Lloyd-Max Quantization Schemes

Helmut Knaust

Department of Mathematical Sciences
The University of Texas at El Paso
El Paso TX 79968-0514

hknaust@utep.edu

January 6, 2011

Learning Outcomes:

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.
- Students will improve their programming skills.

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.
- Students will improve their programming skills.

Prerequisites:

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.
- Students will improve their programming skills.

Prerequisites:

- Multi-variable Calculus

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.
- Students will improve their programming skills.

Prerequisites:

- Multi-variable Calculus
- Some knowledge of wavelet transforms and their use in image compression

Learning Outcomes:

- Students will reflect on the role of quantization in image compression.
- Students will improve their programming skills.

Prerequisites:

- Multi-variable Calculus
- Some knowledge of wavelet transforms and their use in image compression
- Some minimal statistics knowledge

- Quantization reduces ranges of values in a signal to a single value, thereby reducing entropy.

- Quantization reduces ranges of values in a signal to a single value, thereby reducing entropy.
- Quantization is an integral part of lossy compression algorithms.

- Quantization reduces ranges of values in a signal to a single value, thereby reducing entropy.
- Quantization is an integral part of lossy compression algorithms.
- Quantization is usually employed after transformation.

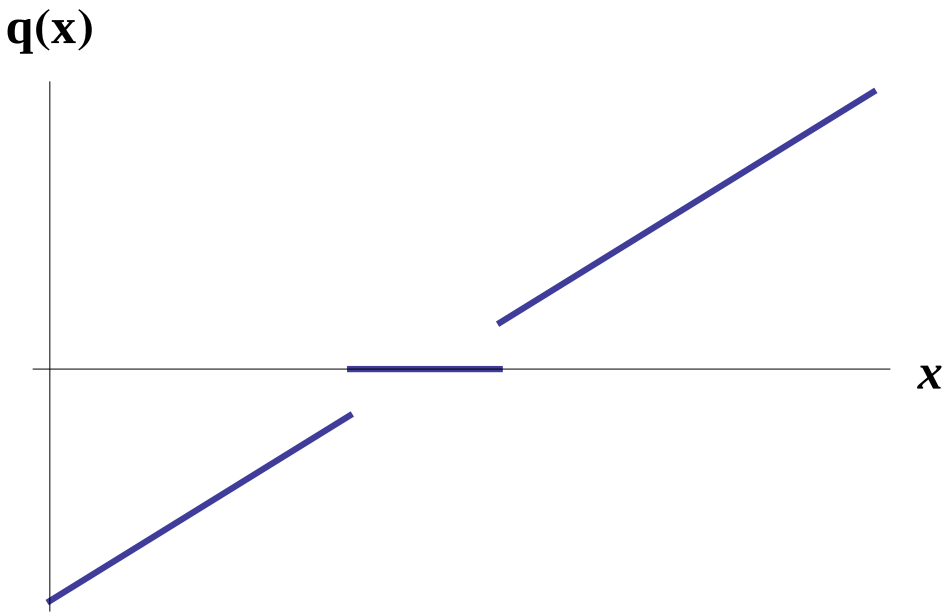
Basic Quantization Scheme: Thresholding

The most basic quantization technique is **Thresholding**:

Given a signal $\vec{x} = (x_i)$ and a single threshold σ, we replace values as follows:

$$q(x_i) = \begin{cases} 0 & \text{if } |x_i| \leq \sigma \\ x_i & \text{if } |x_i| > \sigma \end{cases}$$

The thresholding quantization function:



The **Lossy JPEG2000 Quantization Scheme** also has one fixed parameter, τ. After wavelet transformation, a "step" quantization

$$q(x_i) = sgn(x_i) \cdot \sigma \cdot \left\lfloor \frac{|x_i|}{\sigma} \right\rfloor$$

is applied to each region with a parameter σ determined as follows:

$\frac{\tau}{2}$	τ
τ	2τ

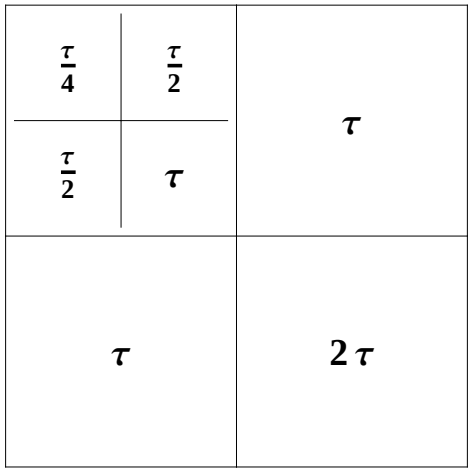
$n = 1$

The JPEG2000 Quantization Scheme

The **Lossy JPEG2000 Quantization Scheme** also has one fixed parameter, τ. After wavelet transformation, a "step" quantization

$$q(x_i) = sgn(x_i) \cdot \sigma \cdot \left\lfloor \frac{|x_i|}{\sigma} \right\rfloor$$

is applied to each region with a parameter σ determined as follows:



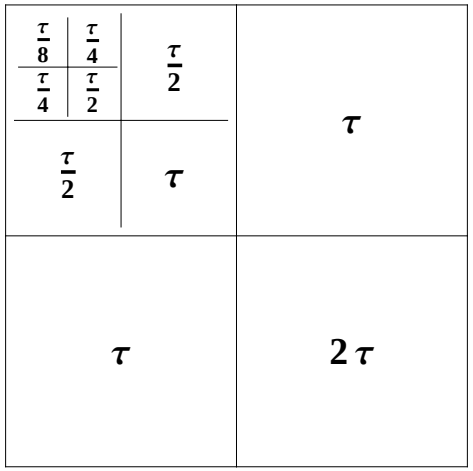
$n = 2$

The JPEG2000 Quantization Scheme

The **Lossy JPEG2000 Quantization Scheme** also has one fixed parameter, τ. After wavelet transformation, a "step" quantization

