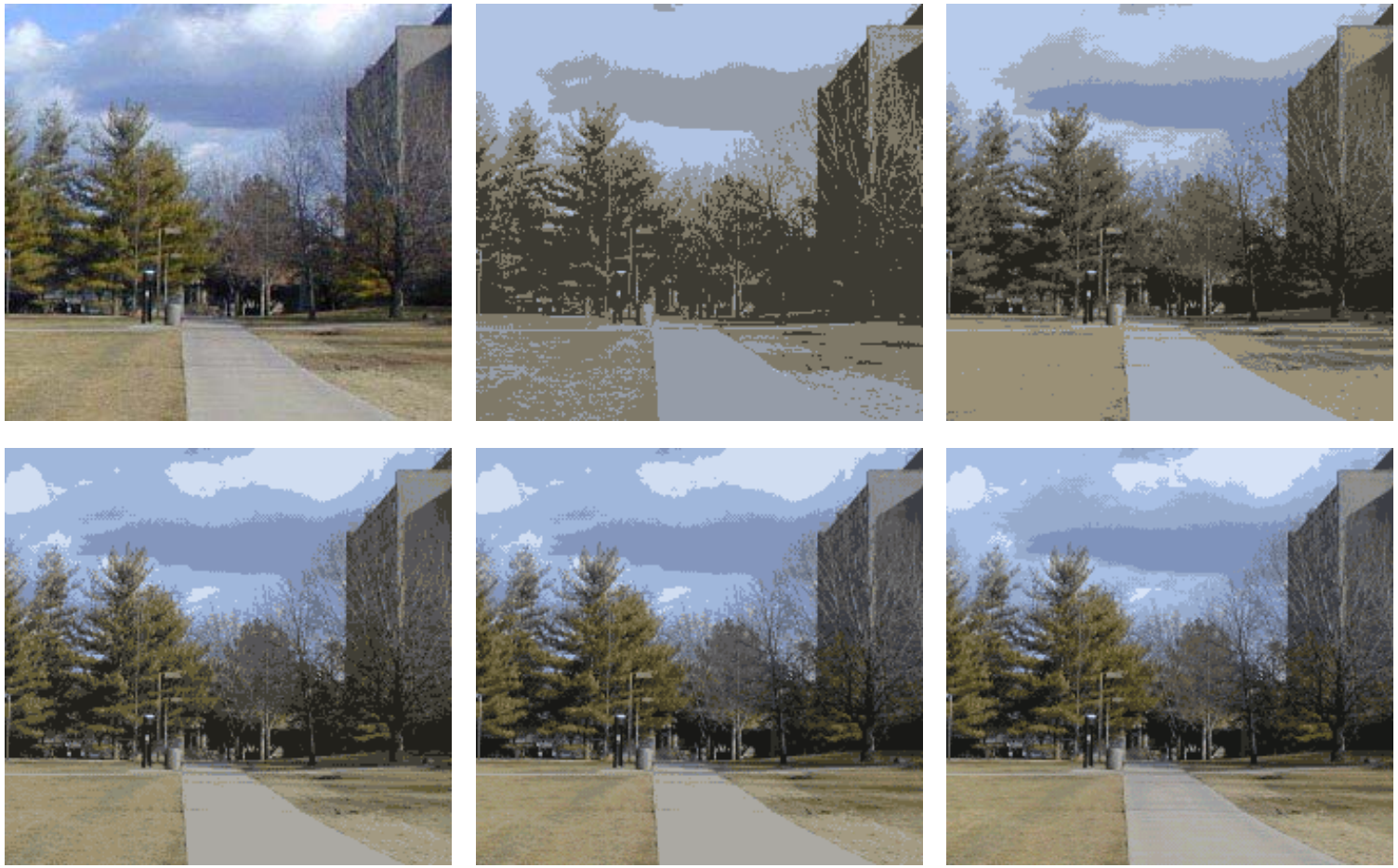
Original 8-bit grayscale image

Centroidal Voronoi 6-bit approximate picture

Centroidal Voronoi 5-bit approximate picture

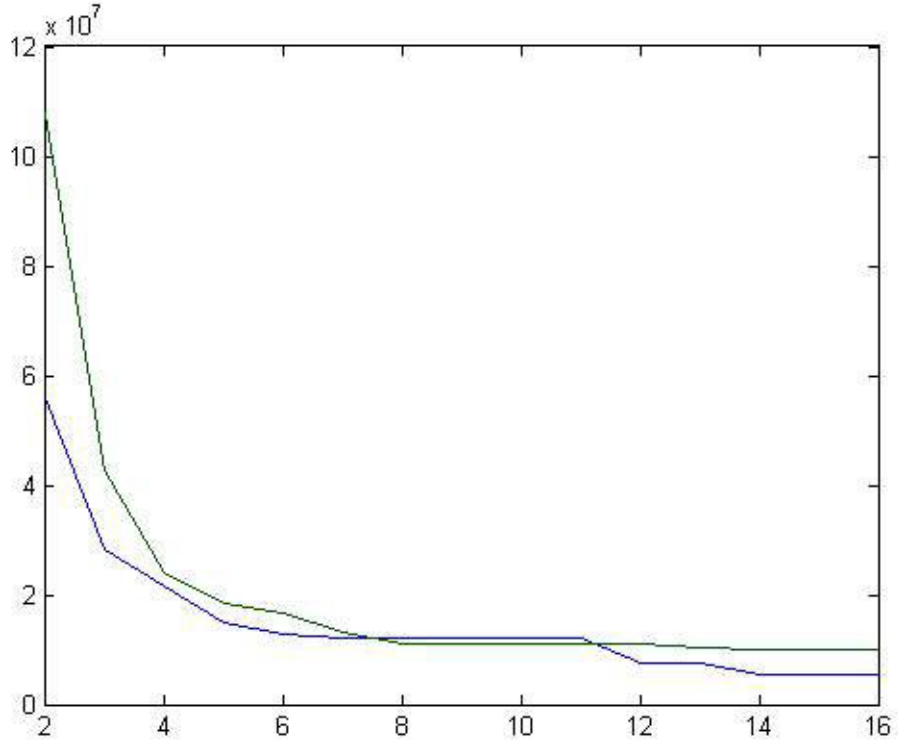
Centroidal Voronoi 4-bit approximate picture

Centroidal Voronoi 2-bit approximate picture



From left to right and top to bottom: original image containing 1434 different colors and CVT-approximate images containing 4, 8, 16, 32, and 64 colors, respectively

- Elbowing effect: the energy vs. number of generators (reduced set of colors) decreases rapidly at first, but then, as one increases the number of generators, reductions in the energy become less pronounced
  - can be used to determine the number of generators that should be used



Energy of CVT-approximate images for two images vs. the number of replacement colors

IMAGE SEGMENTATION AND EDGE DETECTION

- One can regard a rectangular digital image as a function defined over the set of the integral points $(x, y) = (i, j)$, where $i = \{1, 2, \ldots, I\}$ and $j = \{1, 2, \ldots, J\}$ for two positive integers I and J

- We denote that function by u, with the values of u representing some attribute of the picture, e.g., color or brightness.

- Image segmentation is the process of identifying the parts of an image that have a common attribute,

  e.g., that are roughly the same color or have the same brightness,

  and to also identify edges,

  i.e., the boundaries between the different segments of the image

- The segmentation is done on the physical picture

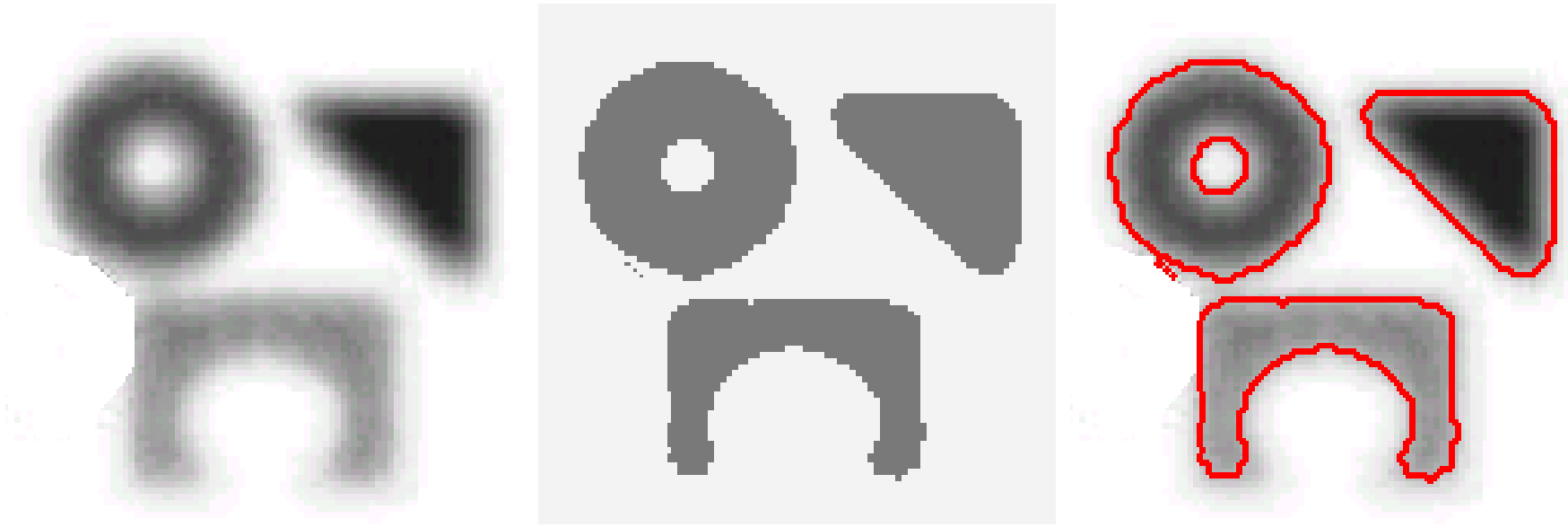
- CVT clustering of an image can be easily used for image segmentation
  - recall that CVT clustering is the partitioning of an image in color space

- Suppose that we have partitioned an image into the CVT clusters $\{V_\ell\}_{\ell=1}^L$
  - in physical space, this corresponds to the segmentation of the image into the L segments $\{\mathcal{V}_\ell\}_{\ell=1}^L$, where

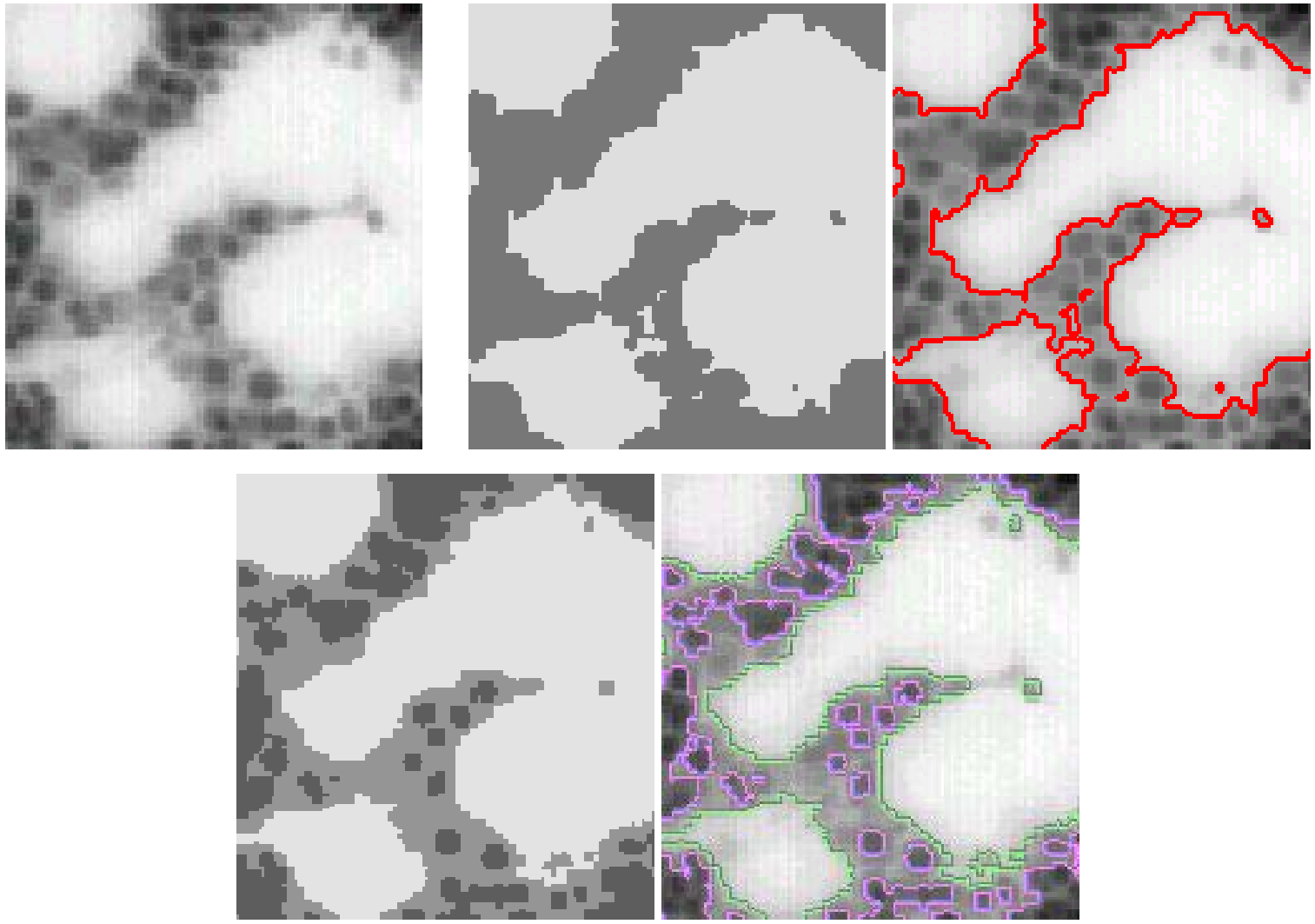
$$\mathcal{V}_\ell = \{(i,j) \ : \ u(i,j) \in V_\ell\} \, .$$

- Edges can be detected by seeing if neighboring points belong to a different cluster
  - the point (i,j) is an edge point of the segment $\mathcal{V}_\ell$ if one of its neighboring points belongs to a different segment $\mathcal{V}_k$, $k \neq \ell$

- The elbowing effect of the CVT "energy" can be used to determine a good number of segments to use



CVT-based segmentation and edge detection into two segments



CVT-based segmentation and edge detection of a bone tissue image (top-left) into two (top-right) and three (bottom) segments

- CVT-based image segmentation is very cheap compared to, e.g., PDE and variationally based methods

- Simple, CVT-based image segmentation, as just illustrated, is well known
  - it is simply k-means clustering applied to image segmentation

- However, the CVT context easily allows for several useful generalizations

Weighted CVT's

- Give some cells more weight than others
  - can help enhance the separation between colors

Original image (left) and weighted CVT-based compressed images with, from left to right, equal weight functions, weight functions proportional to cell volume and square of the cell volume

CVT with averaging for segmenting noisy or irregular images

- Averaging can be used to smooth out noisy or irregular images
  - we effect averaging through discrete convolutions

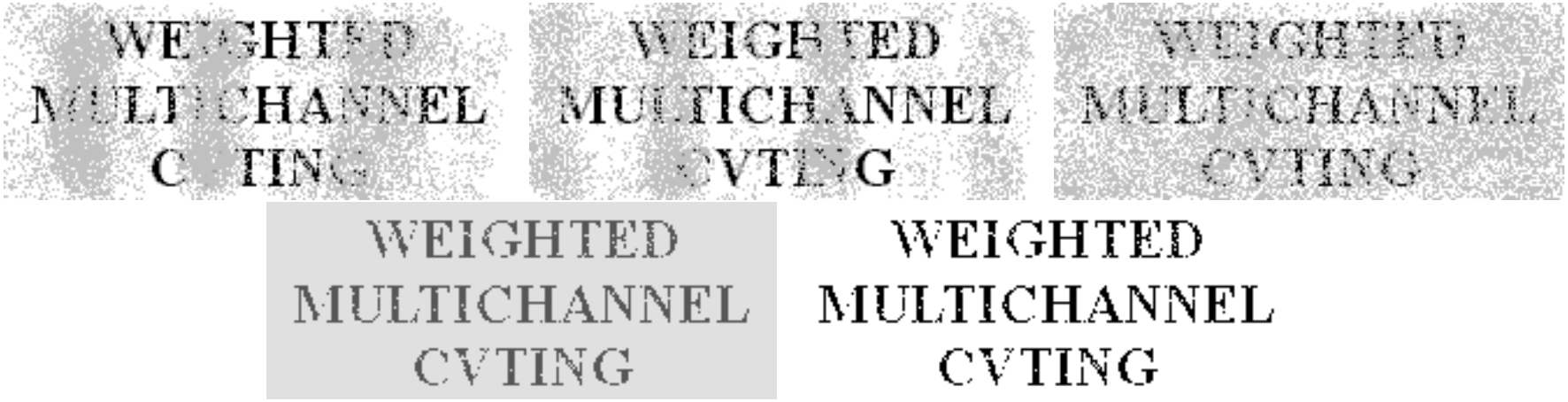
Original image (left); averaged image (middle); image with stronger averaging (right)

Original grayscale bone tissue image (left); CVT-based segmentation into two segments with subsequent averaging (middle); the corresponding edges (right)

Original grayscale image of a house (left); CVT-based segmentation into two segments with subsequent averaging (middle); the corresponding edges (right)

Multichannel image processing

- Suppose one has in hand several versions of a picture, none of which contains all the information necessary to recover the complete image
  - how can the information contained in the different versions be combined so as to recover the whole image?
  - this is a natural task for CVT-based image processing



Three noisy and incomplete versions of the same image (top); multichannel CVT-based approximate image using 2 replacement colors (bottom-left); restored image using renormalization (bottom-right)

- Can combine multi-channel CVT with averaging and weighting

MESHLESS COMPUTING

- Meshless computing refers to numerical methods that do not involve meshes

- They can be used, e.g., for approximating
  - multivariate functions
  - multiple integrals
  - solutions of partial differential equations

- The hope is that since meshes are not required, problems posed on complicated and/or high-dimensional regions can be treated more easily

- Of course, meshless computing is nothing new, e.g., recall
  - Monte Carlo methods for numerical integration and the numerical solution of PDE's
  - particle methods for PDE's
  - kinetic methods for PDE's

- A typical efficient meshless computing method requires
  1. the selection of a set of points
  2. the selection of a support region associated with each point
  3. the selection of a basis function associated with each point having support over the region selected in step 2

- The implementation of a typical meshless method also requires
  4. knowledge about the overlap of the support regions associated with distinct basis functions
  5. the application of a discretization method, e.g., in the PDE setting, one can use any of
     – Galerkin, collocation, mixed, or least squares methods
     – while in the function approximation setting, one can use, among many choices, interpolation or least squares approximation

- One can also analyze meshless methods
  6. derive error estimates

- A great deal of attention has been devoted to steps 3, 5, and 6
  - lots of papers on deriving error estimates for specific choices of basis functions, assuming the point destribution is given and the support radii and the overlap information are known or at least are easily determined

- Some papers determine support radii and overlap information by using graph theoretic ideas

  since graph = mesh, these papers do not address truly meshless methods

- Much less effort has been devoted to the efficient determination, in a truly meshless way, of
  - optimal point distributions
  - support radii
  - overlap information

Point placement

- We use the generators of a centroidal Voronoi tessellation of the computational domain for the points on which the meshless method are based
  - using probabilistic algorithms, the CVT points can be determined in a totally meshless manner

- There are, of course many other methods for selecting point positions
  - for example, for uniform point distibutions in simple domains, Halton sequences are a popular point selection method

- CVT point selection strategies offer substantially more flexibilitiy than most exisiting methods in that they can be used for
  - general domains
  - nonuniform and anisotropic point distributions
  - some or all points can be placed on bounding surfaces

Support radii and overlap determination

- Having chosen a set of K points $\{x_i\}_{i=1}^K$ in the domain $\overline{\Omega} \subset \mathbb{R}^N$, we want to now choose, for each point x_i, an associated radius h_i

- For each point x_i, we then define the sphere centered at x_i and having radius h_i

$$S_i = \{\ y \in \mathbb{R}^N \quad : \quad |y - x_i| \leq h_i\ \}$$

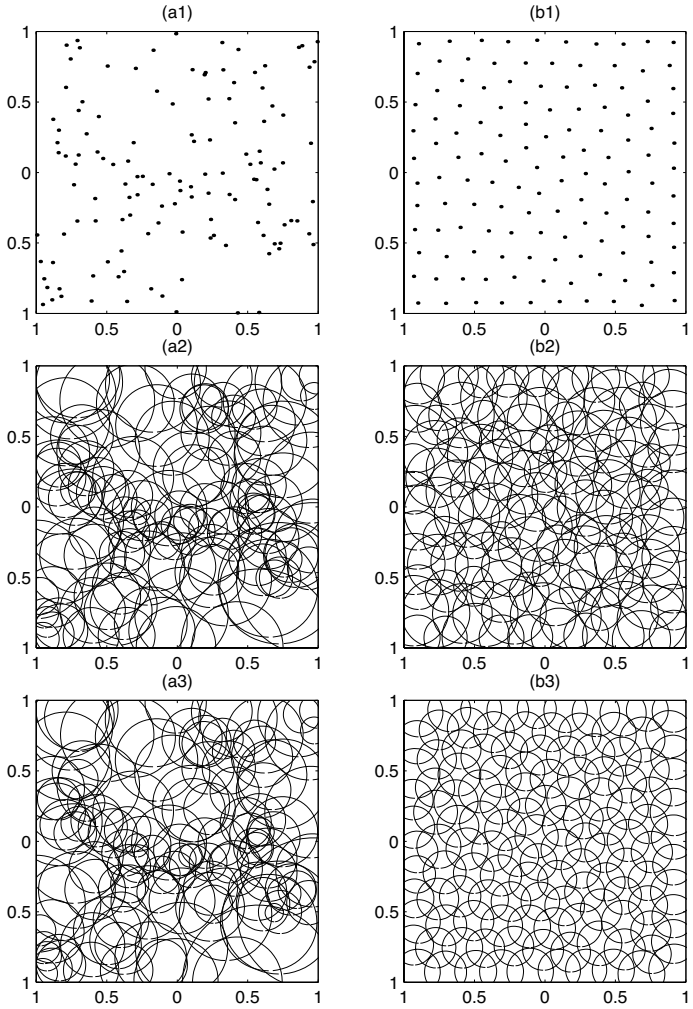
  - other patch shapes associated with points can be used, e.g., ellipsoids, rectangles, or bricks
  - the sphere (or other patch) associated with a point will determine the support region for basis function associated with the point

- The selection of the radii $\{h_i\}_{i=1}^K$ should meet two requirements:
  - the union of the spheres $\{S_i\}_{i=1}^K$ should cover $\overline{\Omega}$ a specified number of times
  - if the intersection $S_i \cap S_j$ of two distinct spheres is not empty, it should not be "too small" nor "too large"

- Algorithms are known (notably due to C. Duarte and J. Oden and to M. Griebel and M. Schweitzer) for support radii determination
  - these algorithms involve a background mesh
  - they do not give good results for nonuniform point distributions

- A new, (totally meshless) algorithm for determining support radii has been developed that
  - guarantees, in a probabilistic sense, that the patches cover every point in the domain p-times, where p is an input integer
    - some meshless methods for PDE's require that each point in the region be covered several times by the support spheres associated with the point set
  - allows for control of the amount of overlap of the support regions
  - gives good results for nonuniform point distributions in general regions

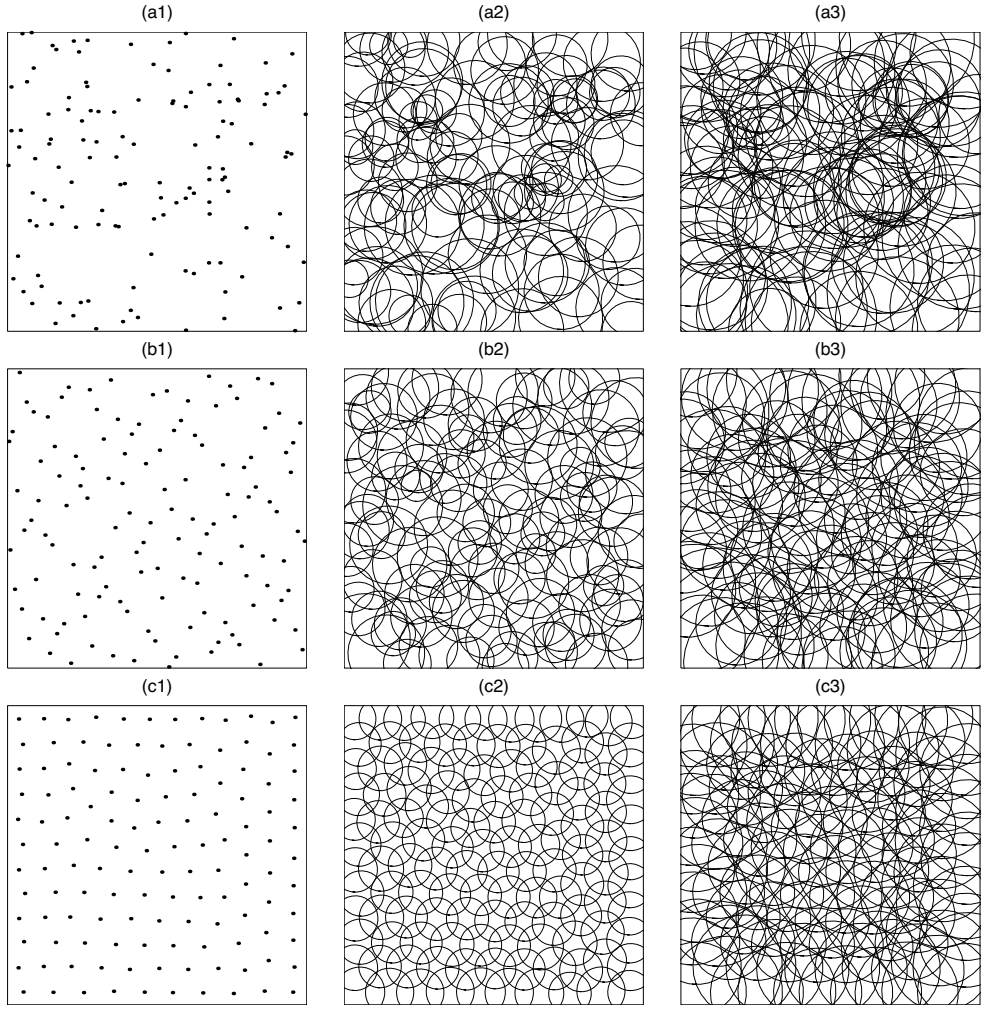
- Although we illustrate using circular patches in 2D, there is no difficulty treating 3D and/or other shaped patches, e.g., rectangles or bricks



Sets of 128 uniformly distributed points in the square and the corresponding sphere coverings; Monte Carlo (left) and CVT (right) point distributions; old support region algorithm due to C. Duarte and J. Oden and also M. Griebel and M. Schweitzer (middle); new algorithm (bottom)

No. of points	Support region algorithm	Point selection method	
		Monte Carlo	CVT
128	old	13.33%	10.66%
	new	12.26%	5.21%
256	old	6.61%	4.87%
	new	6.24%	2.58%
512	old	3.10%	2.49%
	new	2.81%	1.25%

Percentages of nonzero elements in the characteristic matrix for a uniform distribution of points in a square domain

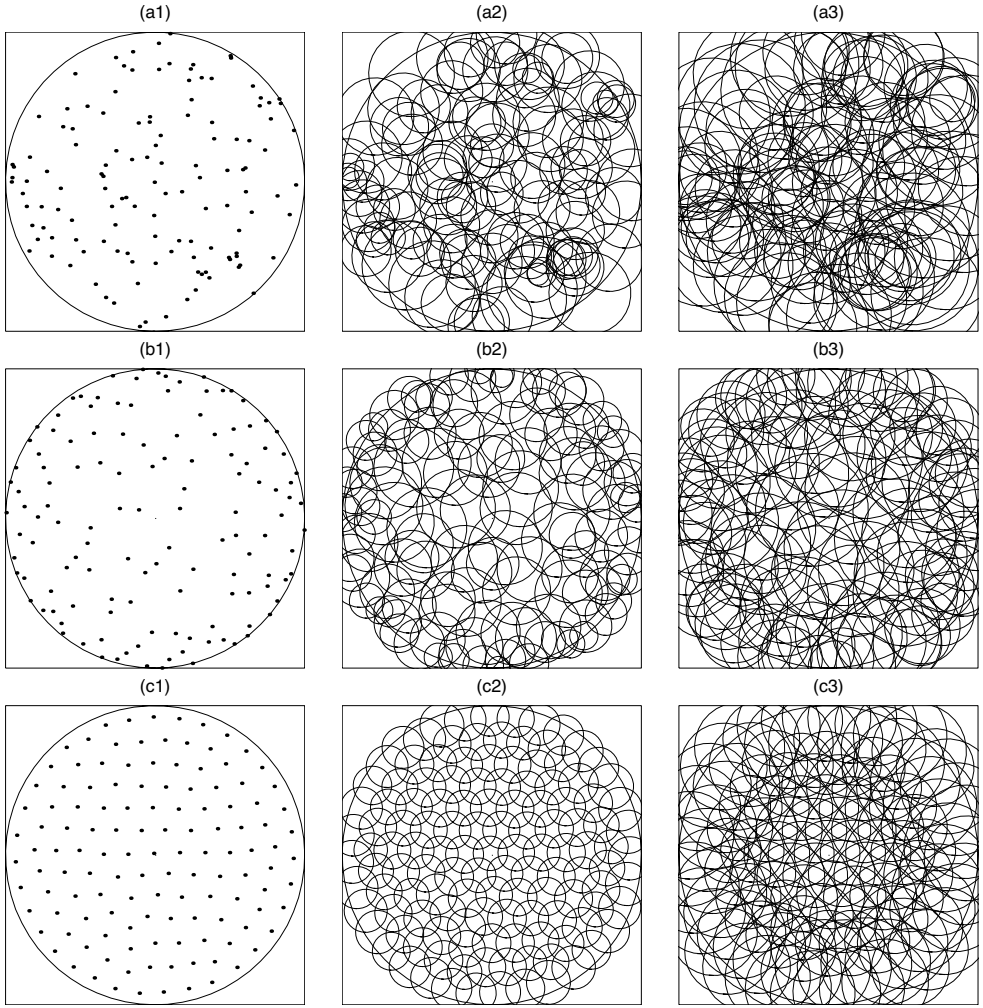


Sets of 128 points in a square (left) and the associated spherical patches determined by the new algorithm for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top), (2,3) Halton sequence (middle), and CVT (bottom) point selection methods for a uniform density function

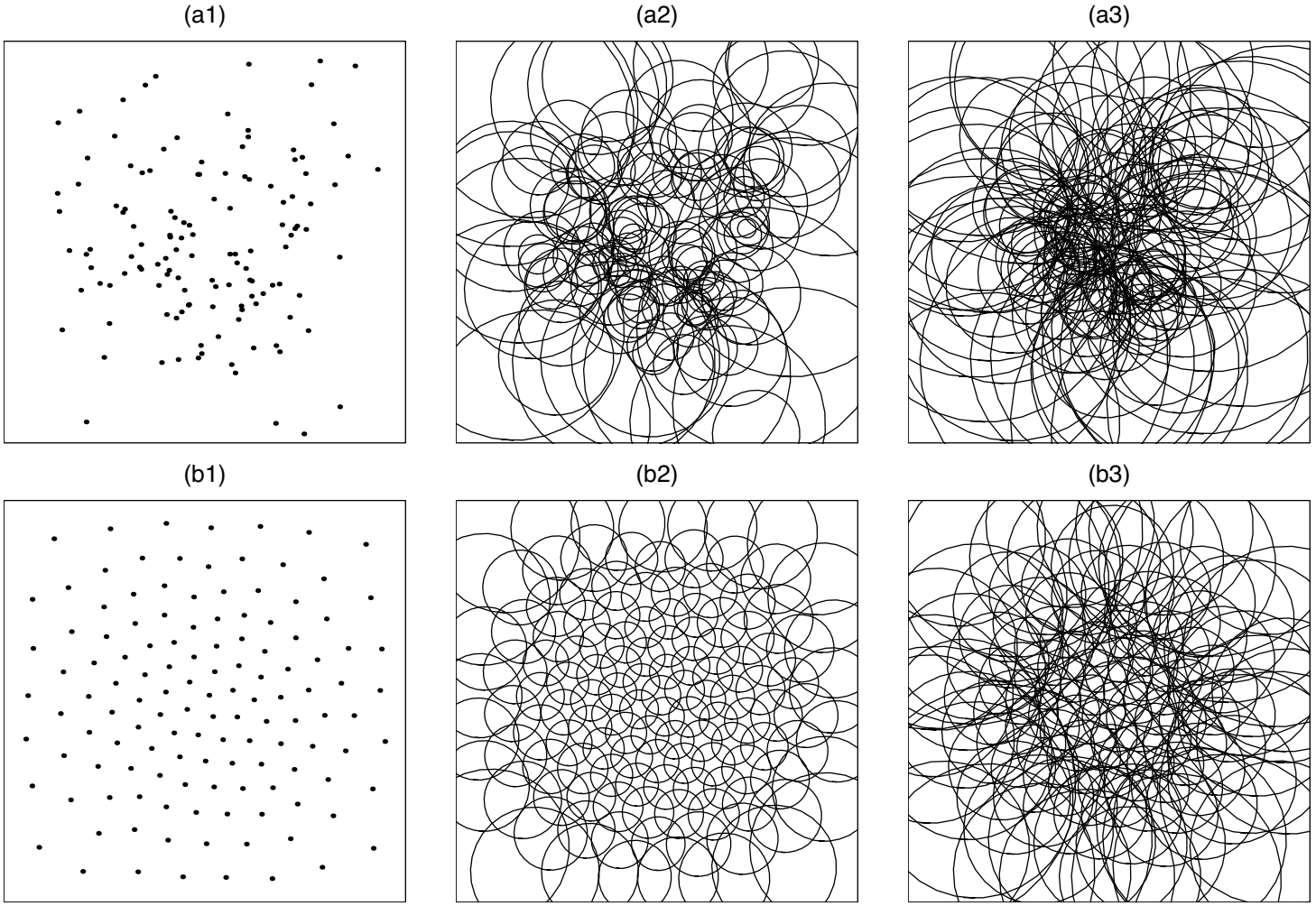
No. of points	p	Point selection method		
		Monte Carlo	Halton sequence	CVT
128	1	12.02%	10.04%	5.05 %
	3	25.41%	21.63%	18.68%
256	1	6.25%	4.95%	2.61%
	3	14.23%	10.97%	9.53%
512	1	3.29%	2.72%	1.33%
	3	7.30%	5.91%	4.72%

Percentages of nonzero elements in the characteristic matrices resulting from the new support region determination algorithm for a uniform distribution of points in a square domain

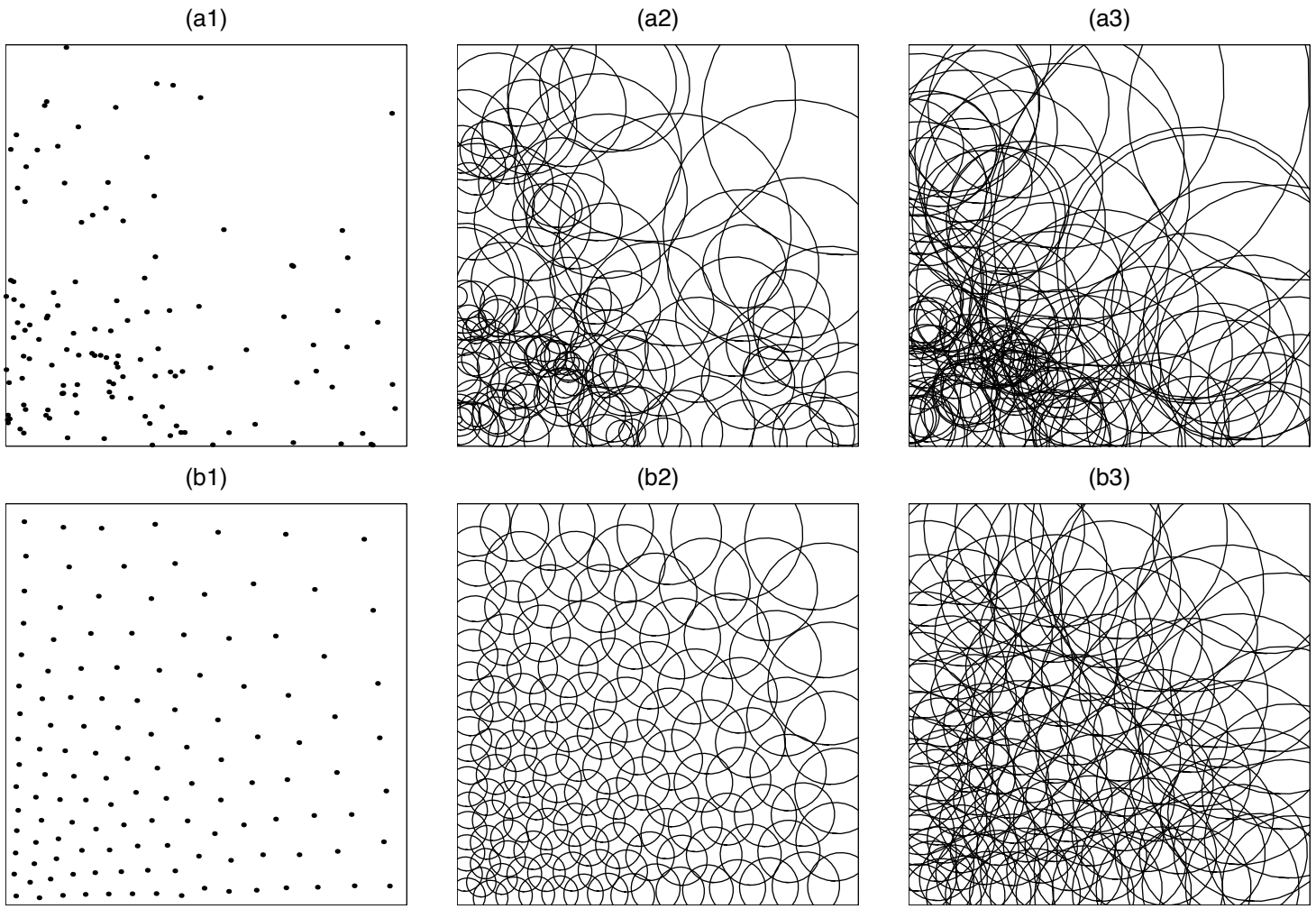
- The CVT numbers using the new support region determination algorithm are almost identical to what one obtains
  - for linear finite element discretizations in the case $p = 1$ (a 1-covering of the domain)
  - for quadratic finite element discretizations in the case $p = 3$ (a 3-covering of the domain)



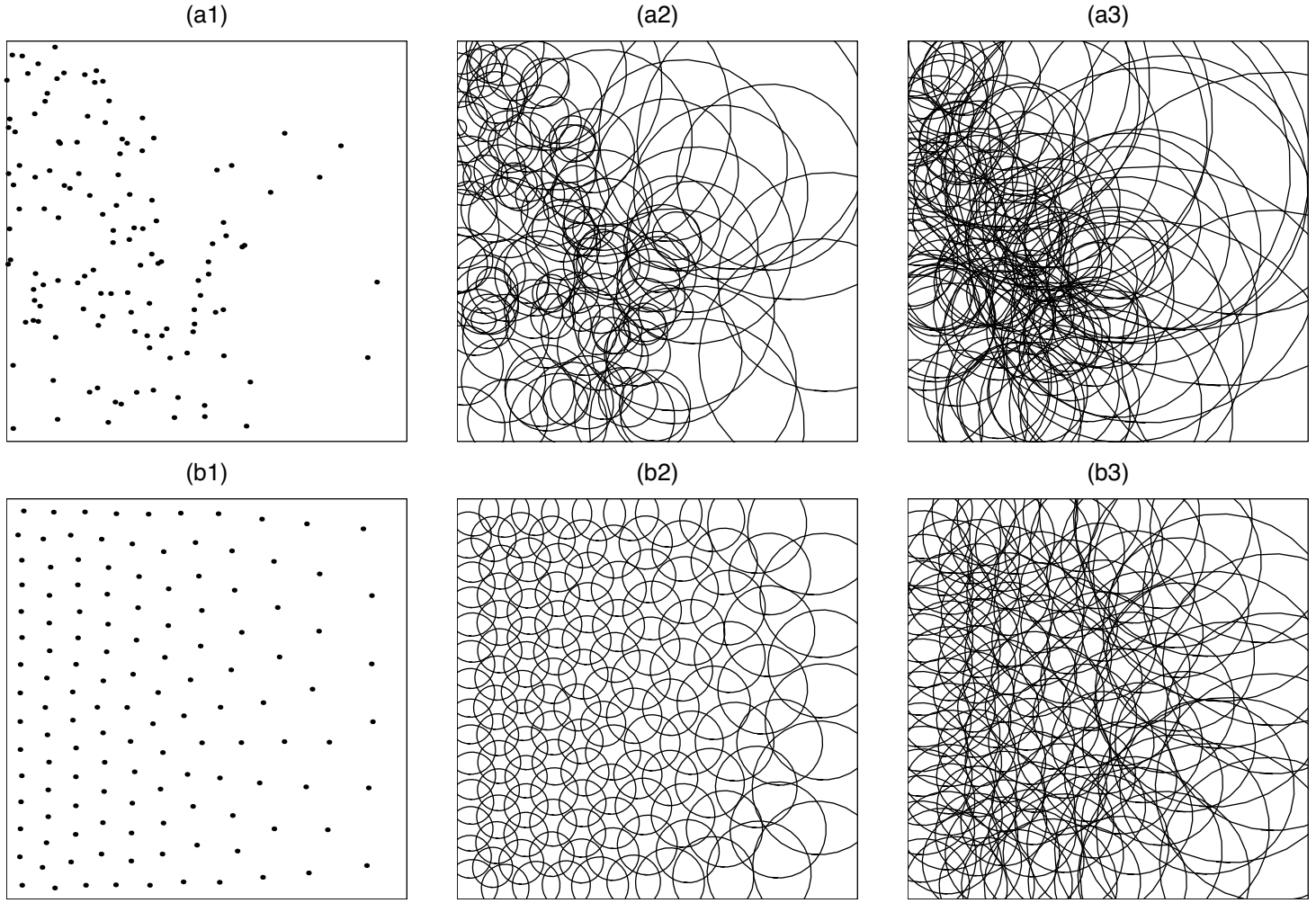
Sets of 128 points in a circle (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top), (2,3) Halton sequence (middle), and CVT (bottom) point selection methods for a uniform density function



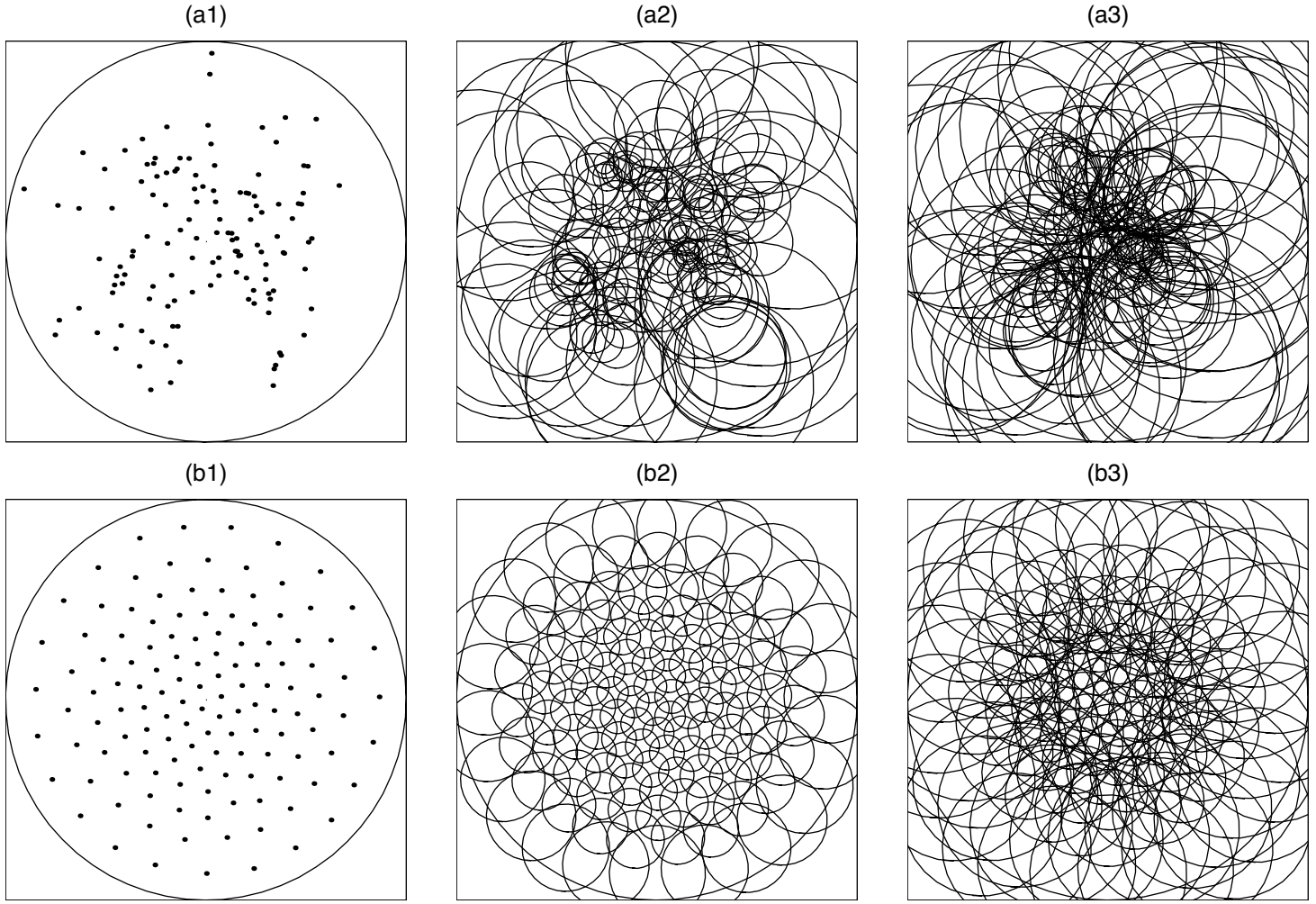
The sets of 128 points in a square (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-3(x^2+y^2)}$



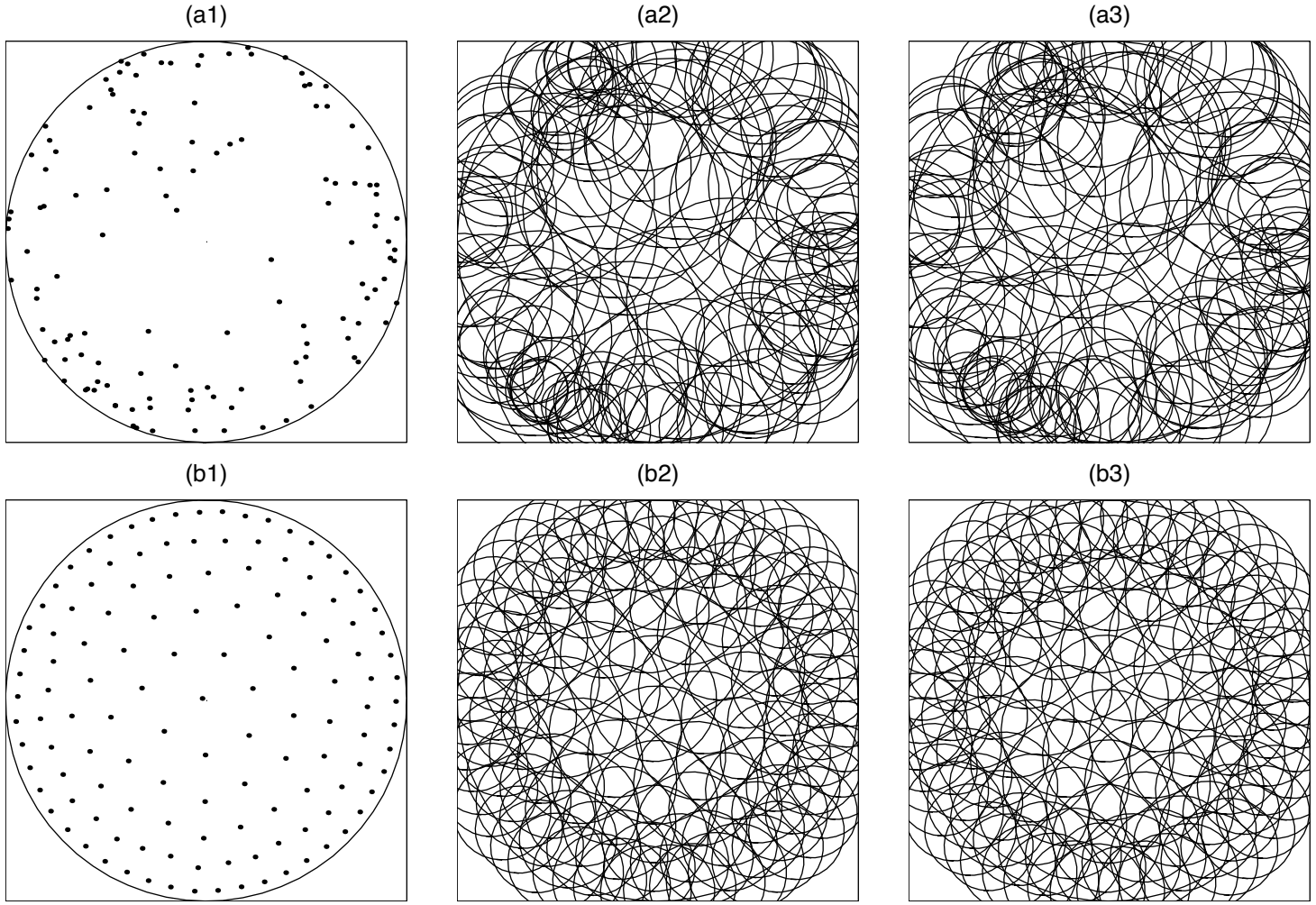
The sets of 128 points in a square (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-1.5(2+x+y)}$