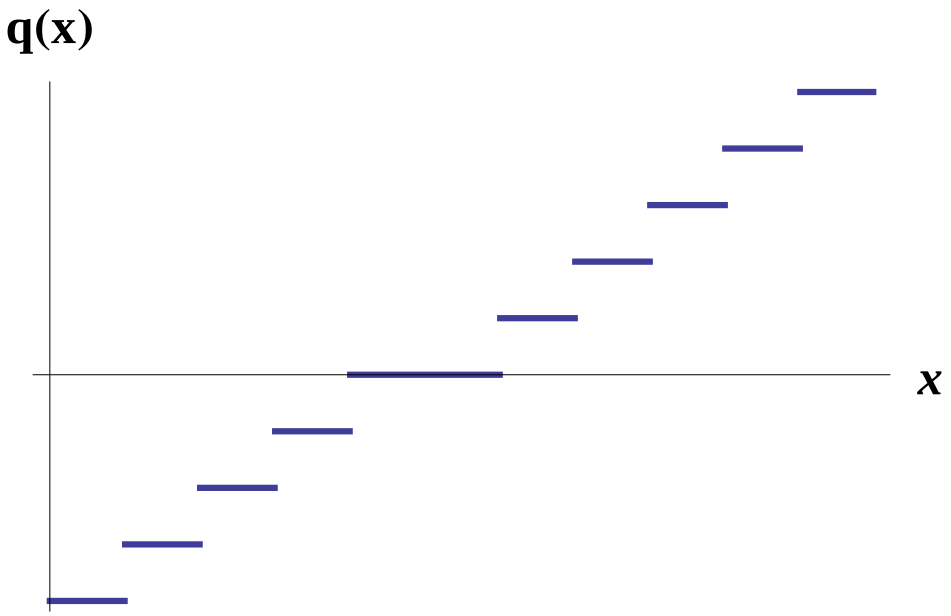
$$q(x_i) = sgn(x_i) \cdot \sigma \cdot \left\lfloor \frac{|x_i|}{\sigma} \right\rfloor$$

is applied to each region with a parameter σ determined as follows:



$n = 3$

The quantization function for a given region:



For a fixed signal $\vec{x}$ and a fixed positive integer n, **Lloyd-Max quantization** uses two sets of parameters:

- Bin-boundaries $\vec{L} = (L_1, L_2, \ldots, L_{n+1})$, with $\min \vec{x} = L_1 < L_2 < \cdots < L_n < L_{n+1} = 1 + \max \vec{x}$,
- and replacement values $\vec{p} = (p_1, p_2, \ldots, p_n)$.

The quantization function replaces the x-values in the bin $[L_j, L_{j+1})$ by the value p_j:

$$q(x_i) = p_j, \text{ when } x_i \in [L_j, L_{j+1})$$

The goal is to choose $\vec{L}$ and $\vec{p}$ to minimize the resulting quantization error

$$E(\vec{L}, \vec{p}) = \sum_{i=1}^{m} |x_i - q(x_i)|^2$$

This is a classical Calculus problem. Rewriting we obtain:

$$E(\vec{L}, \vec{p}) = \sum_{j=1}^{n} \sum_{x_i \in [L_j, L_{j+1})} (x_i - p_j)^2$$

$$E(\vec{L}, \vec{p}) = \sum_{j=1}^{n} \sum_{x_i \in [L_j, L_{j+1})} (x_i - p_j)^2$$

Since minima will occur only if all partial derivatives are equal to 0, the following conditions need to be satisfied:

$$E(\vec{L}, \vec{p}) = \sum_{j=1}^{n} \sum_{x_i \in [L_j, L_{j+1})} (x_i - p_j)^2$$

Since minima will occur only if all partial derivatives are equal to 0, the following conditions need to be satisfied:

$$\frac{\partial E}{\partial p_j} = \sum_{x_i \in [L_j, L_{j+1})} 2(x_i - p_j) = 0 \Leftrightarrow p_j = \frac{\sum_{x_i \in [L_j, L_{j+1})} x_i}{\#\{i \mid x_i \in [L_j, L_{j+1})\}} \quad (1)$$

$$E(\vec{L},\vec{p}) = \sum_{j=1}^{n} \sum_{x_i \in [L_j, L_{j+1})} (x_i - p_j)^2$$

Since minima will occur only if all partial derivatives are equal to 0, the following conditions need to be satisfied:

$$\frac{\partial E}{\partial p_j} = \sum_{x_i \in [L_j, L_{j+1})} 2(x_i - p_j) = 0 \Leftrightarrow p_j = \frac{\sum_{x_i \in [L_j, L_{j+1})} x_i}{\#\{i \mid x_i \in [L_j, L_{j+1})\}} \quad (1)$$

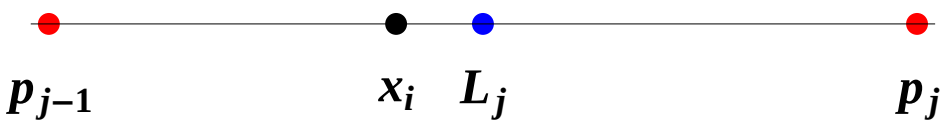
$$\frac{\partial E}{\partial L_j} = 0 \quad \Leftrightarrow \quad L_j = \frac{1}{2}(p_{j-1} + p_j) \quad (2)$$

$$E(\vec{L},\vec{p}) = \sum_{j=1}^{n} \sum_{x_i \in [L_j, L_{j+1})} (x_i - p_j)^2$$

Since minima will occur only if all partial derivatives are equal to 0, the following conditions need to be satisfied:

$$\frac{\partial E}{\partial p_j} = \sum_{x_i \in [L_j, L_{j+1})} 2(x_i - p_j) = 0 \Leftrightarrow p_j = \frac{\sum_{x_i \in [L_j, L_{j+1})} x_i}{\#\{i \mid x_i \in [L_j, L_{j+1})\}} \quad (1)$$

$$\frac{\partial E}{\partial L_j} = 0 \quad \Leftrightarrow \quad L_j = \frac{1}{2}(p_{j-1} + p_j) \quad (2)$$