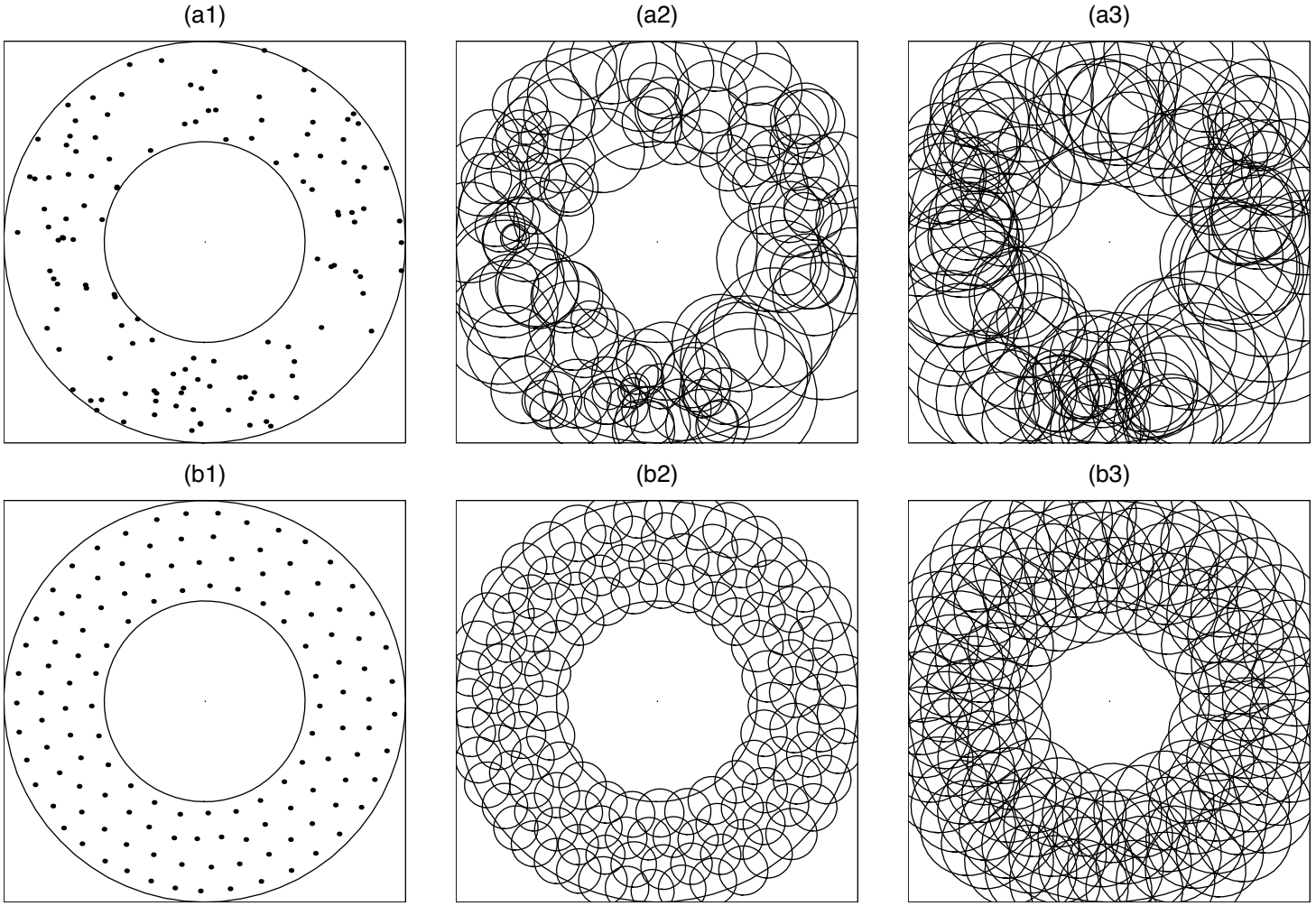
The sets of 128 points in a square (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-(1+x)^2}$



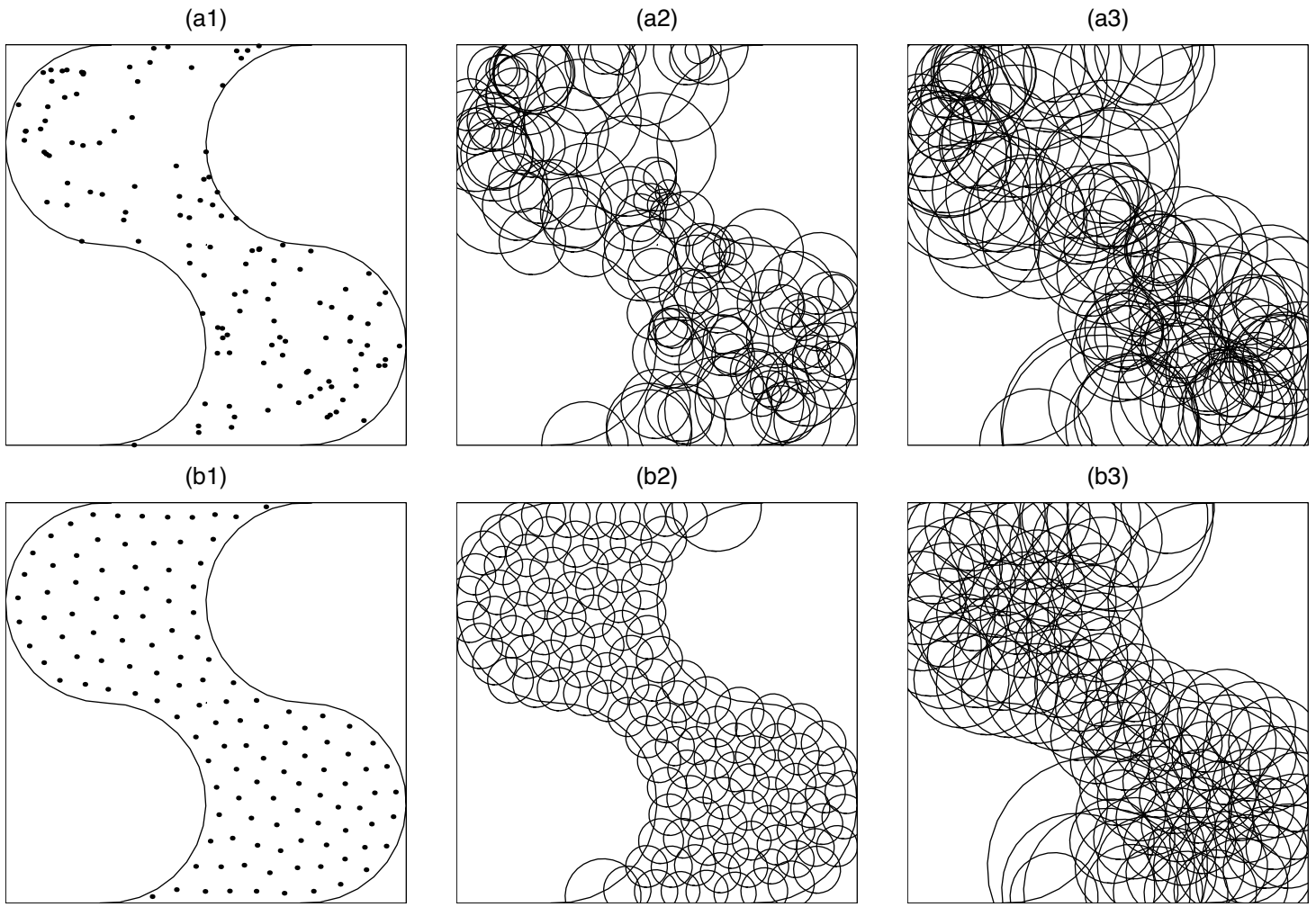
The sets of 128 points in a circle (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-4(x^2+y^2)}$



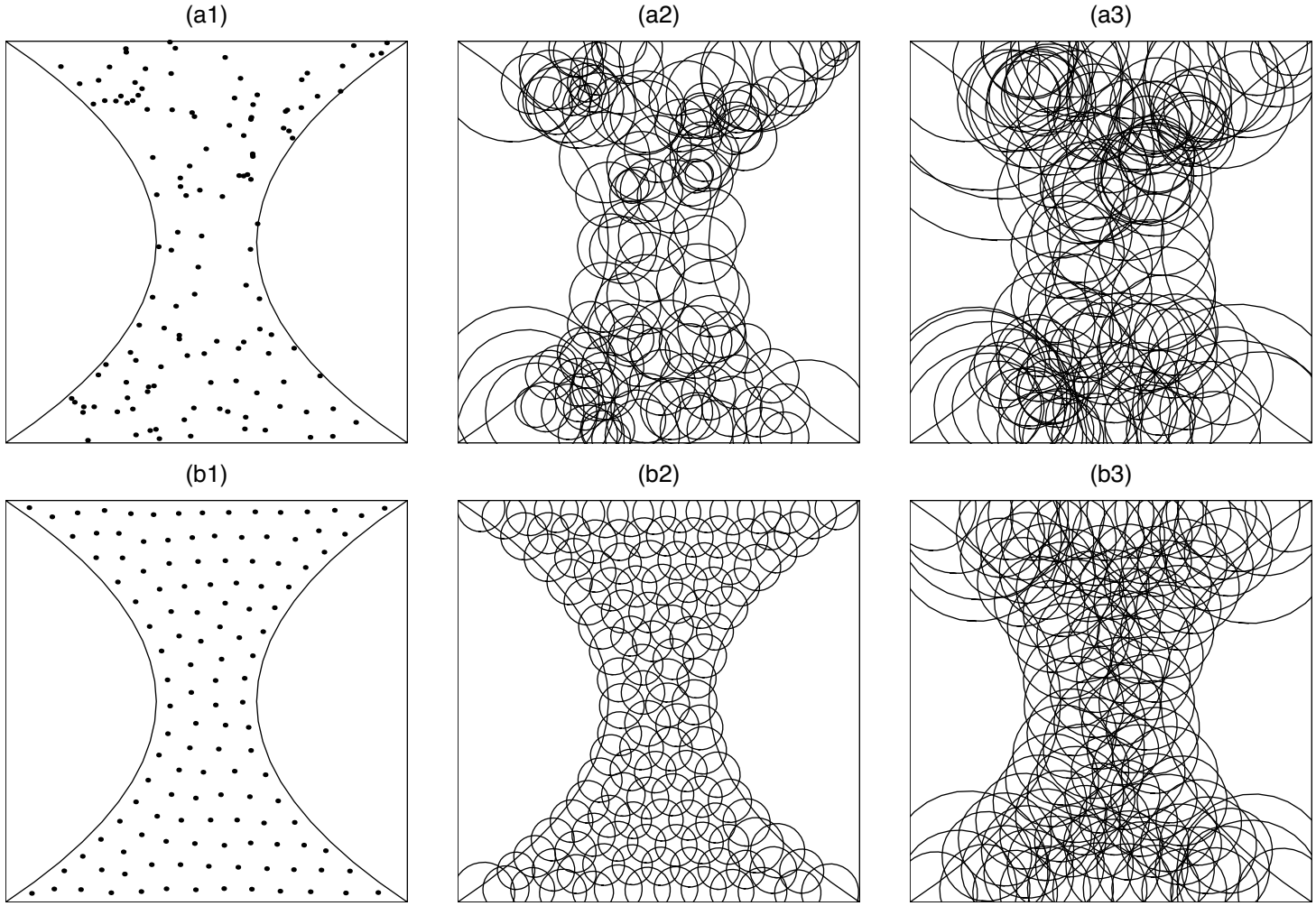
The sets of 128 points in a circle (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $e^{-3(1-x^2-y^2)}$



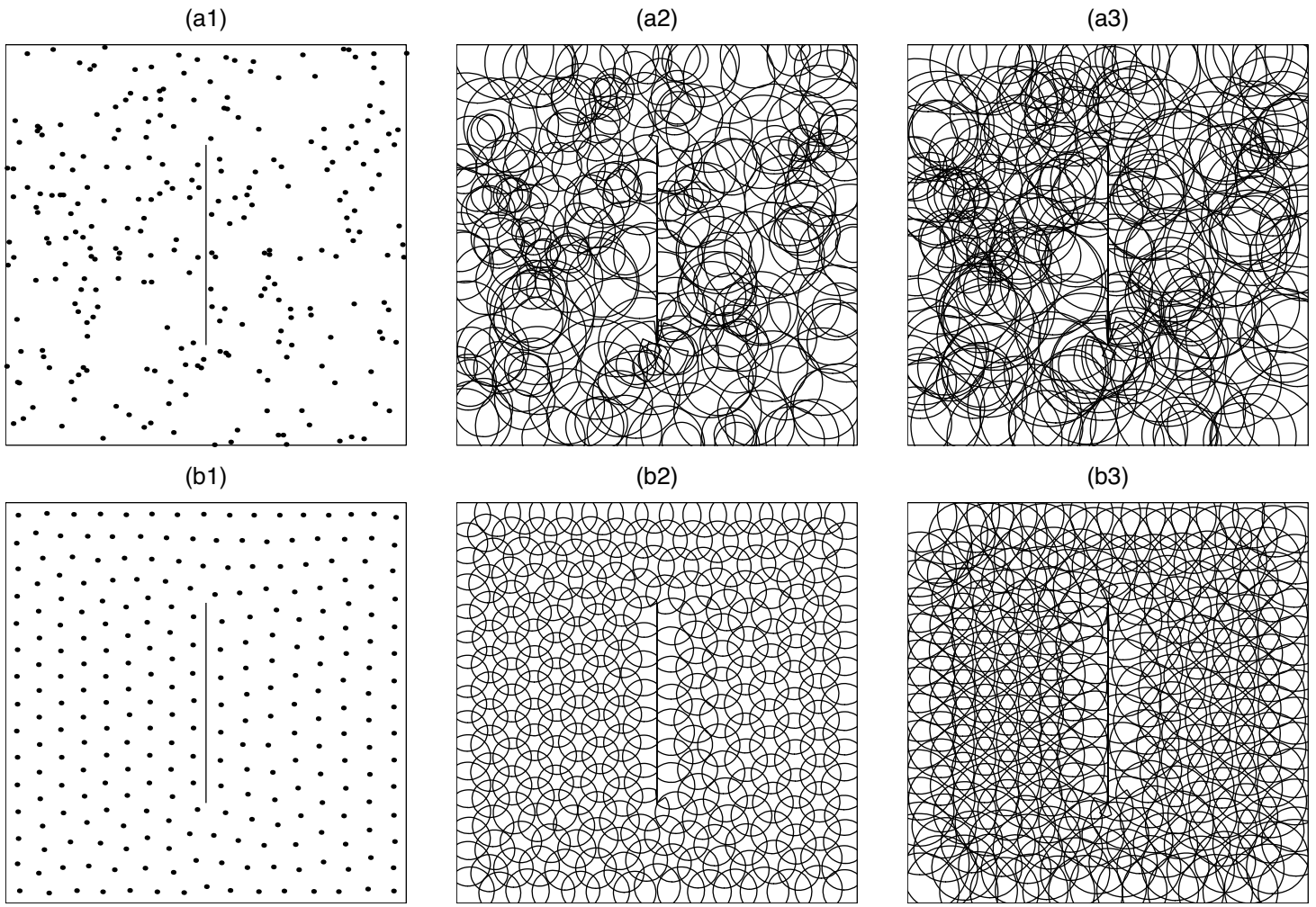
The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



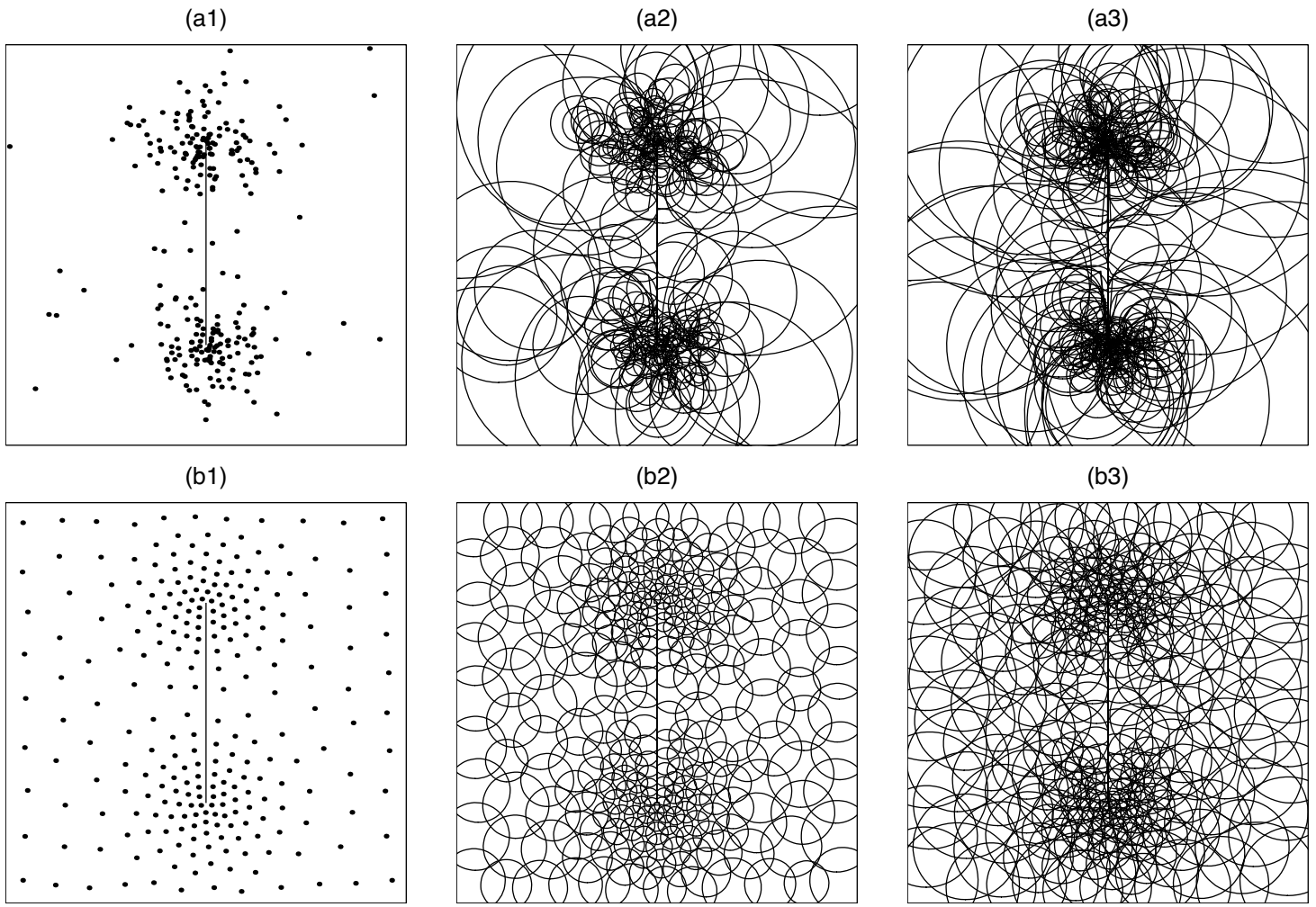
The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



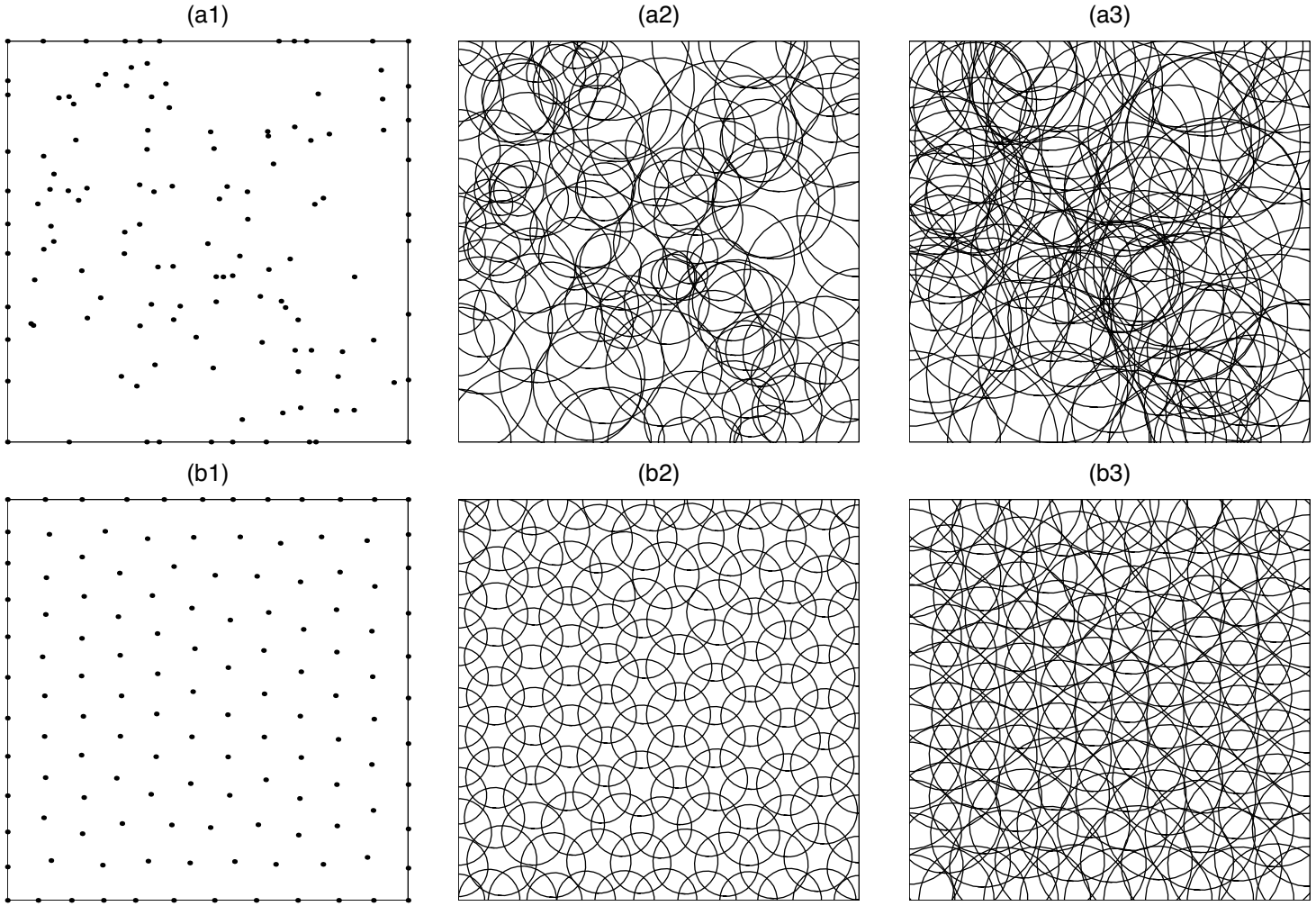
The sets of 128 points in a non-convex domain (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



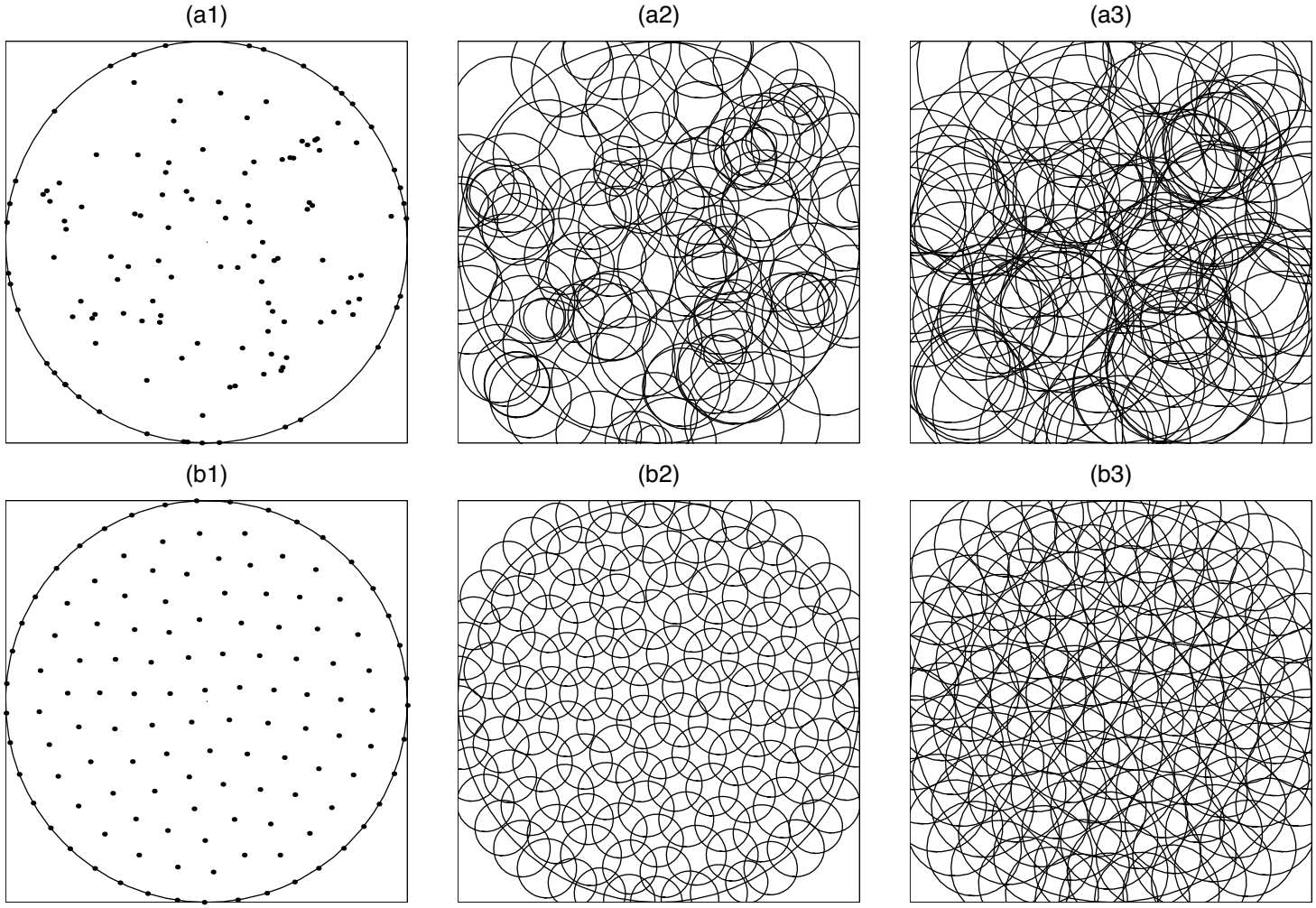
The sets of 256 points in a domain with a slit (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



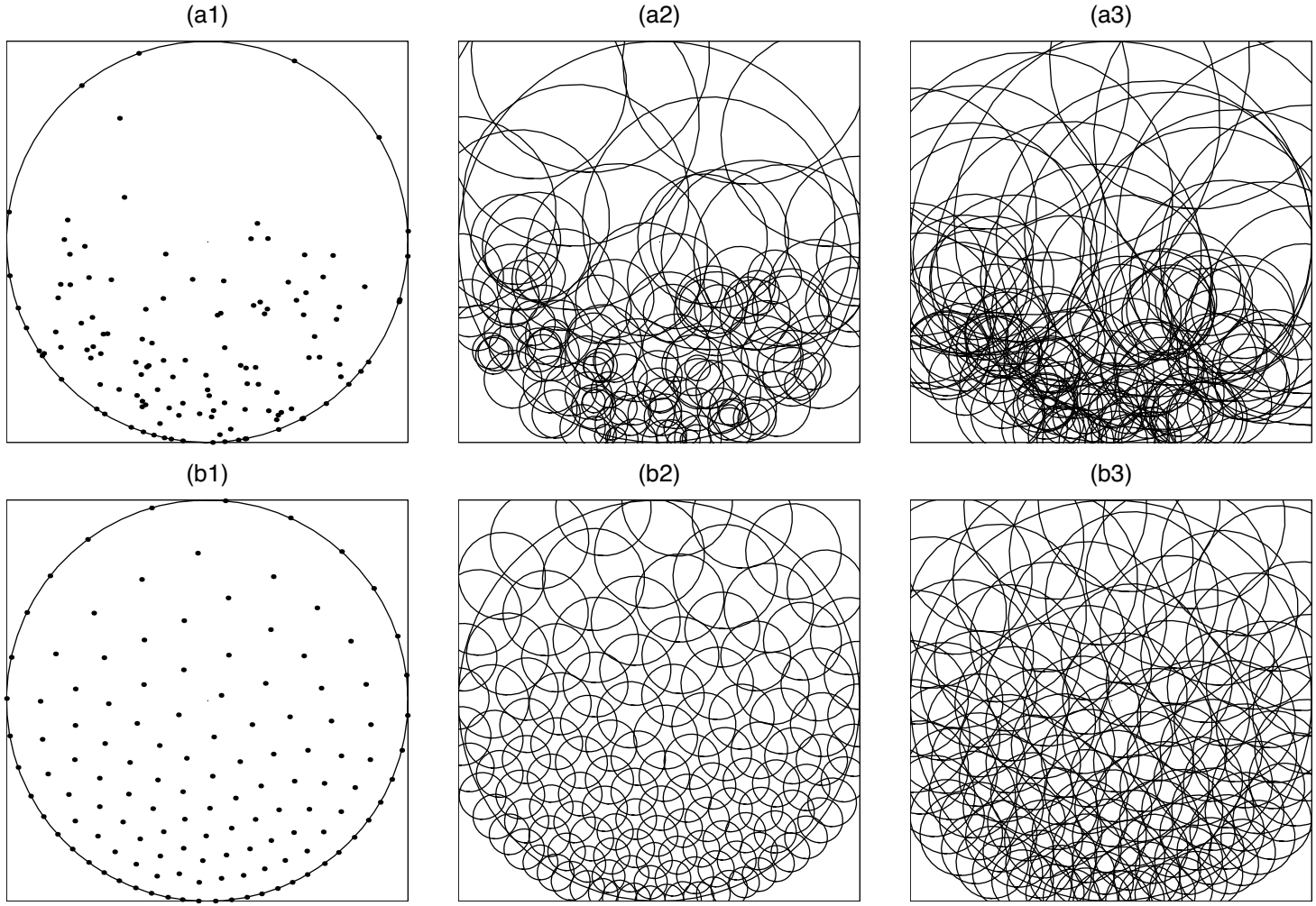
The sets of 256 points in a domain with a slit (left) and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a nonuniform density function



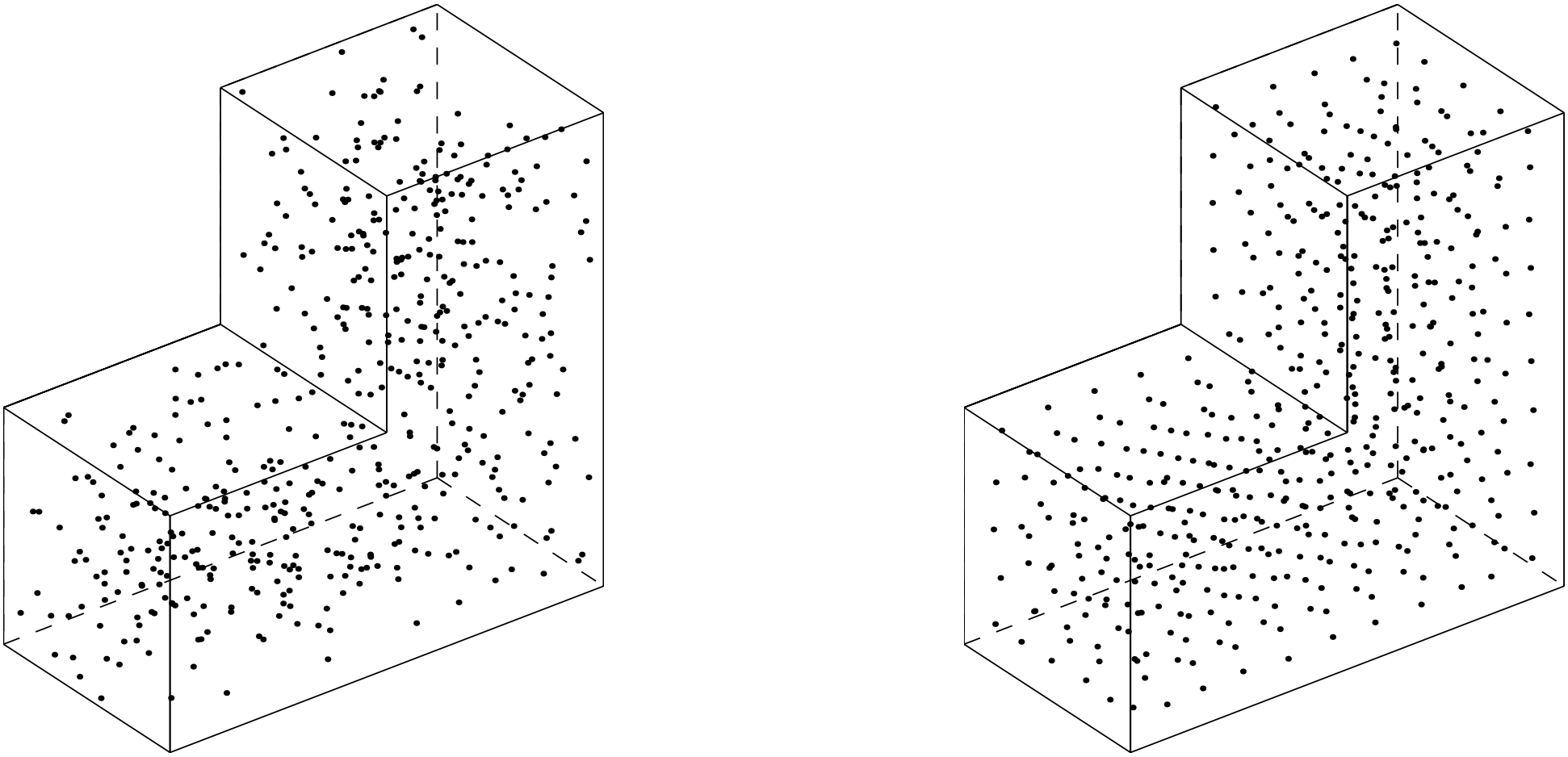
The sets of 128 points in a square (left) under constraint and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



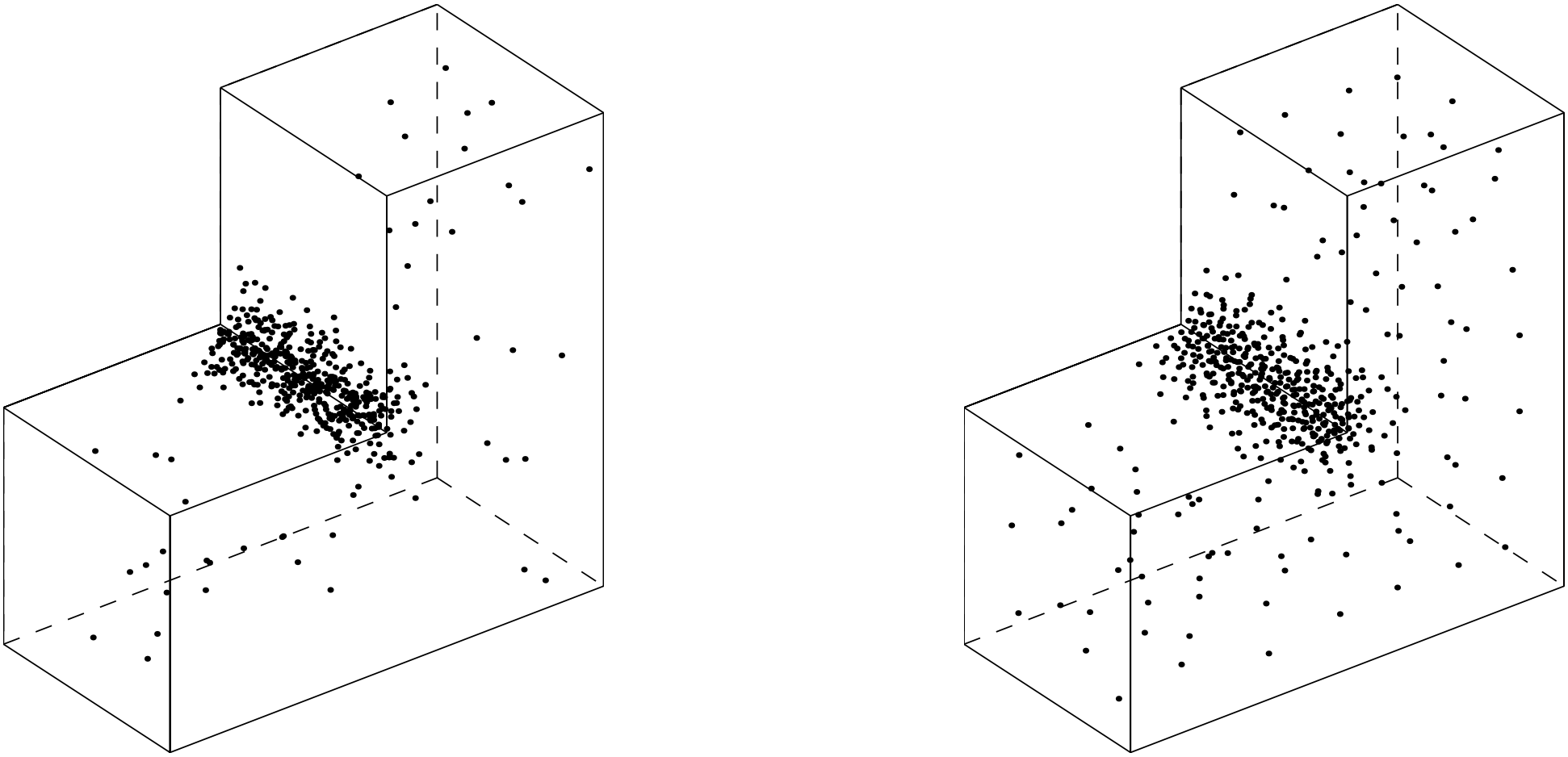
The sets of 128 points in a circle (left) under constraint and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for a uniform density function



The sets of 128 points in a circle (left) under constraint and the associated spherical patches determined by the new algorithms for $p = 1$ (middle) and $p = 3$ (right) for the Monte Carlo (top) and CVT (bottom) point selection methods for the density function $\rho(x, y) = e^{-5.3(1+y)}$



The sets of 512 points in a three-dimensional non-convex domain for the Monte Carlo (left) and CVT (right) point selection methods for a uniform density function



The sets of 512 points in a three-dimensional non-convex domain for the Monte Carlo (left) and CVT (right) point selection methods for a nonuniform density function

GRID GENERATION

- The dual Delaunay meshes corresponding to CVT's are well suited for triangular and tetrahedral mesh generation
  - CVT and the dual Delaunay mesh pair are well suited to finite volume and edge element type discretizations
- In order to render them useful for mesh generation, CVT concepts and algorithms have been extended in several directions so that the CVT methodology can handle
  - automatically locating a subset of the points on the boundary
  - generating interior points for a specified surface point distribution
  - nonuniform point distributions
- CVT point generation is quite cheap (?)
  - on a medium workstation (4 years old), it takes a few hours to determine 500,000 points in 3D
  - this is without taking advantage of several algorithmic optimization opportunities or of the embarassingly inherent parallelism in the algorithms

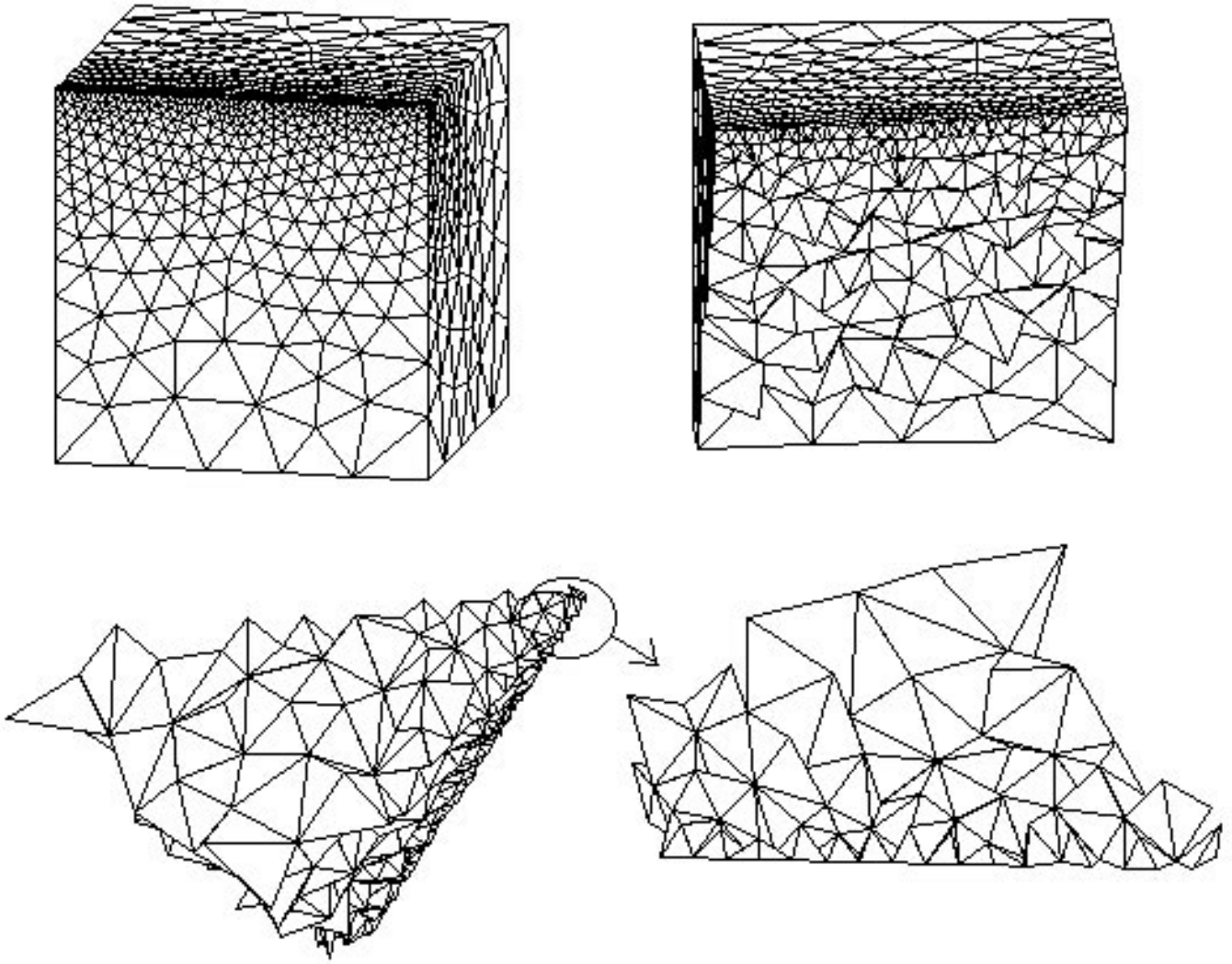
- The key and the curse of the effectiveness of CVT mesh generation can be found in Gersho's conjecture
  - the fact that the point distribution is always locally uniform makes it unlikely that weird shaped elements, e.g., slivers, will be generated
  - on the other hand, it seems that the local uniformity of the mesh also means that anisotropic meshes, e.g., for fluid boundary layers, cannot be generated

- Gersho's conjecture assumes that one uses the standard Euclidean norm to measure the distance between two points
  - if one instead uses anisotropic metrics, e.g.,