These equations can usually not be solved explicitly; instead a **two-step procedure** is used repeatedly until a fixed point has been (nearly?) reached:

These equations can usually not be solved explicitly; instead a **two-step procedure** is used repeatedly until a fixed point has been (nearly?) reached:

- Equations (1) are used to update the values $\vec{p}$:

$$p_j^{\text{new}} = ave\ \left\{x_i \mid x_i \in [L_j, L_{j+1})\right\}$$

These equations can usually not be solved explicitly; instead a **two-step procedure** is used repeatedly until a fixed point has been (nearly?) reached:

- Equations (1) are used to update the values $\vec{p}$:

$$p_j^{\text{new}} = ave\ \left\{x_i \mid x_i \in [L_j, L_{j+1})\right\}$$

- Equations (2) then yield new values for $\vec{L}$:

$$L_j^{\text{new}} = \frac{1}{2}(p_{j-1}^{\text{new}} + p_j^{\text{new}})\ , \quad j = 2, \ldots, n$$

The values for L_1 and L_{n+1} are left unchanged.

As an example, we apply the algorithm to a grayscale image, with $n = 32$. The initial bins are chosen at random.

Original image, entropy=7.65

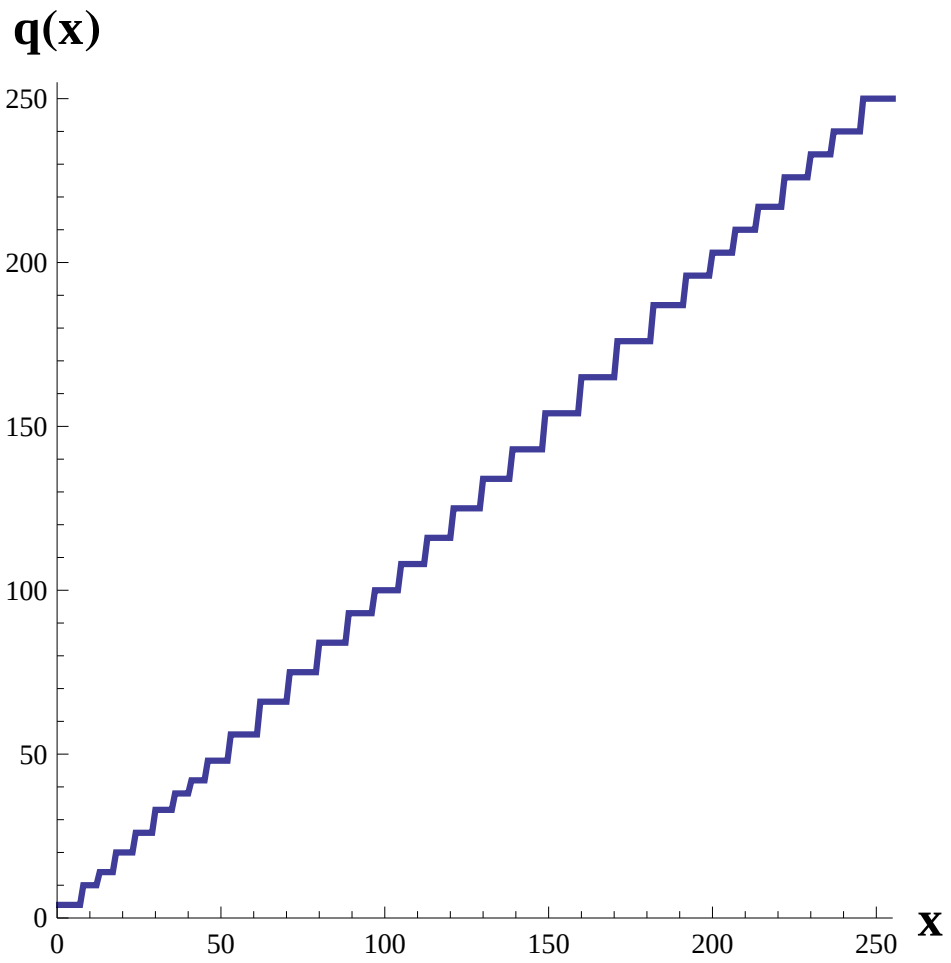
As an example, we apply the algorithm to a grayscale image, with $n = 32$. The initial bins are chosen at random.

21 iterations, entropy=4.75, PSNR 40.7

As an example, we apply the algorithm to a grayscale image, with $n = 32$. The initial bins are chosen at random.

28 iterations, entropy=4.56, PSNR 40.0

Here is the quantization function for the second run:



Quantization applied to "raw" images usually gives bad results in areas of gradual gray-value change (sky, water, etc.):

Original image:
entropy=7.63

Quantization applied to "raw" images usually gives bad results in areas of gradual gray-value change (sky, water, etc.):

$n = 2^6$:
10 iterations, entropy=5.53, PSNR 45.1

Quantization applied to "raw" images usually gives bad results in areas of gradual gray-value change (sky, water, etc.):