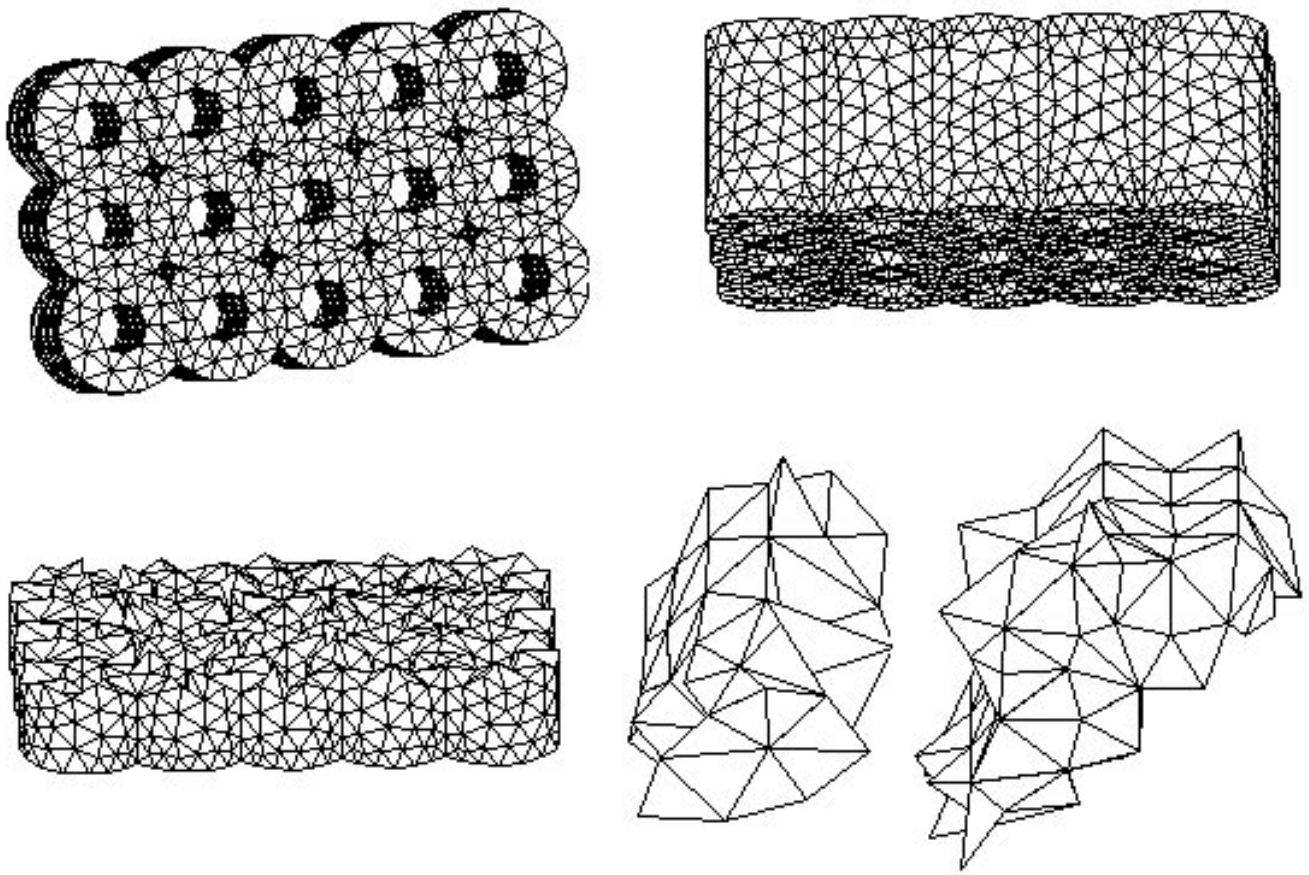
    $$\mu_1(y_1, y_2, y_3)(z_1 - y_1)^2 + \mu_2(y_1, y_2, y_3)(z_2 - y_2)^2 + \mu_3(y_1, y_2, y_3)(z_3 - y_3)^2$$

    one can indeed generate anisotropic CVT grids
  - this is currently under study

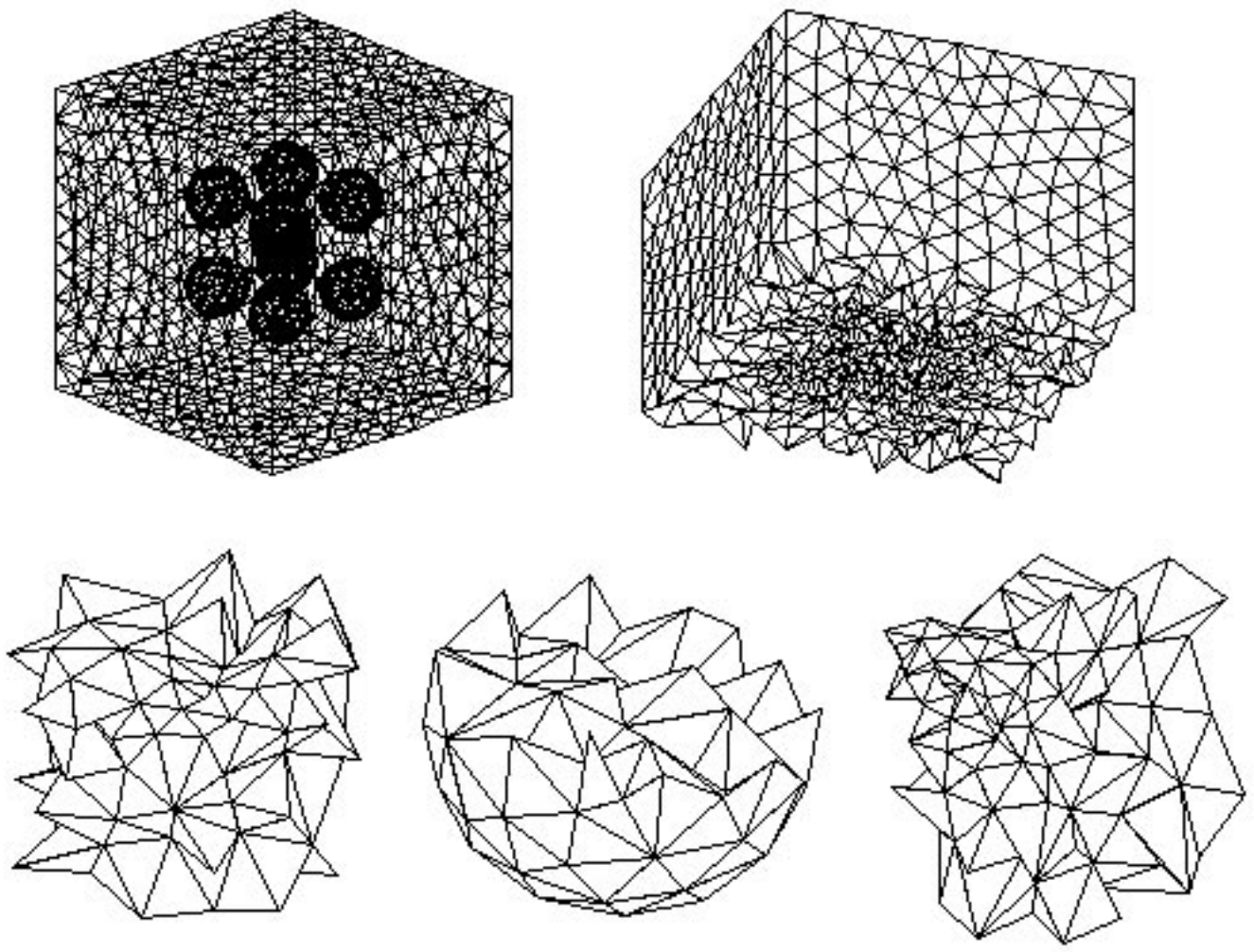
- Another subject under study is connecting the CVT density function and metric function coefficients to a posteriori error estimators to enable CVT adaptive grid generation



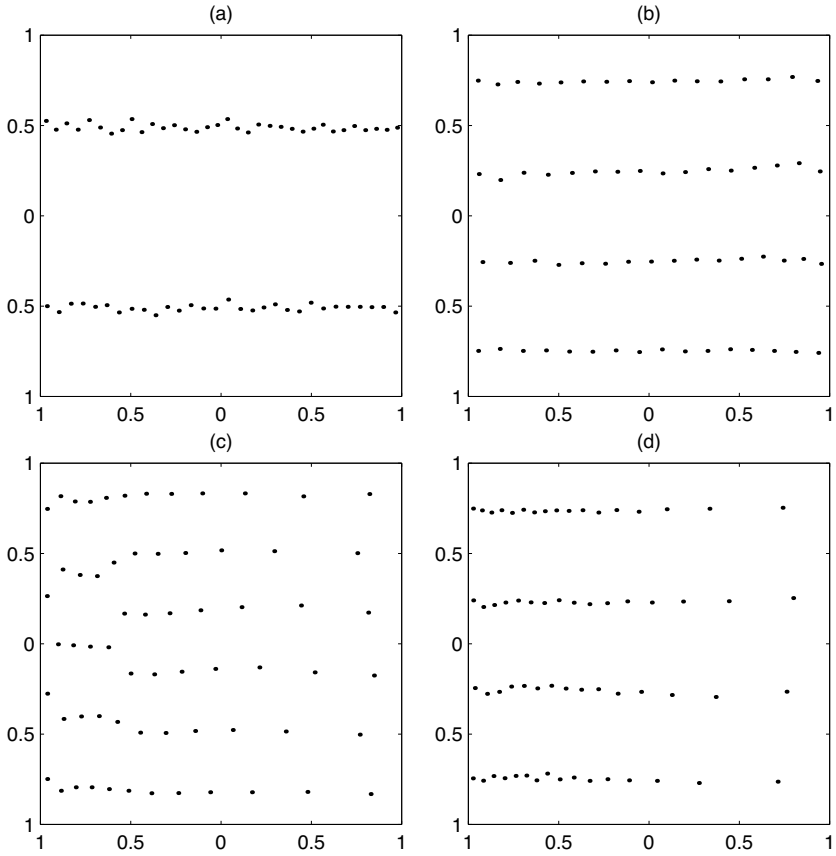
Tetrahedral mesh in a cube



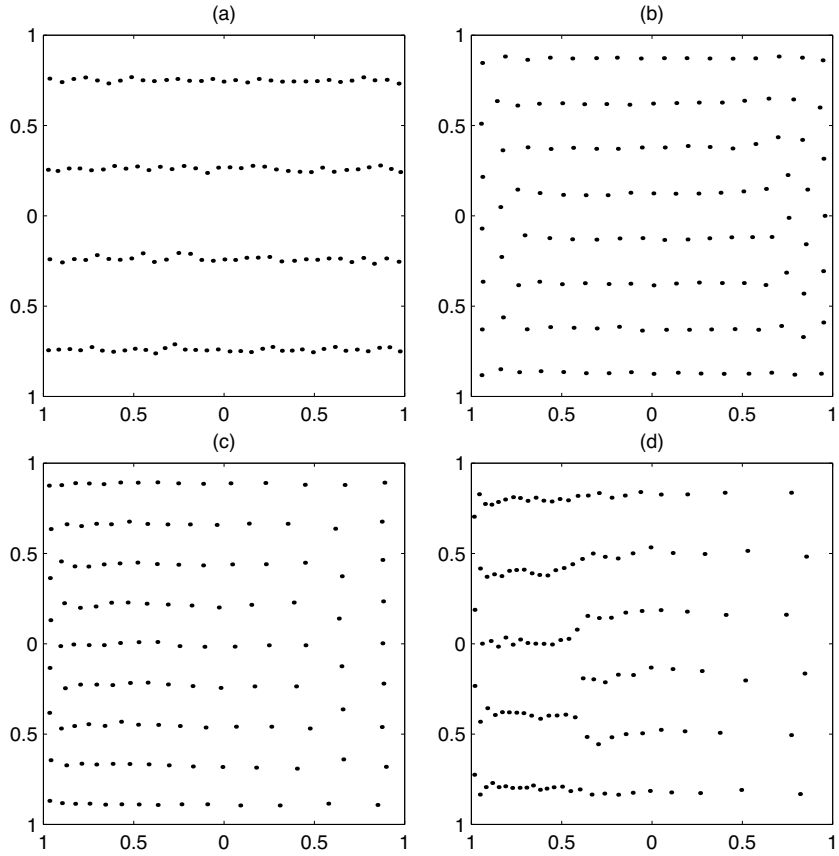
Tetrahedral mesh in a more complex region



Tetrahedral mesh for a composite material simulations



Anisotropic [uniform (top) and nonuniform (bottom)] distributions of 64 points in the square $[-1,1]^2$ (a) $256(z_1-y_1)^2+(z_2-y_2)^2$ (b) $20(z_1-y_1)^2+(z_2-y_2)^2$ (c) $\left(1+20e^{-2(y_1+1)^2}\right)(z_1-y_1)^2+(z_2-y_2)^2$ (d) $\left(1+100e^{-2(y_1+1)^2}\right)(z_1-y_1)^2+(z_2-y_2)^2$



Anisotropic [uniform (top) and nonuniform (bottom)] distributions of 128 points in the square $[-1,1]^2$ (a) $80(z_1-y_1)^2+(z_2-y_2)^2$ (b) $5(z_1-y_1)^2+(z_2-y_2)^2$ (c) $\left(1+10e^{-2(y_1+1)^2}\right)(z_1-y_1)^2+(z_2-y_2)^2$ (d) $\left(1+150e^{-2(y_1+1)^2}\right)(z_1-y_1)^2+(z_2-y_2)^2$

POINT DISTRIBUTIONS AND GRID GENERATION ON SURFACES

- In many applications, point distributions on surfaces are needed

- In order to generalize CVT's to surfaces, two main ingredients are needed
  - the generalization of the concept of Voronoi regions to surfaces
  - the generalization of the concept of mass centroids to surfaces

- There are a number of ways to do each of these
  - we choose generalizations which are "easy" to use

- We consider compact and continuous surfaces $\mathbf{S} \subset \mathbb{R}^N$

- Given a set of points $\{\mathbf{z}_i\}_{i=1}^K \in \mathbf{S}$, we define their corresponding Voronoi regions on $\mathbf{S}$ by

$$V_i = \{\, \mathbf{x} \in \mathbf{S} \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, K,\ j \neq i \,\}$$

for $i = 1, \ldots, K$

- For each Voronoi region V_i, we call $\mathbf{z}_i^c$ the constrained mass centroid of V_i on $\mathbf{S}$ if $\mathbf{z}_i^c$ is a solution of the following problem:

$$\min_{\mathbf{z} \in \mathbf{S}} F_i(\mathbf{z})\,, \qquad \text{where} \qquad F_i(\mathbf{z}) = \int_{V_i} \rho(\mathbf{x}) |\mathbf{x} - \mathbf{z}|^2 \, d\mathbf{x}$$

- We call a Voronoi tessellation a constrained centroidal Voronoi tessellation (CCVT) if and only if the points $\{\mathbf{z}_i\}_{i=1}^K$ which serve as the generators of the Voronoi regions $\{V_i\}_{i=1}^K$ are also the constrained mass centroids of those regions

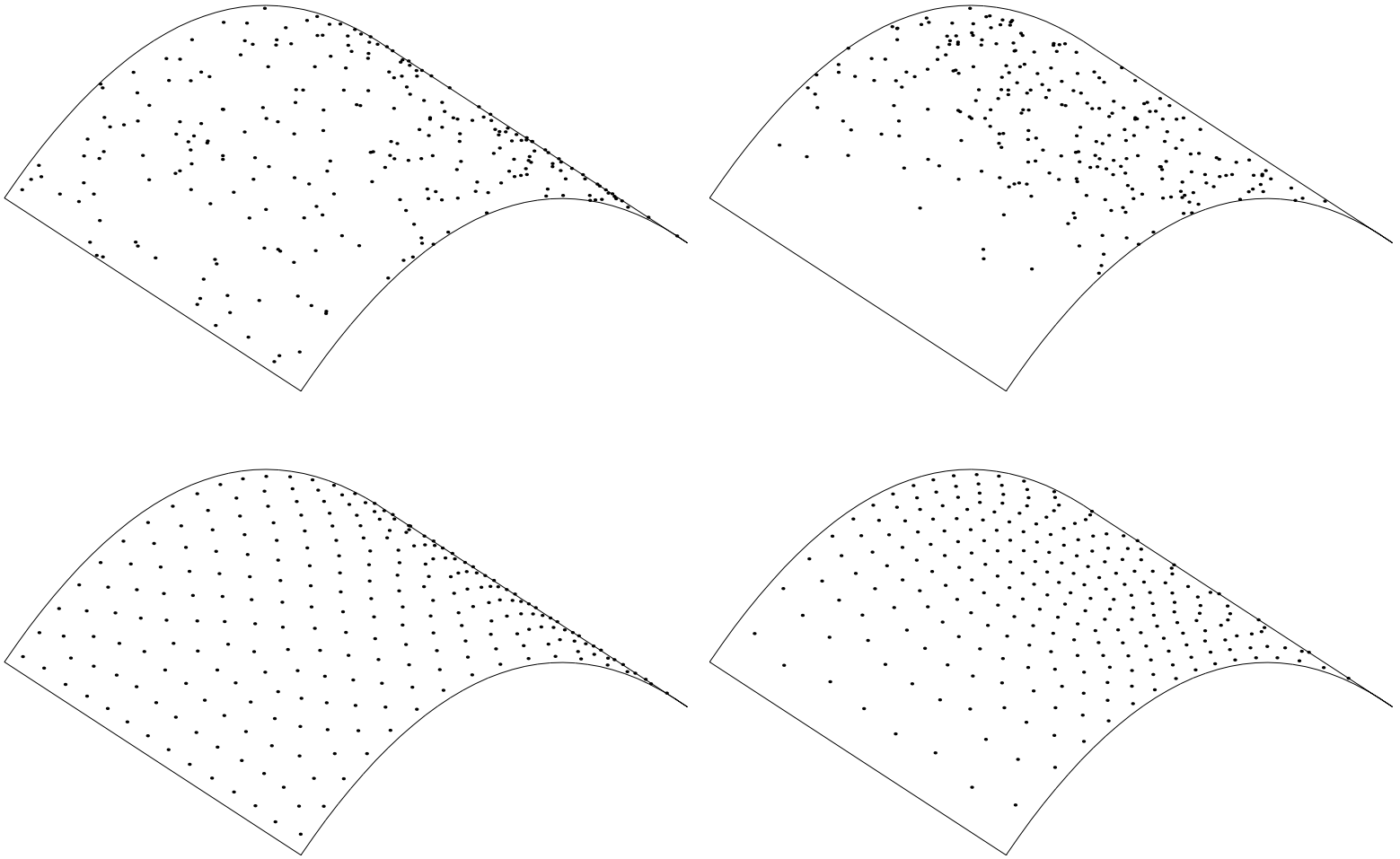
- Note that the definition of CCVT implies that
  - generators are constrained to the surfaces
  - but distances are still the standard Euclidean distances, not the more general geodesic distances

- Constrained centroids are "easy" to construct by normal projection

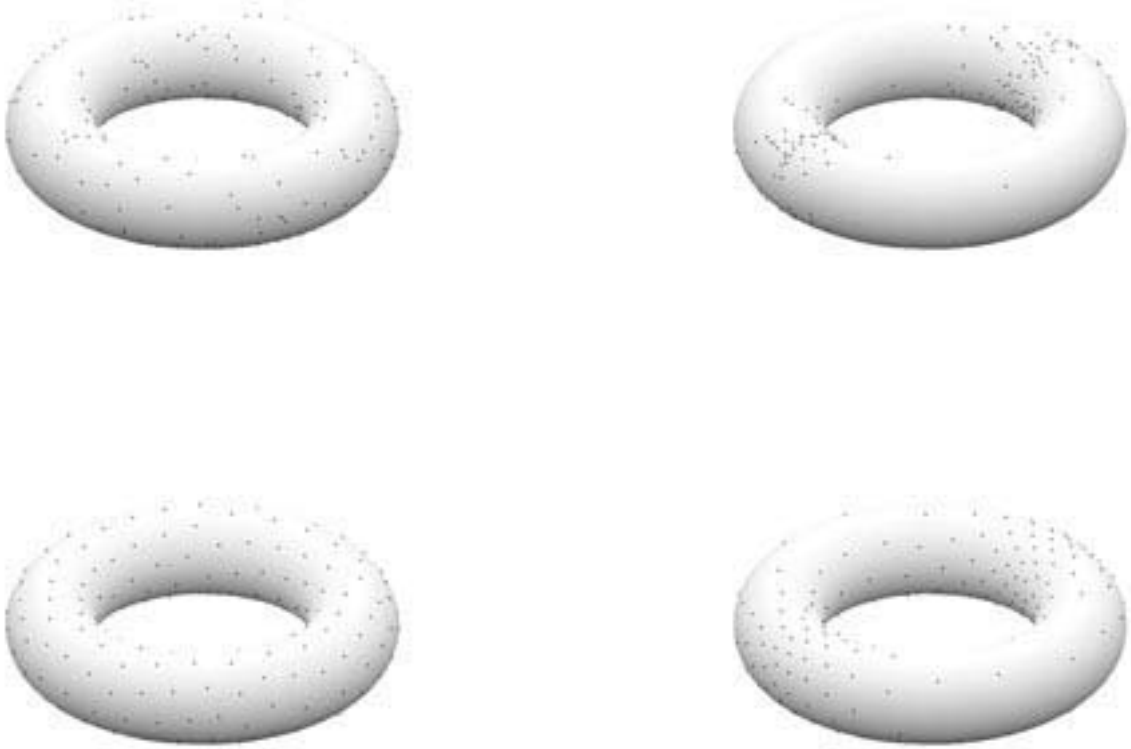
- Constrained CVT's defined in the manner above enjoy an optimization property in much the same way as do CVT's

- Algorithms (deterministic and probabilistic, serial and parallel) for CVT's may be then easily generalized to the case of CCVT's

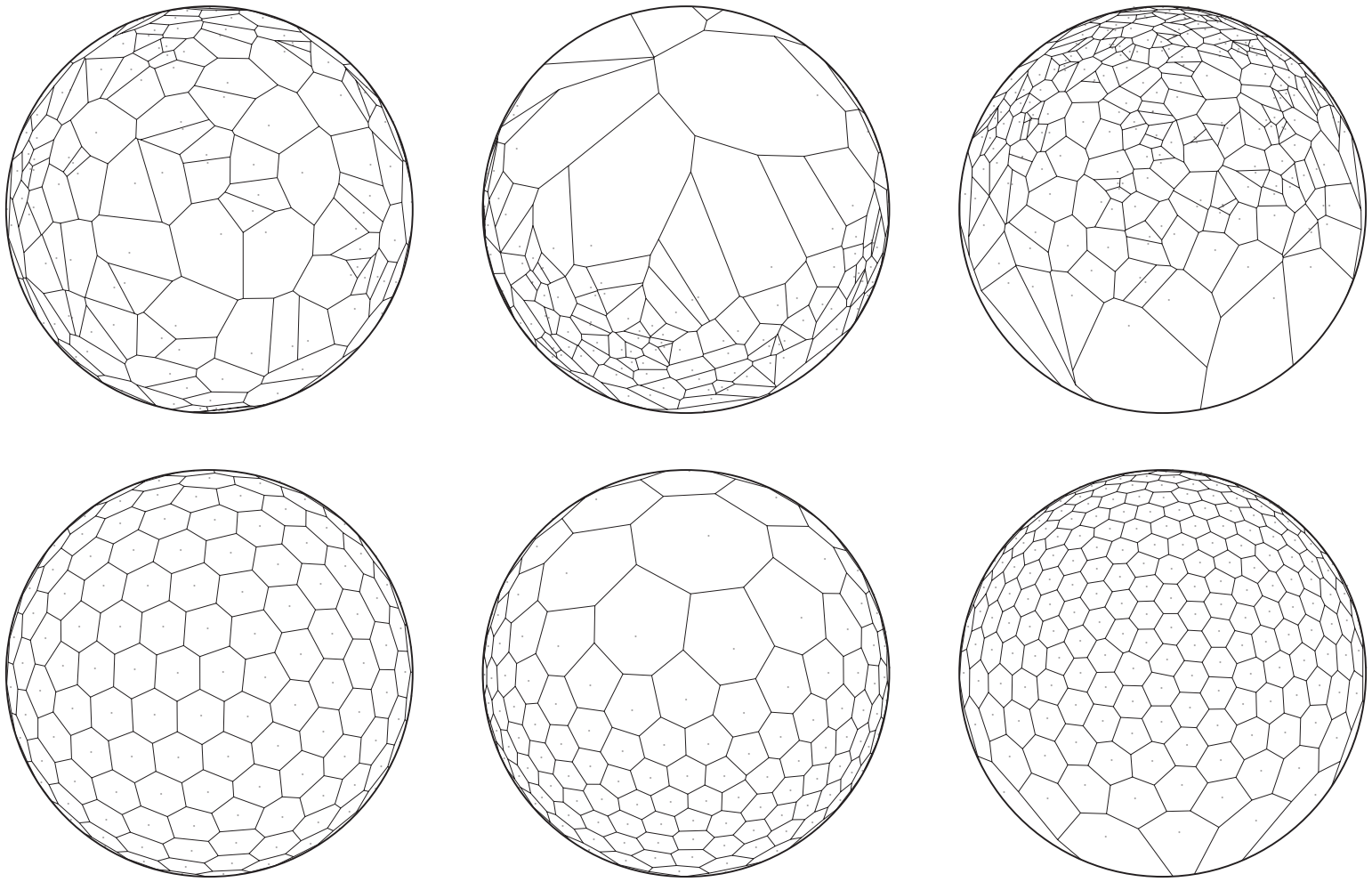
- We illustrate the use of CCVT's on three surfaces
  - a developable surface
  - a torus
  - the sphere



Voronoi diagrams for 256 generators on a developable surface
random sampling (top) constrained CVT (bottom)
$\rho(x, y, z) = 1$ (left) $\rho(x, y, z) = e^{-20.0x^2}$ (right)



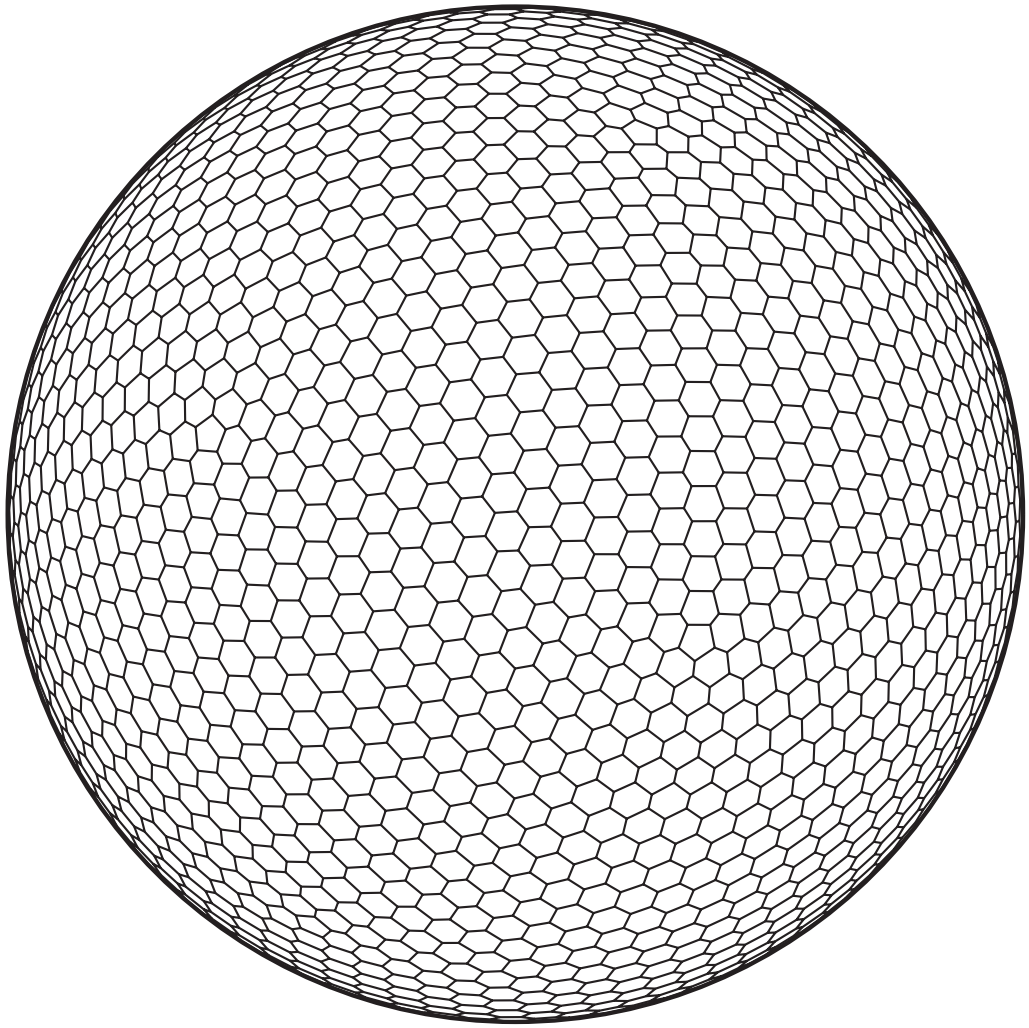
Voronoi diagrams for 256 generators on a torus
random sampling (top) constrained CVT (bottom)
$\rho(x, y, z) = 1$ (left) $\rho(x, y, z) = e^{-5.0|y|}$ (right)



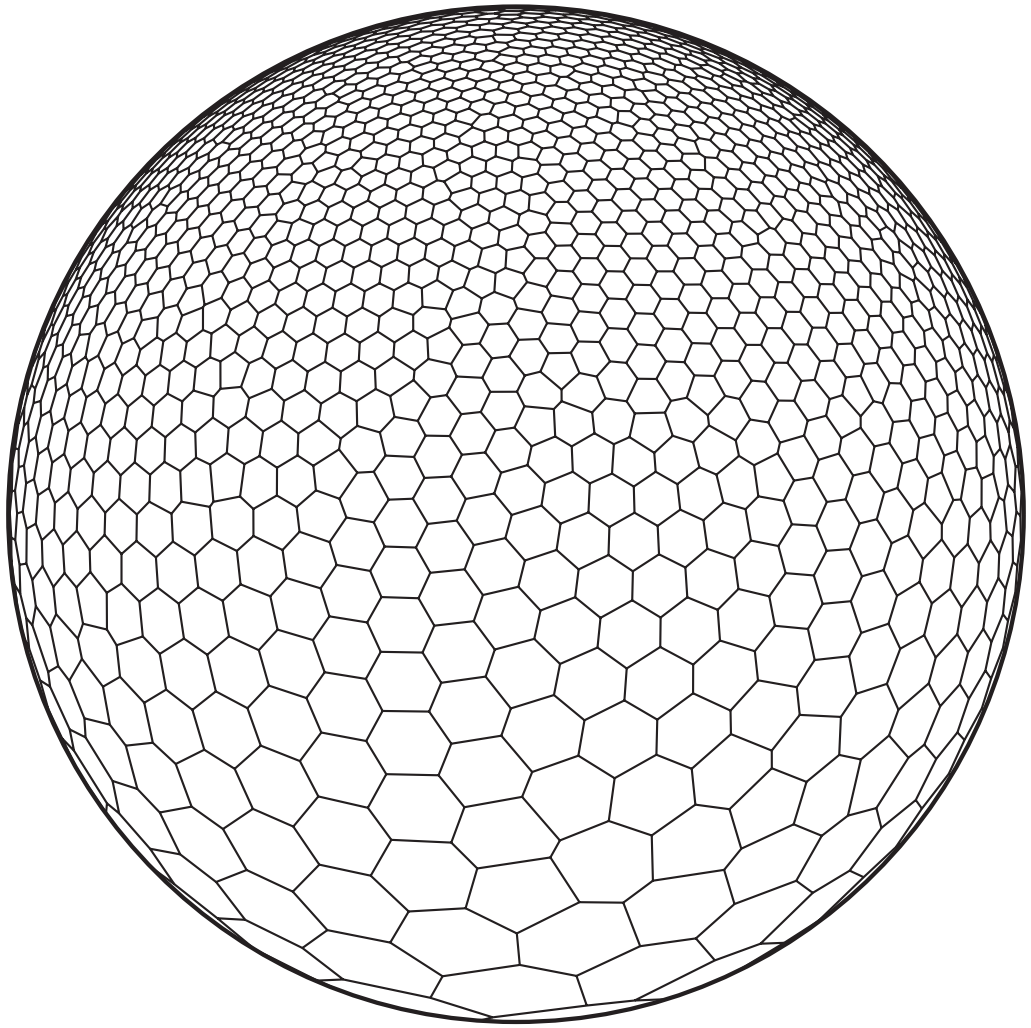
Voronoi diagrams for 256 generators on the sphere

random sampling (top) constrained CVT (bottom)

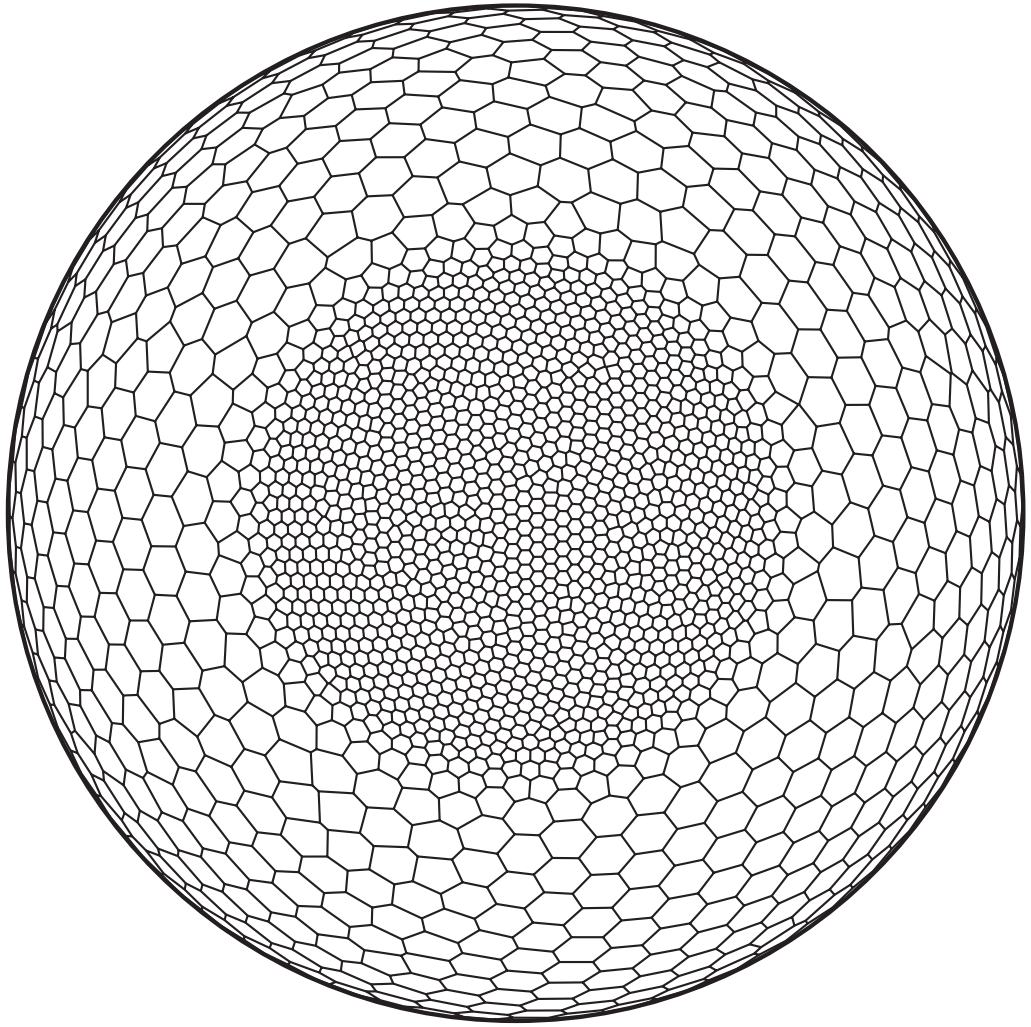
$\rho(x, y, z) =$ 1 (left) $e^{-6.0z^2}$ (middle) $e^{-3.0(1-z)^2}$ (right)



CVT's on the sphere with uniform density



CVT's on the sphere with nonuniform density



CVT's on the sphere with local refinement

HYPERCUBE POINT SAMPLING

- The uniform sampling of points in hypercubes is a task that arises in many settings

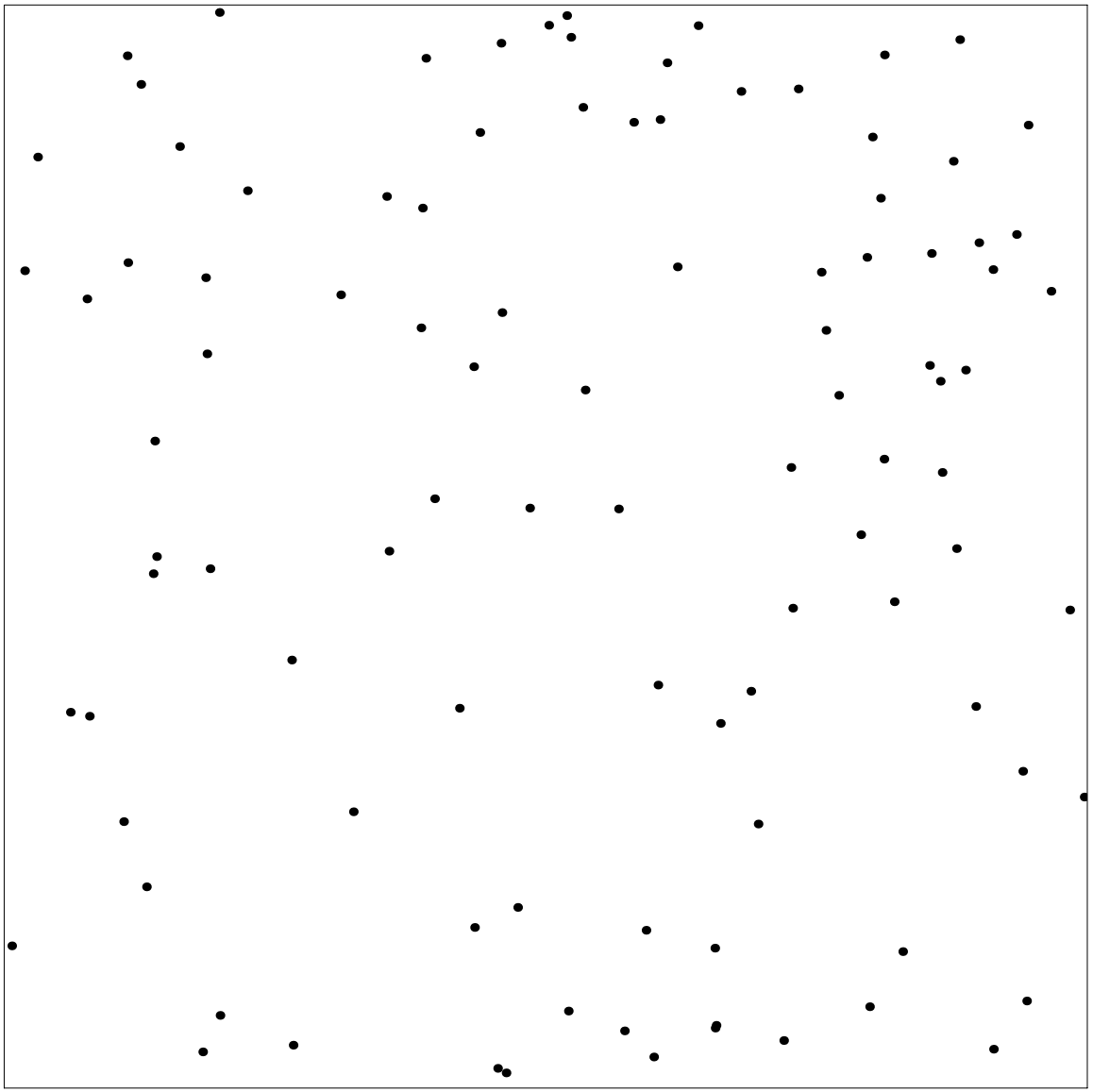
- There have been many (!) methods proposed for the uniform sampling of points in hypercubes
  - Random (Monte Carlo) sampling
  - quasi-random sequences, e.g., Halton, Hammersley, Sobol, etc.
  - Latin hypercube methods and several variants

- We have already seen that CVT point placement provides very good uniform point distibutions
  - so, we want to study more closely the use of CVT for uniform sampling in hypercubes

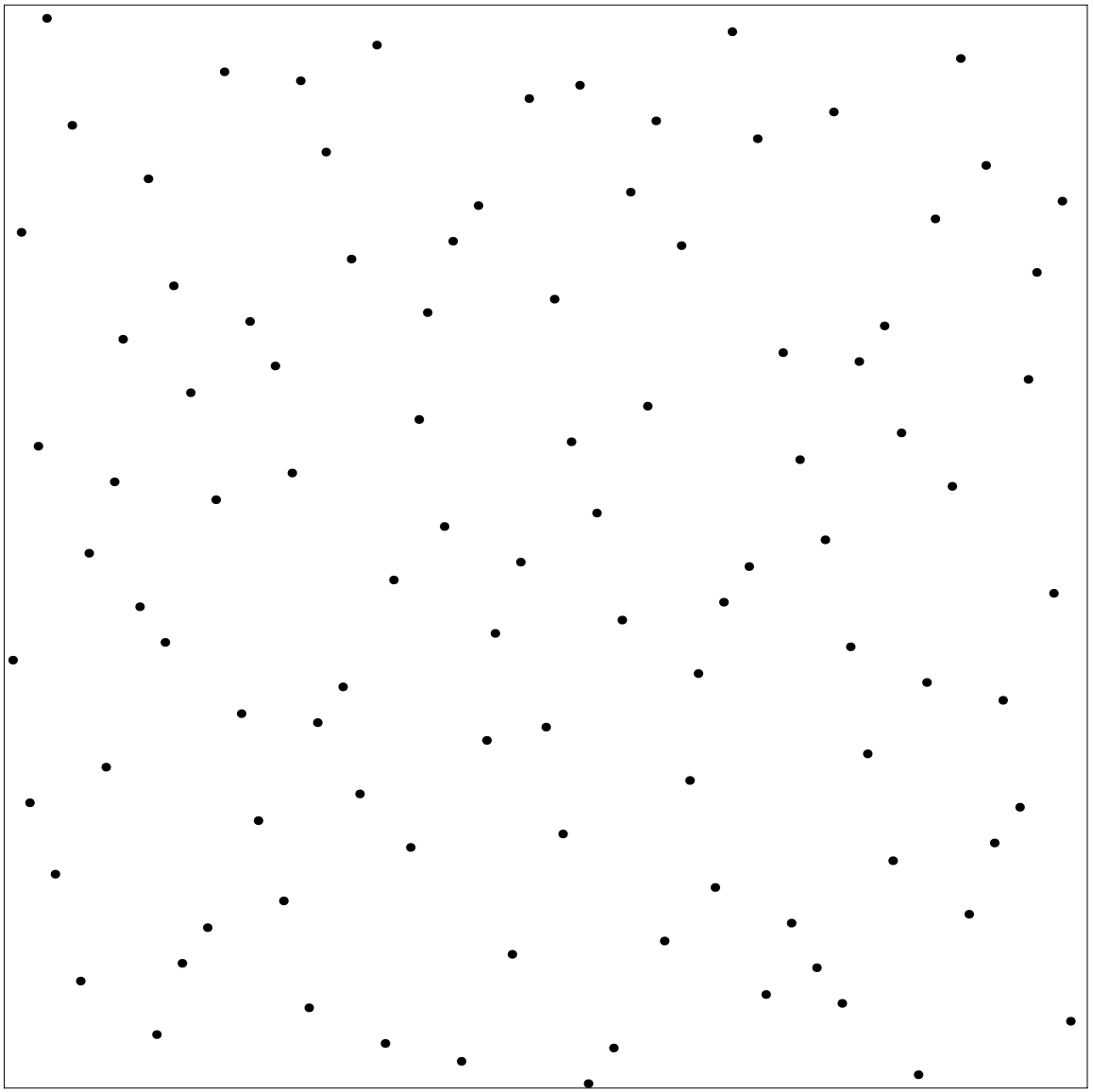
- We compare CVT point sampling with that for some well-known methods
  - visual comparisons in 2D
  - quantative comparions in 2D and 7D

- Of course, CVT can do a lot more than just merely uniform point sampling in hypercubes
  - sampling in general regions
  - nonuniform and anisotropic sampling

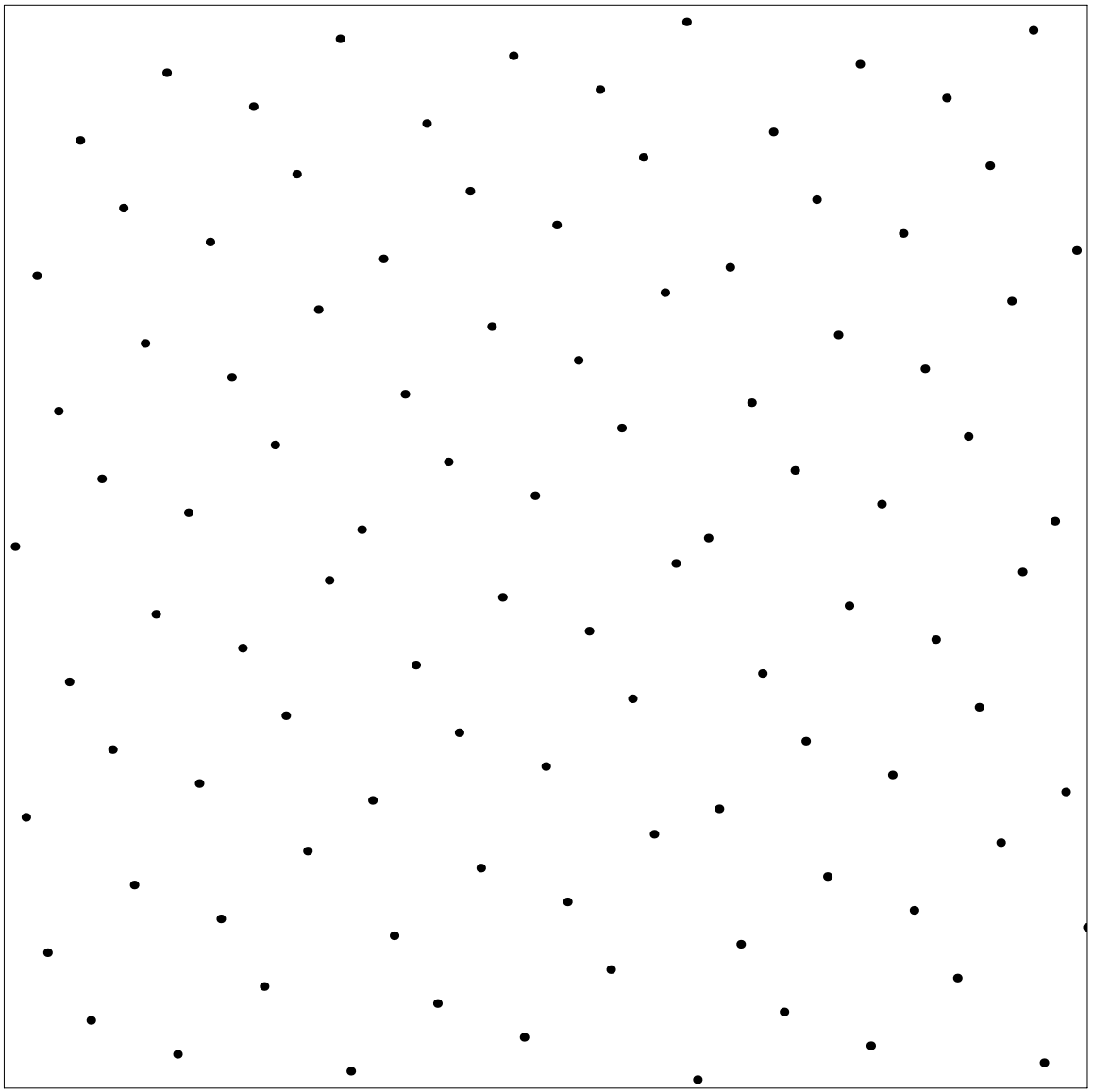
- We also provide some visual comparisons for nonuiform point samples in 2D
  - of course, not all hypercube sampling methods can produce nonuniform point sets



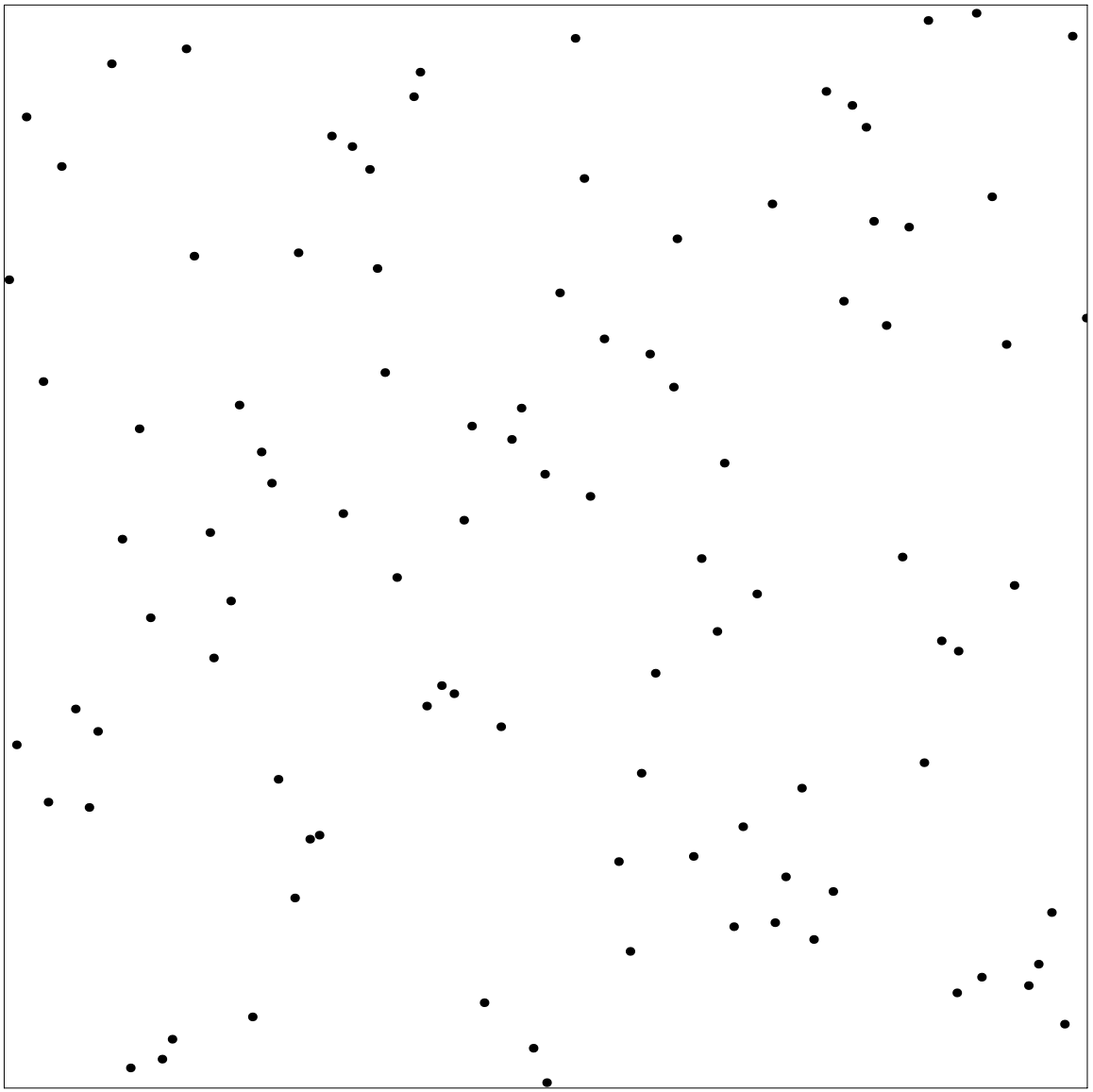
100 uniform Monte Carlo points



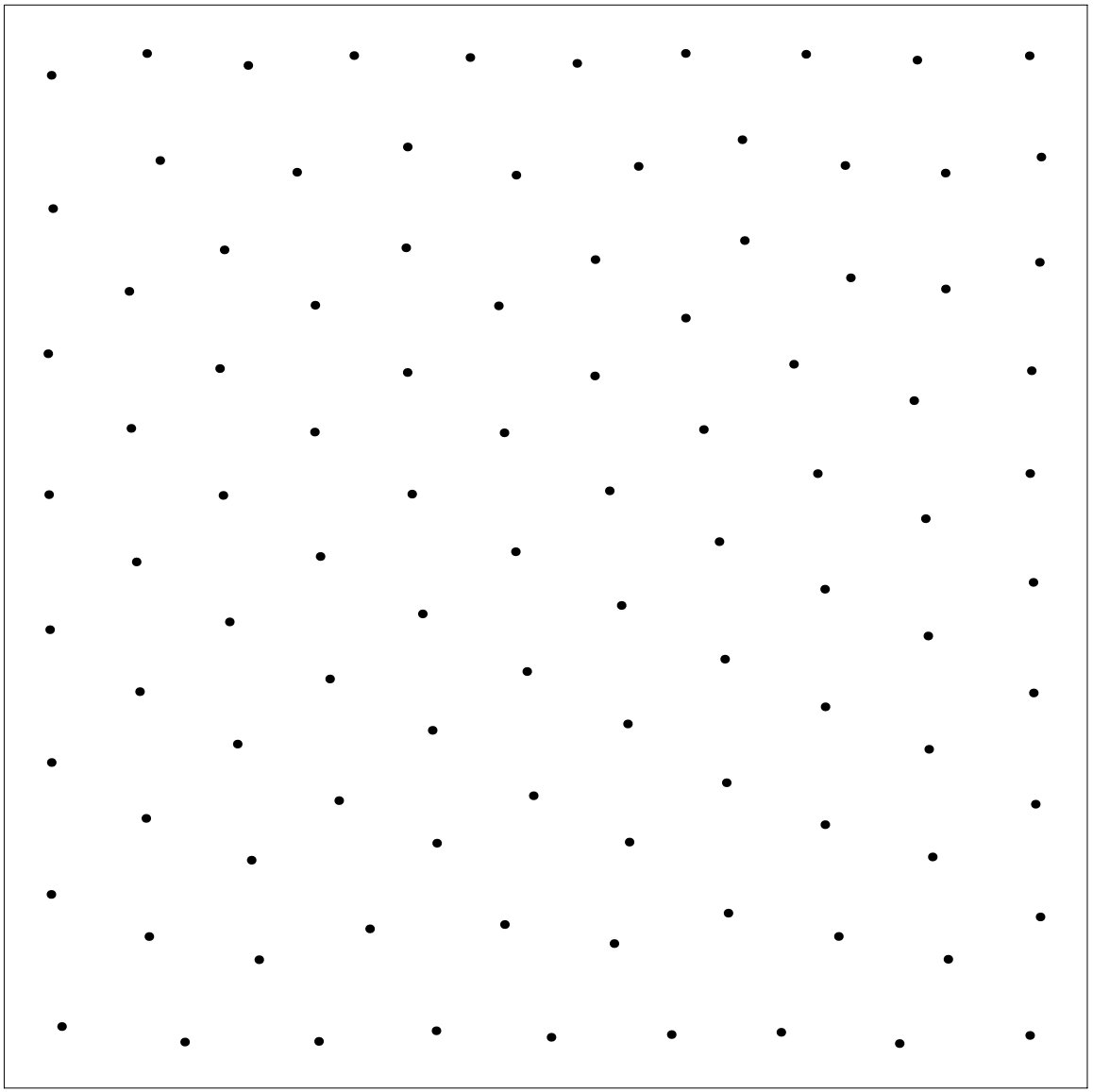
100 uniform Halton points



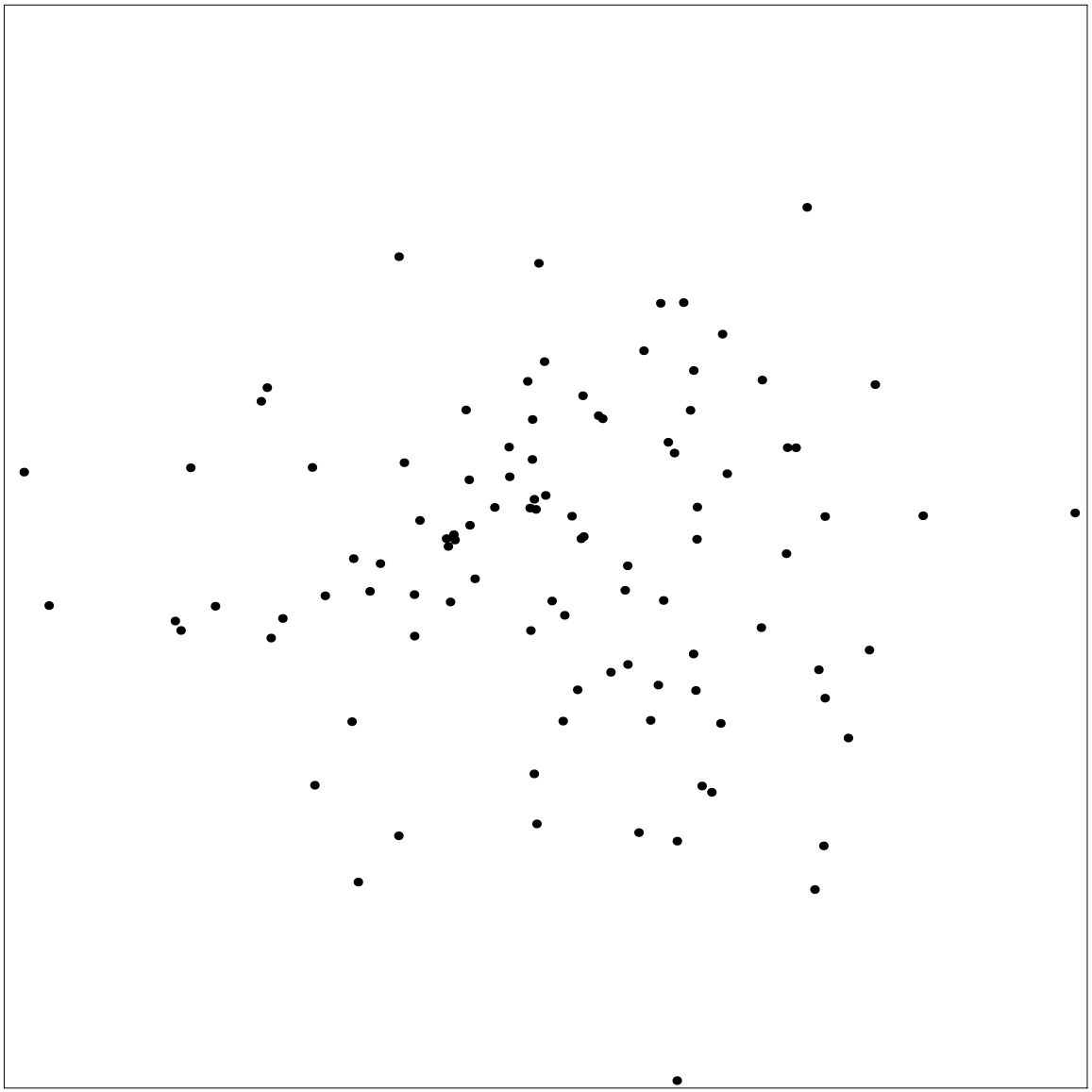
100 uniform Hammersley points



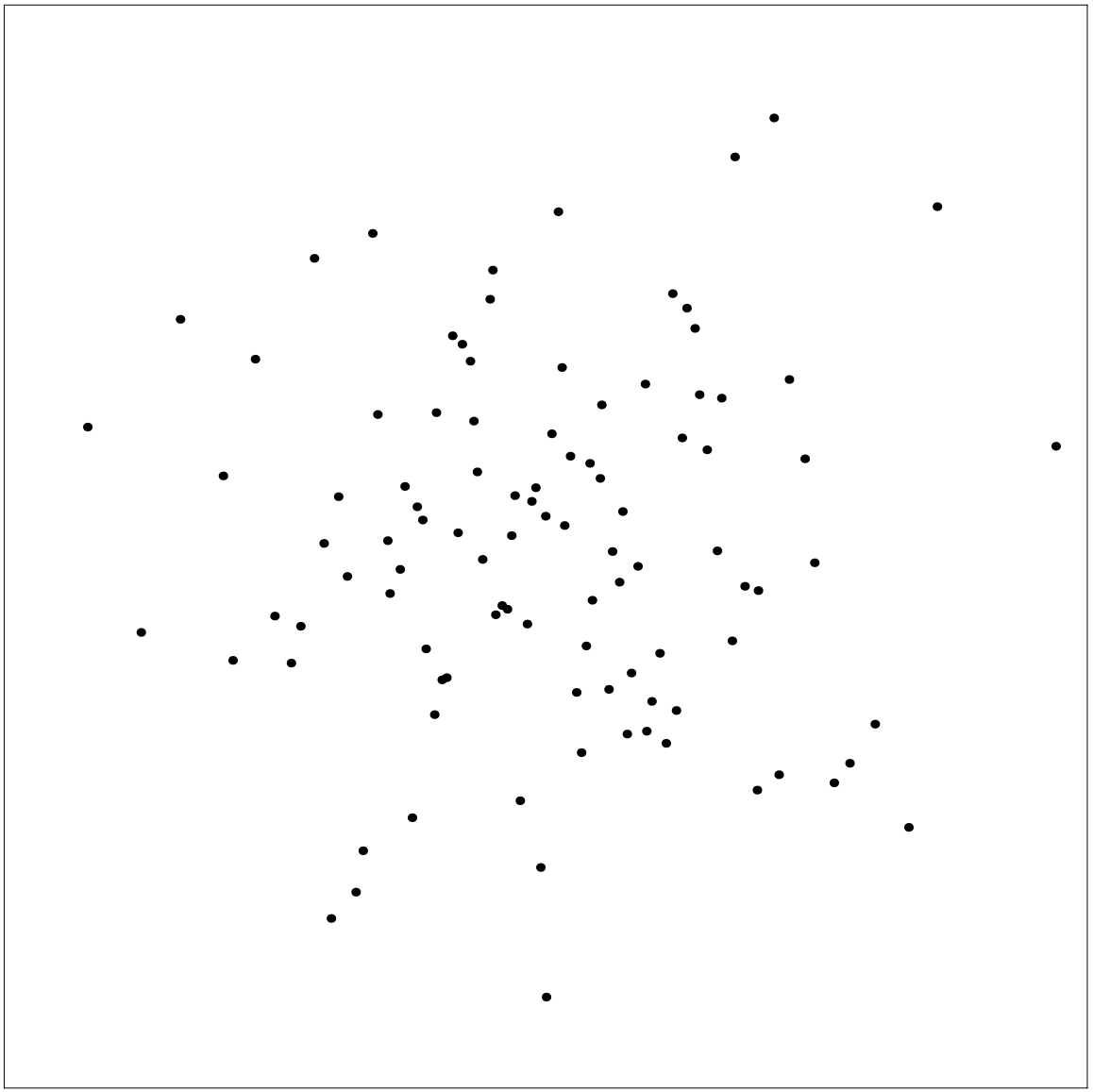
100 uniform Latin hypercube points



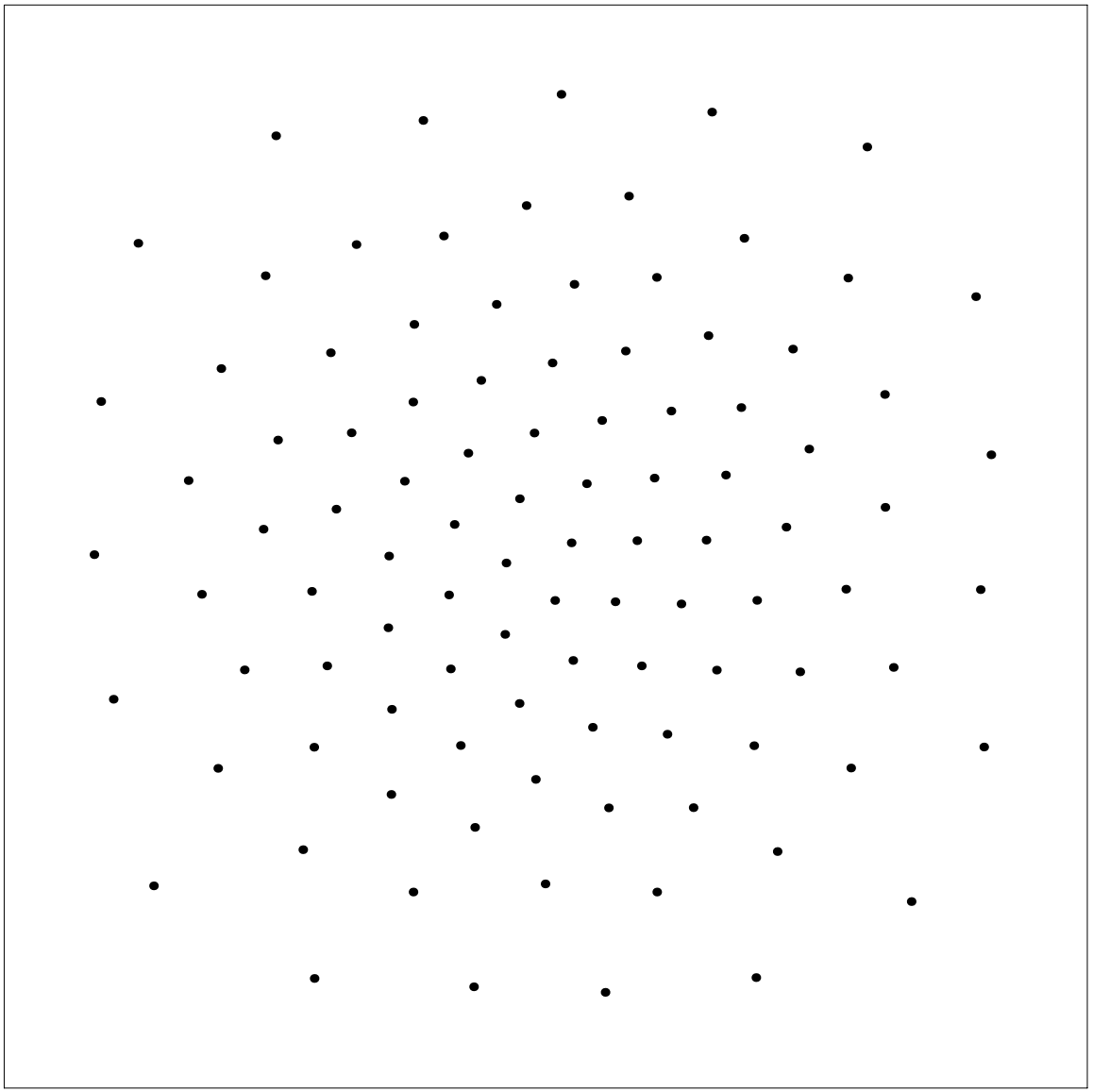
100 uniform CVT points



100 Gaussian Monte Carlo points



100 Gaussian Latin hypercube points



100 Gaussian CVT points

Quantative uniformity measures for points samples

- Uniformly distributed point samples have three attributes
  - the points are equally spaced
  - the points cover the region, i.e., there are no relatively large subregions that contain no points
  - the points are isotropically distributed, i.e., there is no directional bias in the placement of points

- In 2D, the eye does a great job in using all three attributes in comparing the uniformity of different point sets

- Some popular quantative measures of uniformity are flawed in that they only consider the spacing between points

Point-to-point uniformity measures

- These measures take into account the distances between pairs of points

1. COV measure

- Given any set of N points $\{\mathbf{z}_i\}_{i=1}^N$, we compute

$$\gamma_i = \min_{j \neq i} |\mathbf{z}_i - \mathbf{z}_j|$$

so that γ_i is the minimum distance between the point $\mathbf{z}_i$ and any of the other points. Then, the COV measure λ is given by

$$\lambda = \frac{1}{\overline{\gamma}} \left(\frac{1}{N} \sum_{i=1}^{N} (\gamma_i - \overline{\gamma})^2 \right)^{1/2} = \left(N \frac{\sum_{i=1}^N \gamma_i^2}{\left(\sum_{i=1}^N \gamma_i \right)^2} - 1 \right)^{1/2}$$

where

$$\overline{\gamma} = \frac{1}{N} \sum_{i=1}^{N} \gamma_i$$

- For a perfectly uniform mesh, $\gamma_1 = \gamma_2 = \cdots = \gamma_N = \overline{\gamma}$ so that $\lambda = 0$
  - thus, the smaller λ is, the more uniform is the mesh.

2. The mesh ratio

- Given any set of N points $\{\mathbf{z}_i\}_{i=1}^N$, the mesh ratio γ is given by

$$\gamma = \frac{\max_{i=1,\ldots,N} \gamma_i}{\min_{i=1,\ldots,N} \gamma_i}$$

- For a perfectly uniform mesh, $\gamma_1 = \gamma_2 = \cdots = \gamma_N$ so that $\gamma = 1$
  - thus, the smaller γ is, i.e., the closer it is to unity, the more uniform is the mesh.

Volumetric uniformity measures

- Given any set of N points $\{\mathbf{z}_i\}_{i=1}^N$ in a region Ω, we can use those points to generate a Voronoi tessellation $\{V_i\}_{i=1}^N$ of Ω

- Then, we can associate with each point a corresponding Voronoi region

- Then, we can determine various quantities associated with the points and the regions that can be used to measure the quality of the set of points

3. The point distribution norm

- Given a Voronoi tessellation $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^{N}$, the point distribution norm h is given by

$$h = \max_{i=1,\ldots,N} h, \qquad \text{where} \qquad h_i = \max_{\mathbf{y} \in V_i} |\mathbf{z}_i - \mathbf{y}|$$

- Thus, h_i gives the maximum distance between the particular generator $\mathbf{z}_i$ and the points in its associated cell V_i and h gives the maximum distance between any generator and the points in its associated cell

- The point distribution norm h can be used as a measure of the uniformity of point distribution, i.e., of how "close" a point distribution is to an ideal uniform point distribution
  - the smaller the value of h the more uniform is the point distribution

4. The point distribution ratio

- Given a Voronoi tessellation $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^N$, the point distribution ratio μ is given by

$$\mu = \frac{\max_{i=1,\ldots,N} h_i}{\min_{i=1,\ldots,N} h_i}, \qquad \text{where} \qquad h_i = \max_{\mathbf{y} \in V_i} |\mathbf{z}_i - \mathbf{y}|$$

- For an ideal uniform point distribution, $\mu = 1$ so that the smaller μ is
  - the closer it is to unity, the more uniform the point distribution

5. The regularity measure

- Given a Voronoi mesh $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^N$, we define the regularity measure χ by

$$\chi = \max_{i=1,\ldots,N} \chi_i, \qquad \text{where} \qquad \chi_i = \frac{2h_i}{\gamma_i}$$

- For an ideal uniform mesh, $\chi = \chi_i$ for all i; any deviation from uniformity will increase the value of χ

- Thus, χ can be used as another measure of the uniformity of a mesh
  - the smaller the value of χ, the more uniform is the mesh
- In addition, the value of χ provides us a measure of the mesh regularity, i.e., the local uniformity of a mesh
  - if a mesh is locally uniform in the sense that the cells in a neighborhood of any cell are nearly congruent to that cell, then the value of χ will again be small

6. Cell volume deviation

- Given a Voronoi mesh $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^N$, we define the cell volume deviation ν by

$$\nu = \frac{\max_{i=1,\ldots,N} |V_i|}{\min_{i=1,\ldots,N} |V_i|}$$

  where V_i denotes the volume of the cell V_i

- For a perfectly uniform distribution of N points $\{\mathbf{z}_i\}_{i=1}^N$ in a given region Ω, the corresponding volumes $|V_i|$ would all be equal, i.e., $|V|_1 = |V|_2 = \cdots = |V|_N$ so that $\nu = 1$
  - thus, the smaller is ν, i.e., the closer it is to unity, the better is the uniformity of the point distribution.

7. The second moment trace measure

- Given a Voronoi mesh $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^N$, let T_i denote the trace of the second moment tensor (about the region generator) associated with each Voronoi region V_i

- Let $\overline{T} = \frac{1}{N}\sum_{i=1}^N T_i$ denote the average of the trace over the n regions.

- Then, we define the second moment trace measure τ by

$$\tau = \max_{i=1,\ldots,n} |T_i - \overline{T}|$$

- For a perfectly uniform distribution of N points $\{\mathbf{z}_i\}_{i=1}^N$ in a given region V, we would have $T_1 = T_2 = \cdots = T_N = \overline{T}$ so that $\tau = 0$.

  – thus, the smaller τ is the better is the uniformity of the point distribution

8. The second moment determinant measure

- Given a Voronoi mesh $\mathcal{V} = \{\mathbf{z}_i, V_i\}_{i=1}^N$, let D_i denote the determinant of the deviatoric tensor associated with each Voronoi region V_i

- Then, the second moment determinant measure d

$$d = \max_{i=1,\ldots,n} |D_i|$$

- For a perfectly uniform distribution of N points $\{\mathbf{z}_i\}_{i=1}^N$ in a given region V, we would have $D_1 = D_2 = \cdots = D_N = 0$ so that $\tau = 0$

  – thus, the smaller τ is the better is the uniformity of the point distribution.

	COV	γ	h	μ	χ	ν	τ	d
Ideal	0	1	0.0707	1	1.414	1	0	0
Monte Carlo	0.5075	88.75	0.1767	3.792	17.17	25.11	0.2833	0.01246
Halton	0.2911	3.37	0.1266	2.325	5.213	5.386	0.1732	0.01057
Hammersley	0.1559	2.94	0.1424	2.003	4.084	2.744	0.1257	0.00457
Faure	0.2552	2.70	0.1472	2.245	4.640	3.081	0.1432	0.00926
Sobol	0.5246	12.55	0.1378	2.068	20.16	3.369	0.1453	0.01865
Niederreiter	0.3072	3.05	0.1279	1.879	5.172	3.026	0.1294	0.01630
Latin hypercube	0.4771	8.60	0.1690	3.382	13.11	10.04	0.2907	0.02130
IHS	0.1588	2.46	0.1225	2.115	5.551	3.103	0.1033	0.00549
Hanson	0.0484	1.29	0.1074	1.706	2.095	2.136	0.0601	0.00458
CVT	0.0509	1.30	0.0792	1.311	1.720	1.456	0.0355	0.00107

Eight measures of uniformity for different types of 100 "uniformly" distributed point in the unit square

Red – flawed measures of sample uniformity
Green – good measures of sample uniformity

	h	χ	τ	d
Monte Carlo	0.100874D+01	0.608636D+01	0.520662D-01	0.195518D-12
Halton	0.923242D+00	0.458080D+01	0.295432D-01	0.686130D-13
Hammersley	0.920083D+00	0.481531D+01	0.277011D-01	0.441584D-13
Latin hypercube	0.900931D+00	0.652482D+01	0.279853D-01	0.773161D-13
CVT	0.719571D+00	0.277510D+01	0.725527D-02	0.194127D-17

Four "good" measures of uniformity for different types of 100 "uniformly" distributed point in the 7D square

REDUCED-ORDER MODELING

- Solutions of (nonlinear) complex systems are expensive with respect to both storage and CPU costs

- As a result, it is difficult if not impossible to deal with a number of situations such as
  - continuation or homotopy methods for computing state solutions
  - parametric studies of state solutions
  - optimization and control problems (multiple state solutions)
  - feedback control settings (real-time state solutions)

- Not surprinsingly, a lot of attention has been paid to reducing the costs of the nonlinear state solutions by using reduced-order models for the state
  - these are low-dimensional approximations to the state

- Reduced-order modeling has been and remains a very active research direction in many seemingly disparate fields, e.g., to name five:
  - linear algebra: singular value decomposition (SVD), Hankel SVD
  - statisitics: Karhunen-Loève analysis, clustering
  - information science: representation, interpolation, reconstruction
  - boundary layers: e.g., in a fluids setting, replacing the Navier-Stokes equations with the simpler Prandtl boundary layer equations
  - tubulence modeling: e.g., in a fluids setting, replacing the Navier-Stokes equations by another complex system, e.g., a k-ϵ or LES model, that is "easier" to approximate

- In fact, many of us already do the ultimate in reduced-order modeling
  - any computational approximation to a complex system goverened by partial differential equations constitutes an attempt at reduced-order modeling, i.e., reducing an infinite-dimensional problem to a finite-dimensional one

Reduced-order modeling

- For a state simulation, a reduced-order method would proceed as follows
  - one chooses a reduced basis $\mathbf{u}_i$, $i = 1, \ldots, d$

    d is hopefully very small compared to the usual number of functions used in a finite element approximation or the number of grid points used in a finite difference approximation

  - next, one seeks an approximation $\widetilde{\mathbf{u}}$ to the state of the form

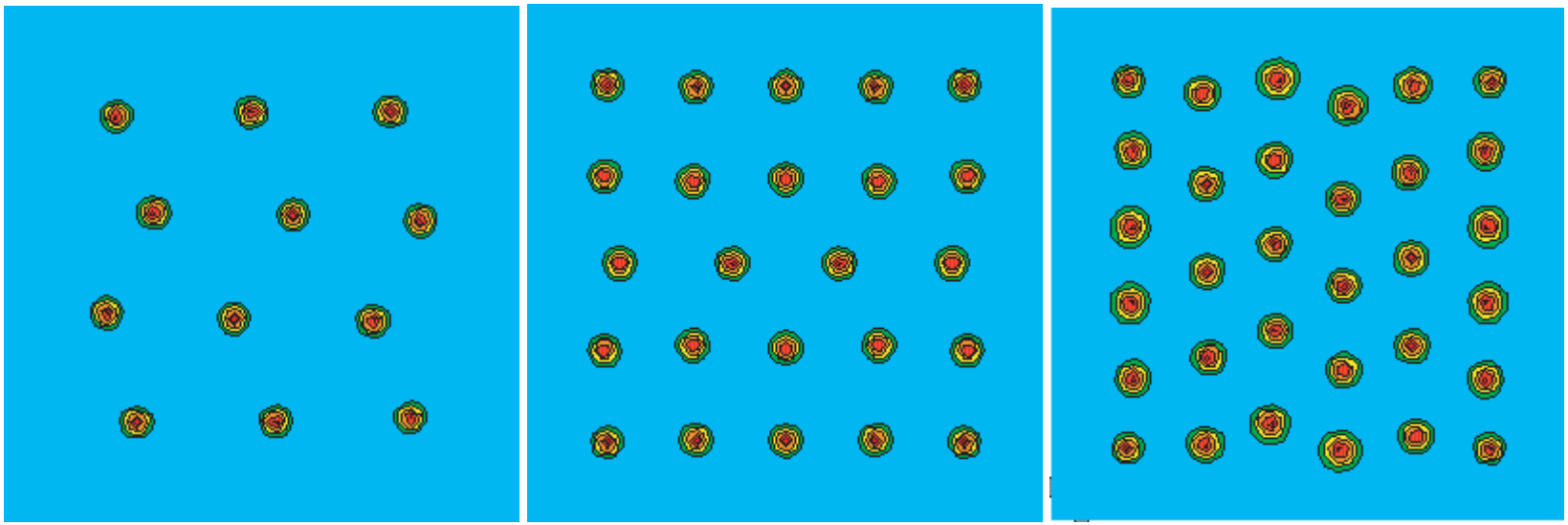
$$\widetilde{\mathbf{u}} = \sum_{i=1}^{d} c_i \mathbf{u}_i \in V \equiv \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_d\}$$

  - then, one determines the coefficients c_i, $i = 1, \ldots, d$, by solving the state equations in the set V

    e.g., one could find a Galerkin solution of the state equations in a standard way, using V for the space of approximations

  - the cost of such a computation would be very small if d is small (ignoring the cost of the off-line determination of the reduced basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_d\}$)

- Does reduced-order modeling work?
  - it is clear that reduced-order methods should work in an interpolatory setting
  - What happens in an extrapolatory setting is not so clear



Superconducting vortices for three different values of the applied magnetic field

Snapshot sets

- The state of a complex system is determined by parameters that appear in the specification of a mathematical model for the system

- Of course, the state of a complex system also depends on the independent variables appearing in the model

- Snapshot sets consist of (expensive computational or, in principle, even experimental) state solutions corresponding to several parameter values and/or evaluated at several values of one or more of the dependent variables
  - steady-state solutions corresponding to several sets of design parameters
  - a time-dependent state solution for a fixed set of design parameter values evaluated at several time instants during the evolution process
  - several state solutions corresponding to different sets of parameter values evaluated at several time instants during the evolution process

Proper orthogonal decomposition (POD) based reduced-order modeling

- Given n snapshots $\widetilde{\mathbf{x}}_j \in \mathbb{R}^N$, $j = 1, \ldots, n$, set

$$\mathbf{x}_j = \widetilde{\mathbf{x}}_j - \widetilde{\boldsymbol{\mu}}\,, \qquad j = 1 \ldots, n, \qquad \text{where} \qquad \widetilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{j=1}^{n} \widetilde{\mathbf{x}}_j$$

  the set $\{\mathbf{x}_j\}_{j=1}^n$ are the modified snapshots

- Let A denote the snapshot matrix , i.e., the $N \times n$ matrix whose columns are the modified snapshots $\mathbf{x}_j$, i.e.,

$$A = \left(\mathbf{x}_1\ ,\ \mathbf{x}_2\ ,\ \ldots\ ,\ \mathbf{x}_n\right) = \left(\widetilde{\mathbf{x}}_1 - \widetilde{\boldsymbol{\mu}}\ ,\ \widetilde{\mathbf{x}}_2 - \widetilde{\boldsymbol{\mu}}\ ,\ \ldots\ ,\ \widetilde{\mathbf{x}}_n - \widetilde{\boldsymbol{\mu}}\right)$$

- The POD basis of dimension d consists of the first d left singular vectors of the snapshot matrix A

- POD is closely related to the statistical methods known as Karhunen-Loève analysis or the method of empirical orthogonal eigenfunctions or principal component analysis

- The POD basis satisfies an optimality property

- POD is the most popular reduced-order modeling technique

- Many variations on POD have been proposed

- Here, we introduce a new, CVT-based approach to reduced-order modeling

CVT's and model reduction

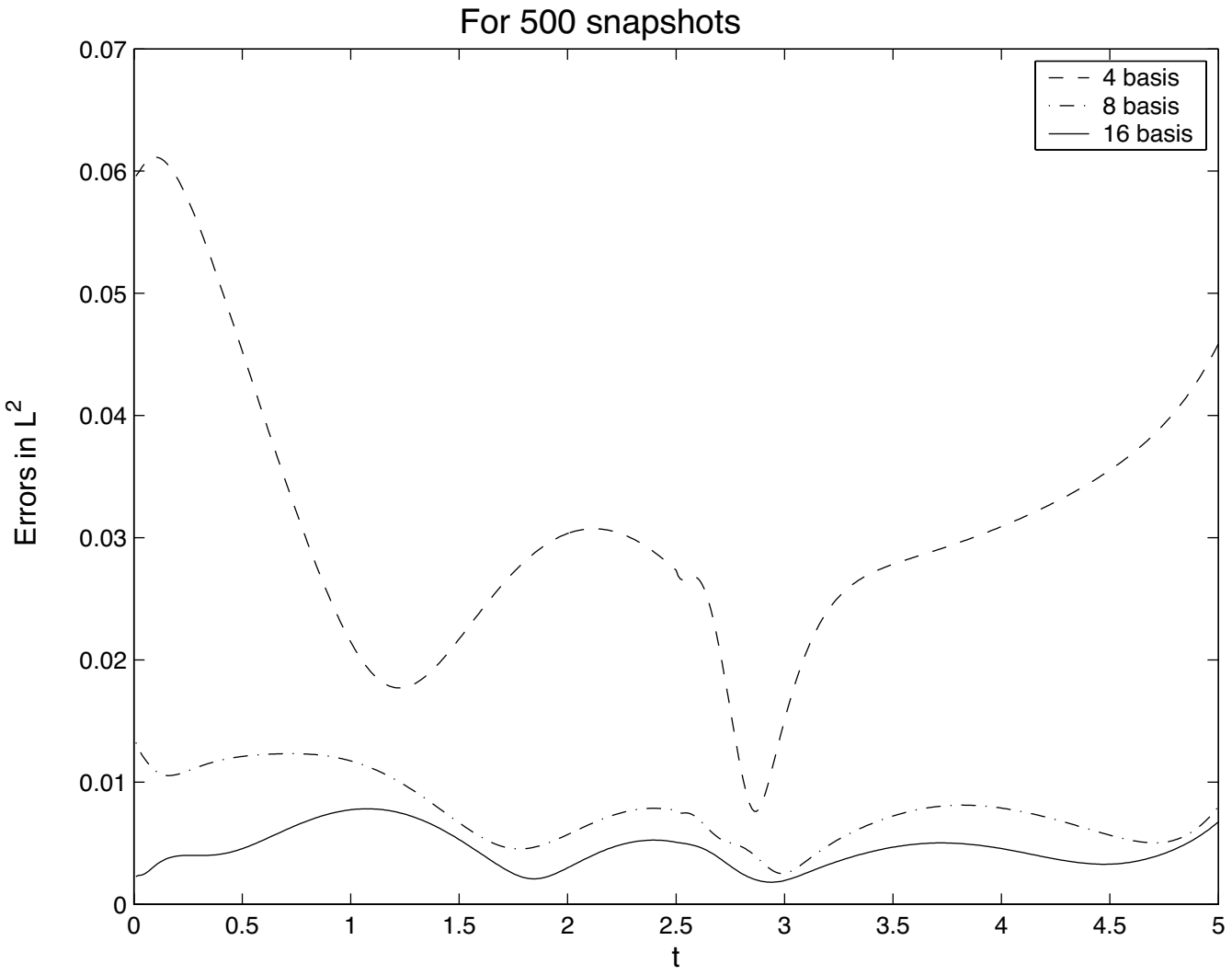
- CVT's have been successfully used in data compression; one particular application was to image reconstruction
  - therefore, it is natural to examine CVT's in another data compression setting, namely reduced-order modeling

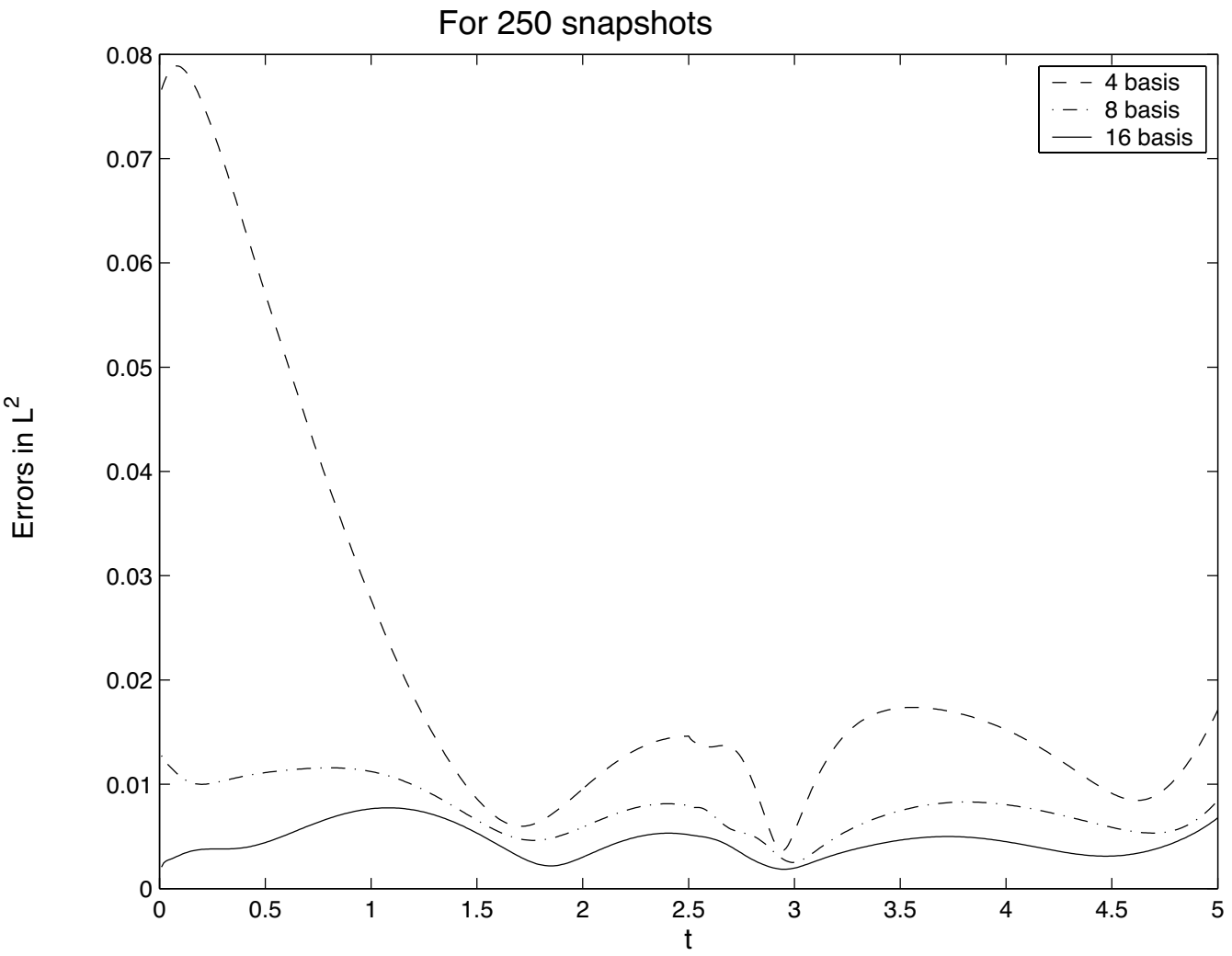
- The idea, just as it is in the POD setting, is to extract, from a given set of (modified) snapshots $\{\mathbf{x}_j\}_{j=1}^n$ of vectors in $\mathbb{R}^N$, a smaller set of vectors also belonging to $\mathbb{R}^N$
  - in the POD setting, the reduced set of vectors was the d-dimensional set of POD vectors $\{\boldsymbol{\phi}_j\}_{i=1}^d$
  - in the CVT setting, the reduced set of vectors is the k-dimensional set of vectors $\{\mathbf{z}_k\}_{i=1}^k$ that are the generators of a centroidal Voronoi tessellation of the set of modified snapshots

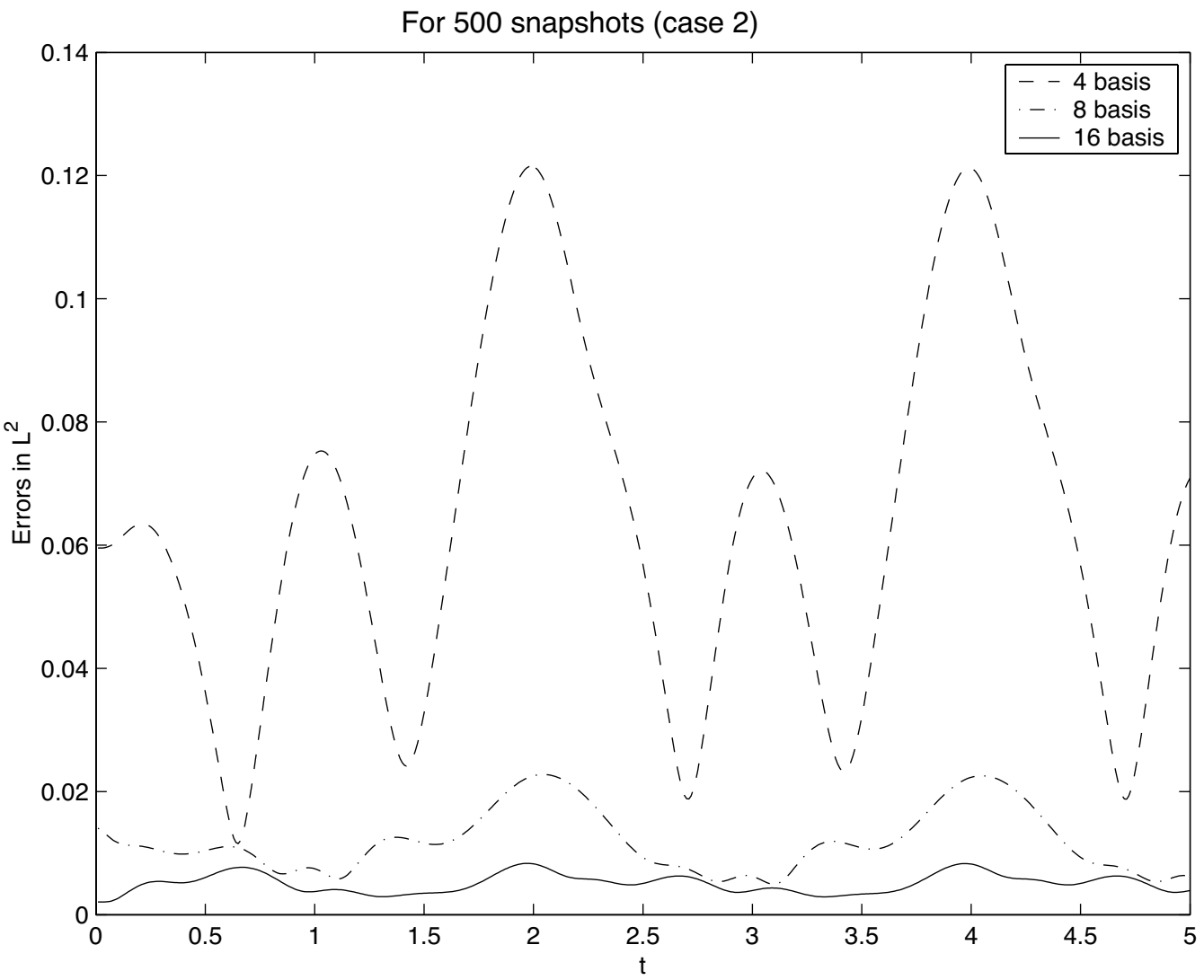
- Just as POD produces an optimal reduced basis in some sense, CVT produces an optimal reduced basis in the sense that the CVT "energy" is minimized

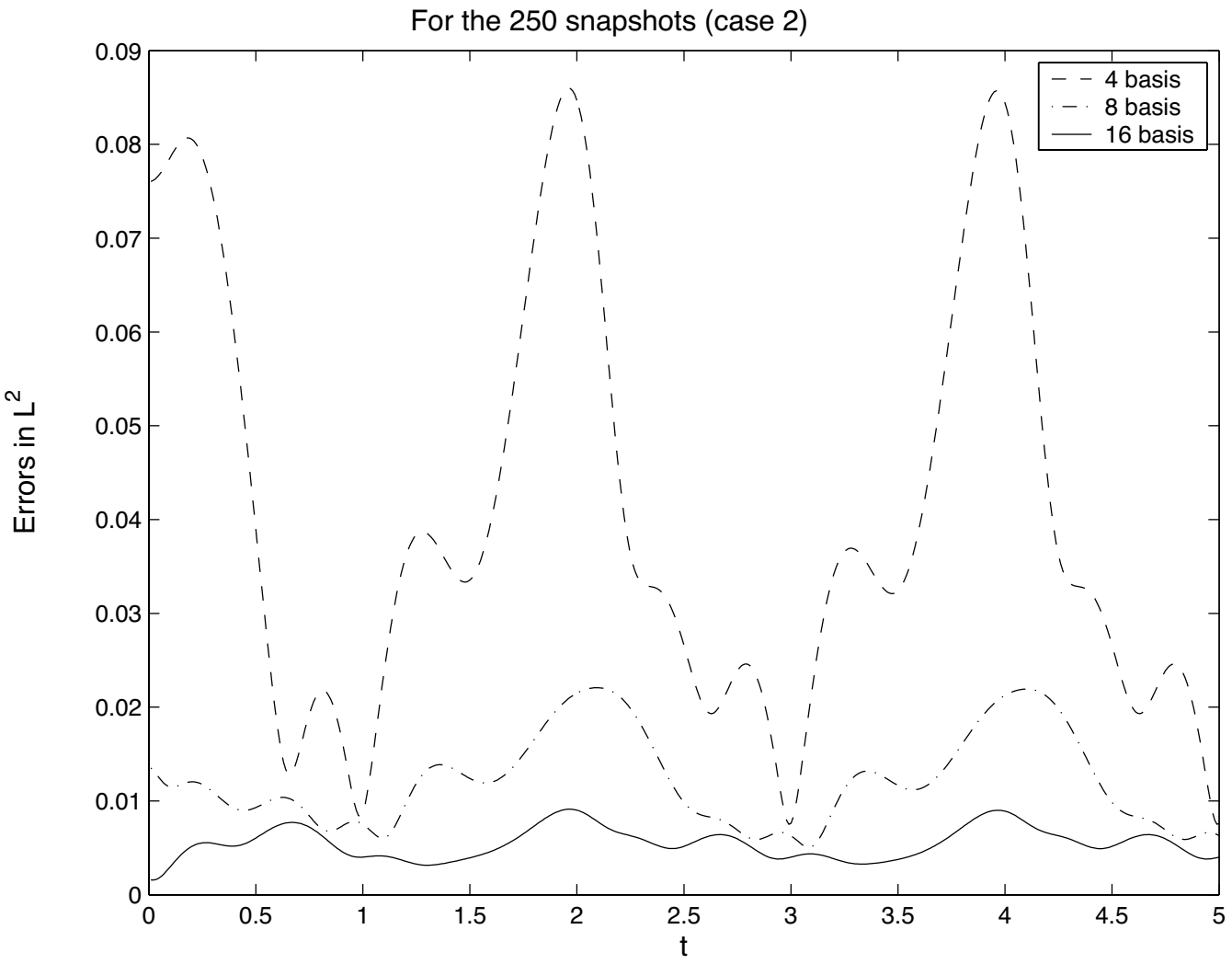
- One can, in principle, determine the dimension d of an effective POD basis, e.g., using the singular values of the snapshot matrix
  - similarly, using the elbowing effect, one can determine the dimension k of an effective CVT basis by examining the (computable) CVT "energy"

Computational results for CVT model reduction




For 250 snapshots
4 basis
8 basis
16 basis
Errors in L2
t
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


For 500 snapshots (case 2)
4 basis
8 basis
16 basis
Errors in L2
t
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


For the 250 snapshots (case 2)
Errors in L2
t
4 basis
8 basis
16 basis
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

CVT vs. POD

- Question: why should one use CVT instead of POD?
  - although justifications have to be substantiated through analyses and extensive further numerical experiments, one can make some arguments

- CVT naturally introduces the concept of clustering into the construction of the reduced basis

- CVT is "cheaper" than POD
  - POD involves the solution of an $n \times n$ eigenproblem, where n is the number of snapshots
  - CVT requires no eigenproblem solution
    - CVT can handle many more snapshots
    - adaptively changing the reduced basis is much less expensive with CVT

- Another interesting feature of CVT is that it avoids the over-crowding of the reduced basis into a few dominant modes

CVT combined with POD (CVOD)

- We have already mentioned that the concept of centroidal Voronoi tessellations can be extended to more general notions of distance
  - this allows us to combine POD and CVT to (hopefully) take advantage of the best features of both approaches.

- Why should one use CVOD instead of POD or CVT?
  - CVOD offers the possibility of taking advantage of the best features of both POD and CVT
  - CVOD is cheaper than POD since it requires the solution of several smaller eigenproblems instead of one large one

CVT and snapshot generation

- CVT can play another important role in reduced-order modeling, regardless of how the reduced-order basis is determined

- Any reduced-order basis cannot be better than the snapshot set from which it is generated
  - if it ain't in the snapshot set, it ain't in the reduced-order basis

- Thus, the intelligent, effective, and efficient generation of snapshot vectors is crucial to the success of reduced-order models
  - very little attention has been paid to this aspect of reduced-order modeling
  - ad-hoc techinques are used to generate snapshots

- In a real sense, snapshot generation is an exercise in the design of experiments
  - CVT point sampling in parameter space can be an effective in this regard

HOMER SIMPSON?