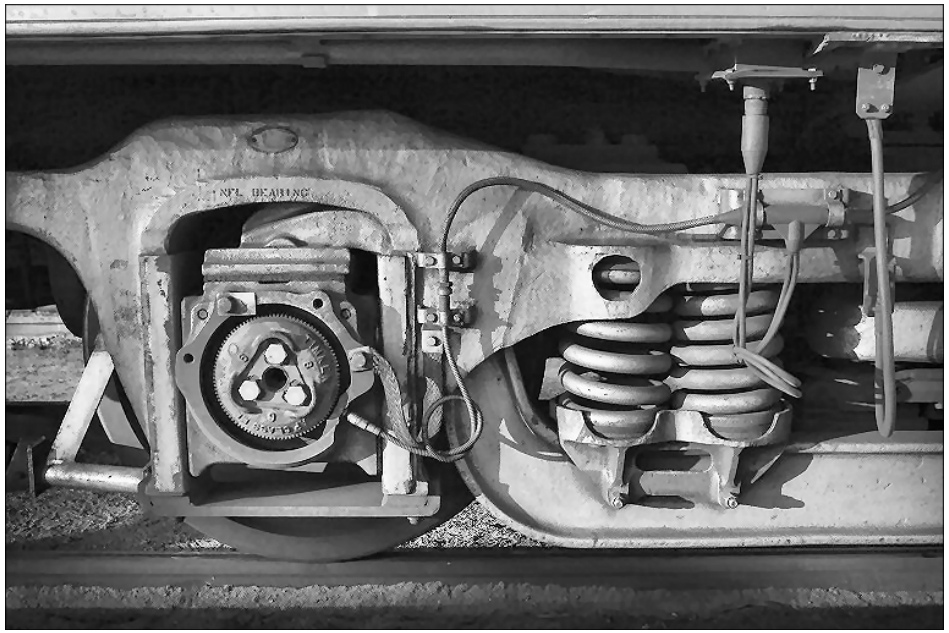
$n = 2^5$:
18 iterations, entropy=4.35, PSNR 38.0

Quantization applied to "raw" images usually gives bad results in areas of gradual gray-value change (sky, water, etc.):

$n = 2^4$:
22 iterations, entropy=3.76, PSNR 34.7

We now compare step quantization to Lloyd-Max quantization for a transformed image:

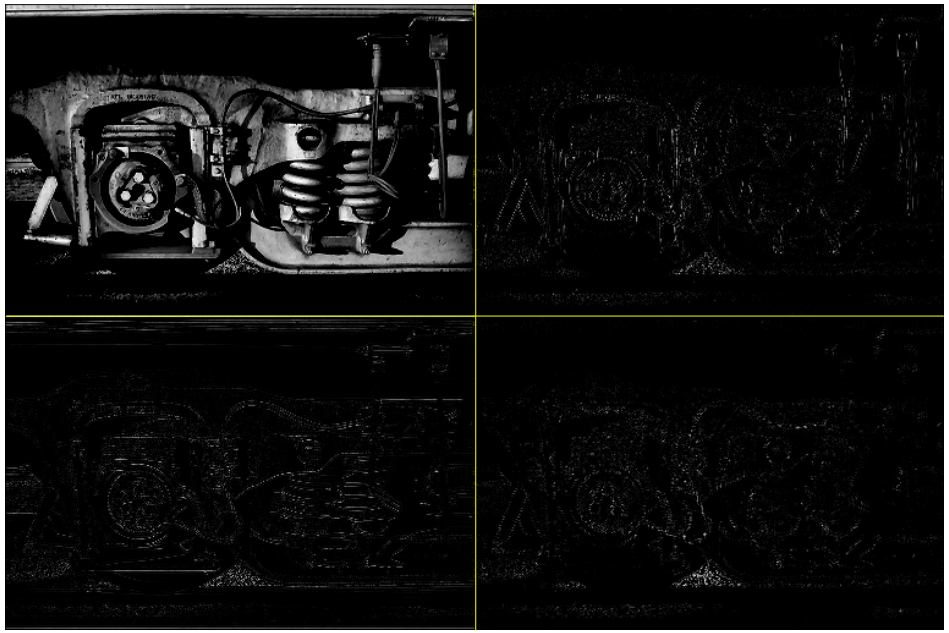
- We use the CDF97 wavelet transform once.
- For the step quantization we use $\tau = 16$:



Original image (-128): entropy=7.74

We now compare step quantization to Lloyd-Max quantization for a transformed image:

- We use the CDF97 wavelet transform once.
- For the step quantization we use $\tau = 16$:



CDF97-transformed image

We now compare step quantization to Lloyd-Max quantization for a transformed image:

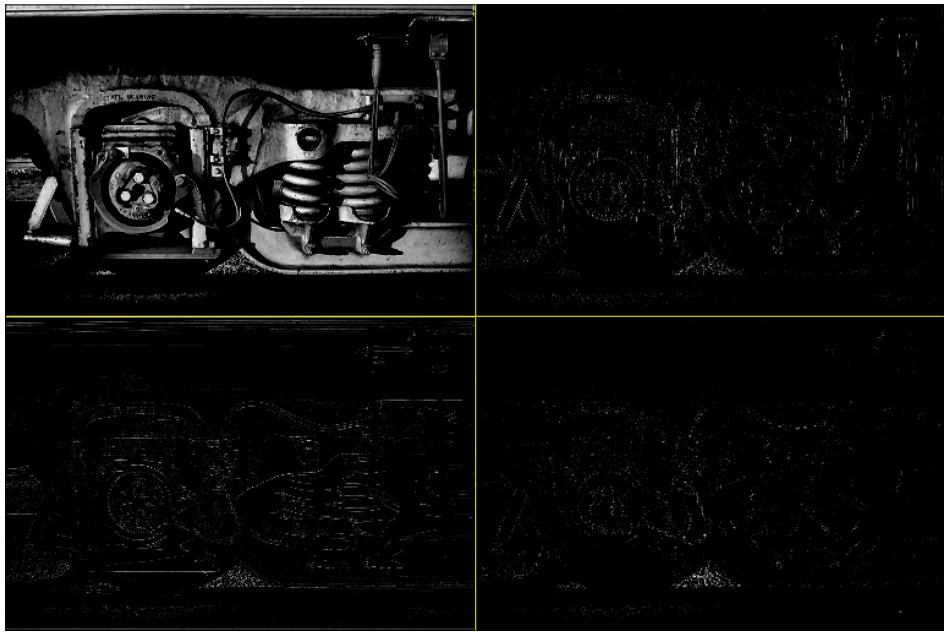
- We use the CDF97 wavelet transform once.
- For the step quantization we use $\tau = 16$:

$\tau/2=8$	$\tau=16$
$\tau=16$	$2\tau=32$

Step sizes for the quantization

We now compare step quantization to Lloyd-Max quantization for a transformed image:

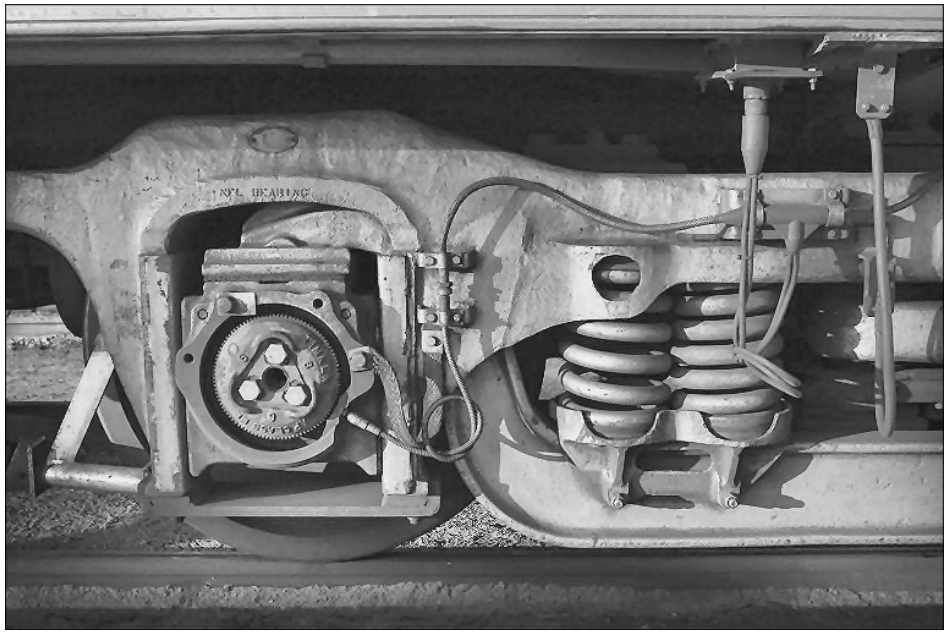
- We use the CDF97 wavelet transform once.
- For the step quantization we use $\tau = 16$:



Quantized transform: entropy=2.75

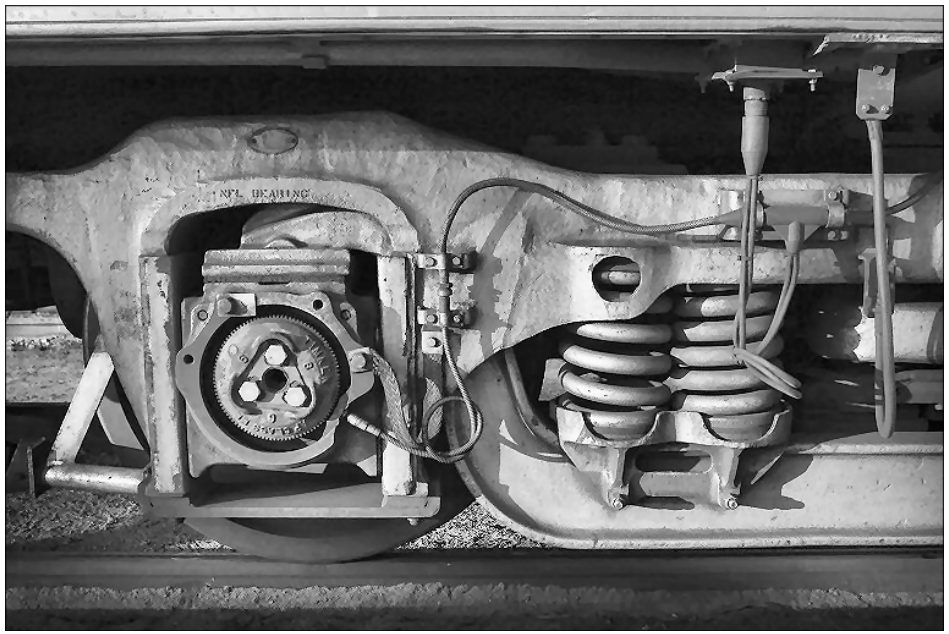
We now compare step quantization to Lloyd-Max quantization for a transformed image:

- We use the CDF97 wavelet transform once.
- For the step quantization we use $\tau = 16$:



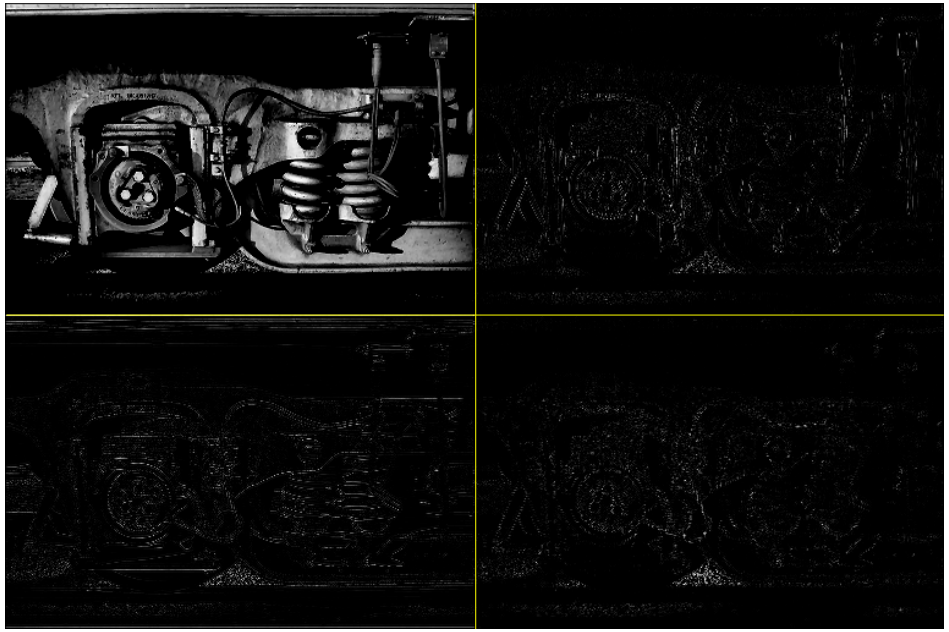
Reconstructed image: PSNR 32.44

Here is the same example with Lloyd-Max quantization:



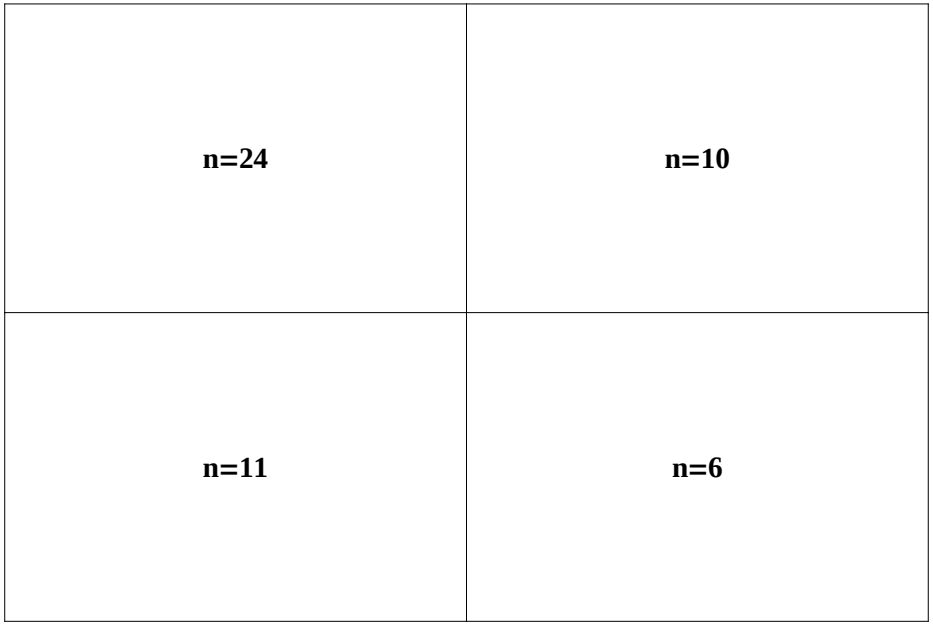
Original image (-128): entropy=7.74

Here is the same example with Lloyd-Max quantization:



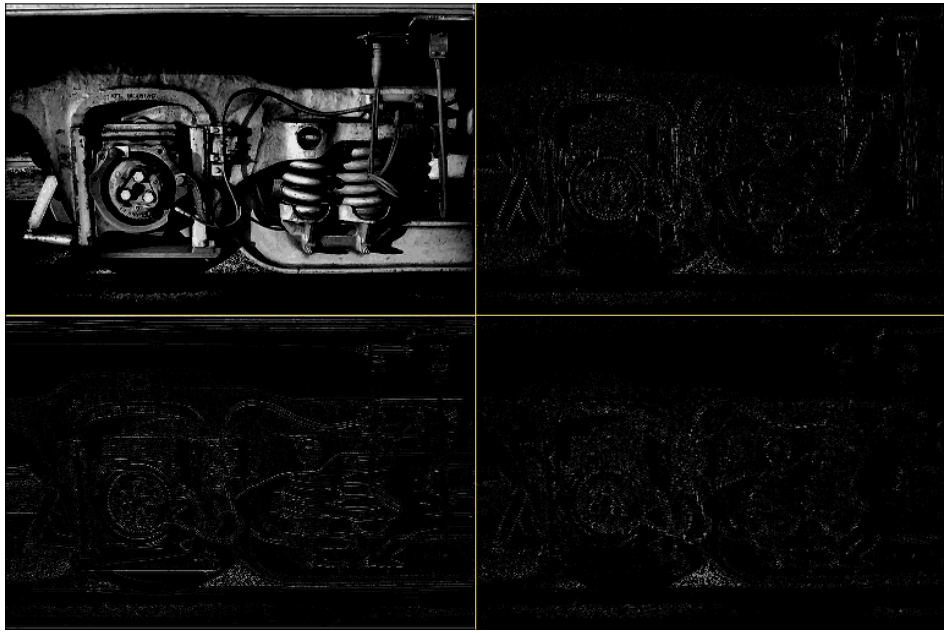
CDF97-transformed image

Here is the same example with Lloyd-Max quantization:



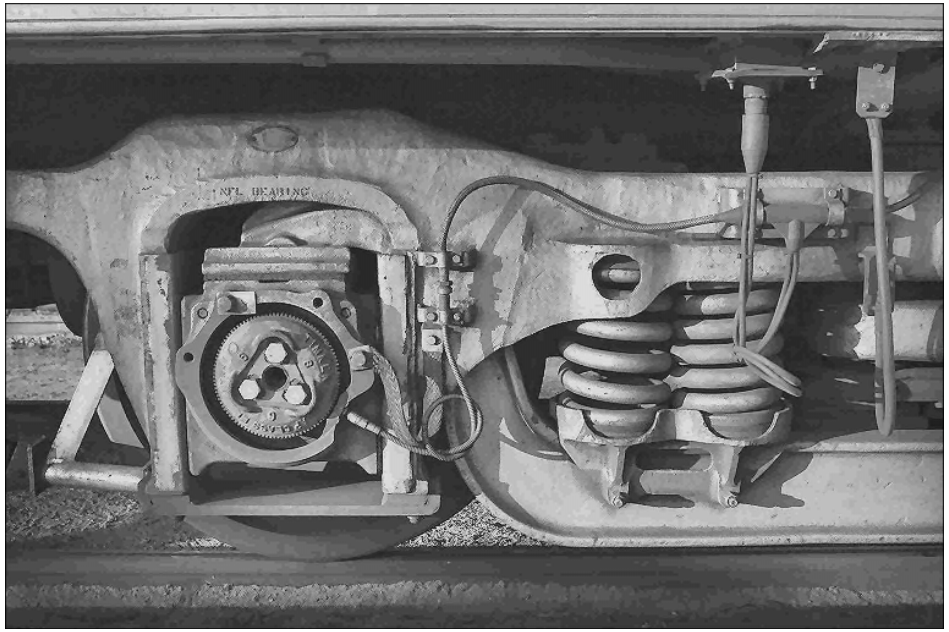
Number of bins for the quantization

Here is the same example with Lloyd-Max quantization:



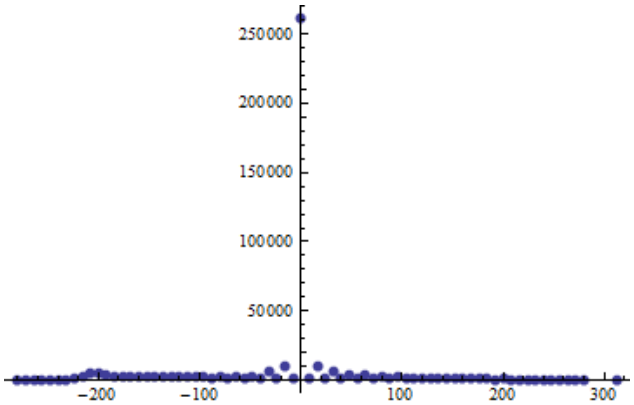
Quantized transform: entropy=3.73

Here is the same example with Lloyd-Max quantization:

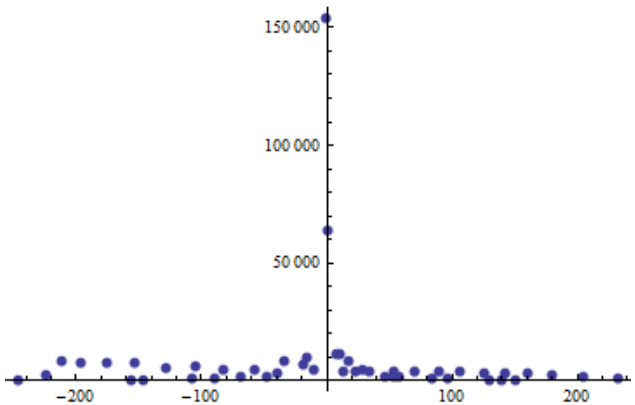


Reconstructed image: PSNR=33.44

Why is the entropy comparatively high after LM-quantization?
Here are the histograms of the quantized transformed signals:



Step quantization

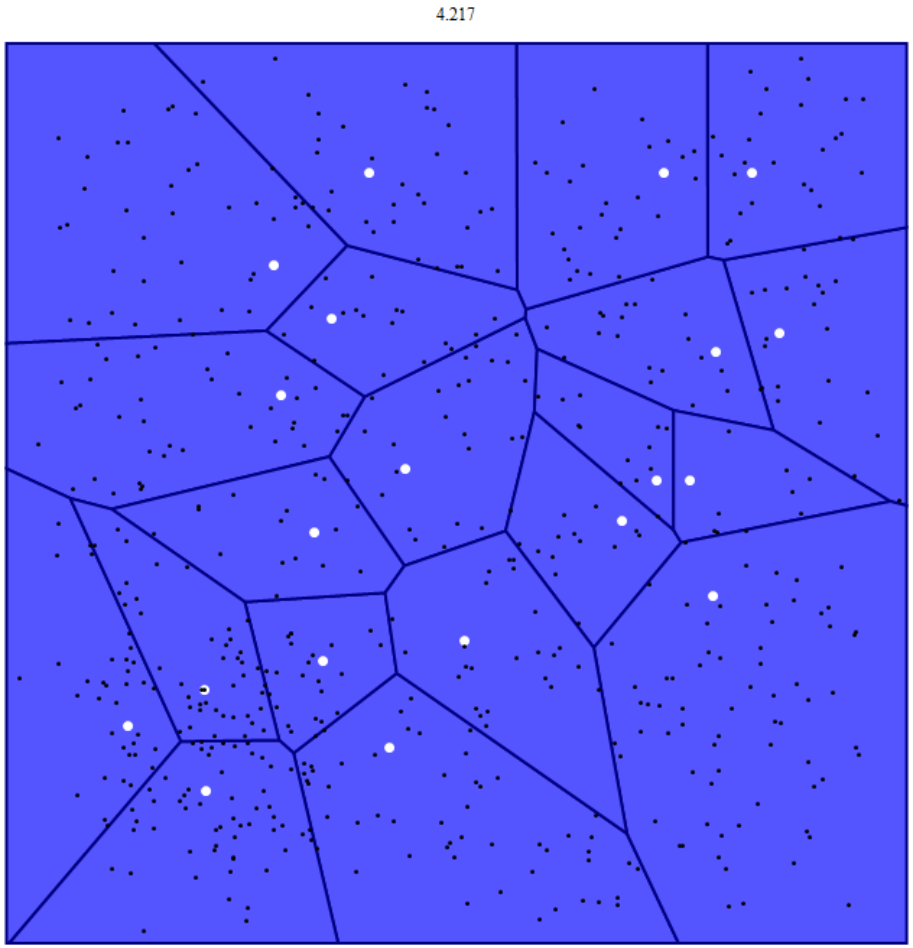


Lloyd-Max quantization

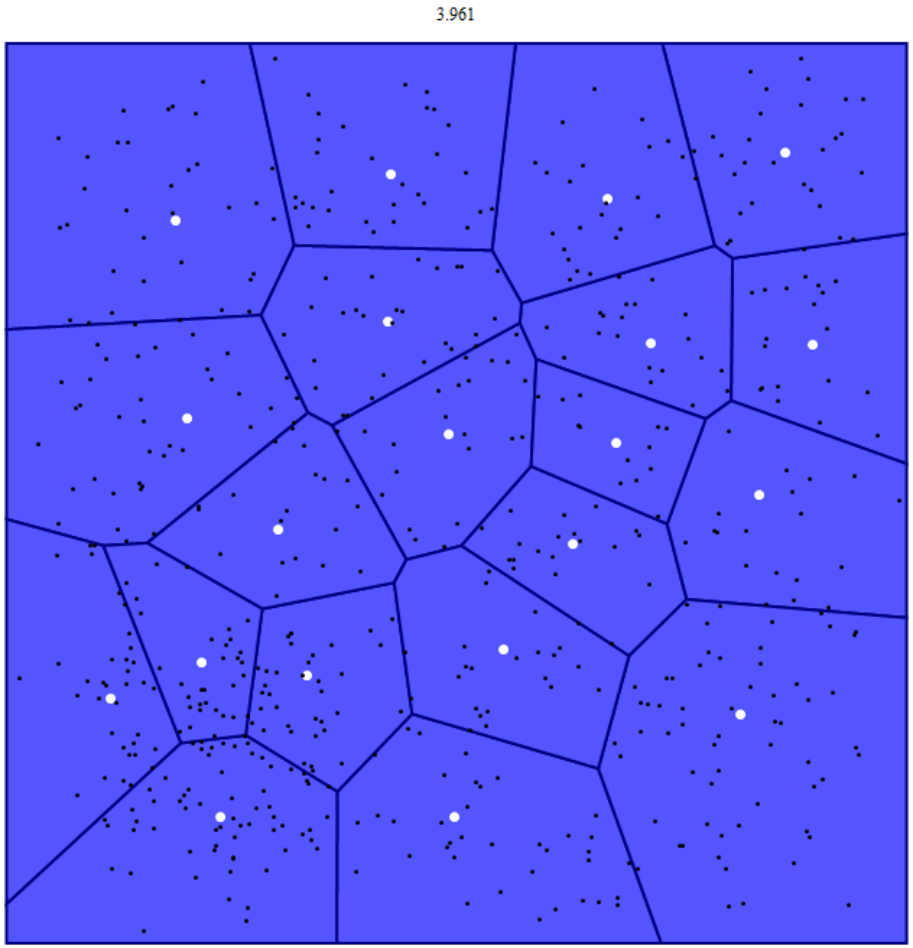
Lloyd-Max quantization can be generalized to higher dimensions.

- The bin boundary conditions in Equations (2) now become Voronoi cell conditions.
- The update step given by Equations (1) still remains: The new replacement values are the average of the signal values in the Voronoi cell.

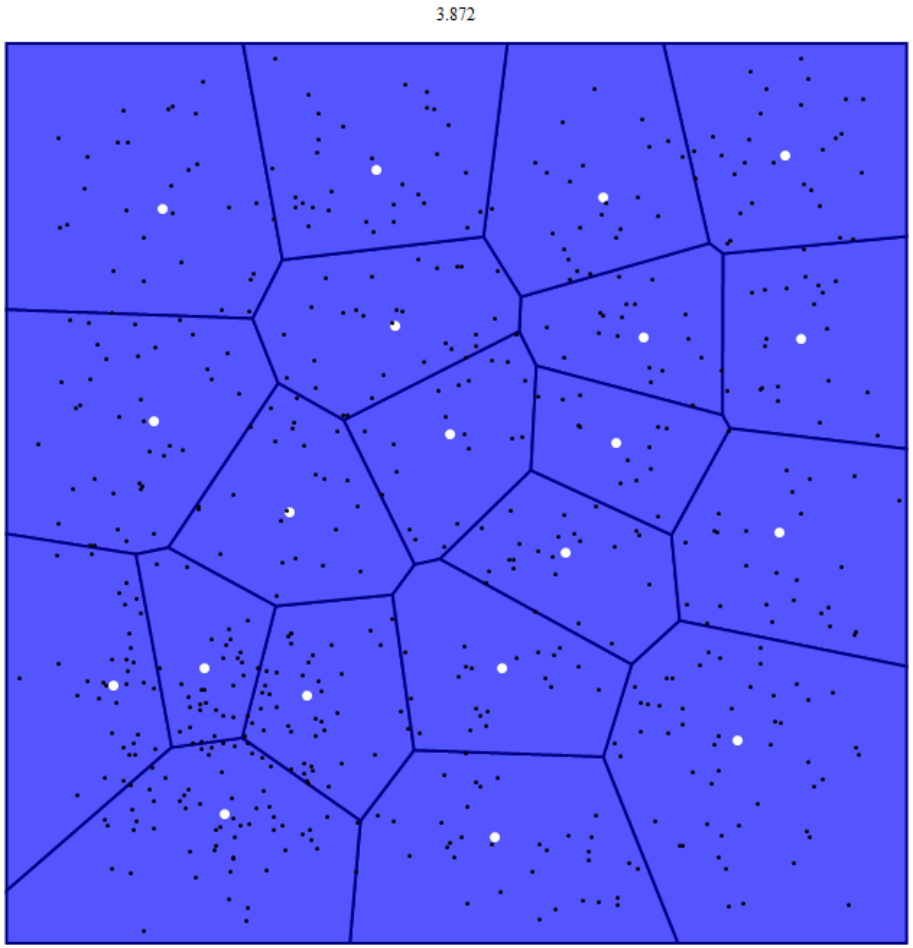
Here is a two-dimensional example:



Here is a two-dimensional example:

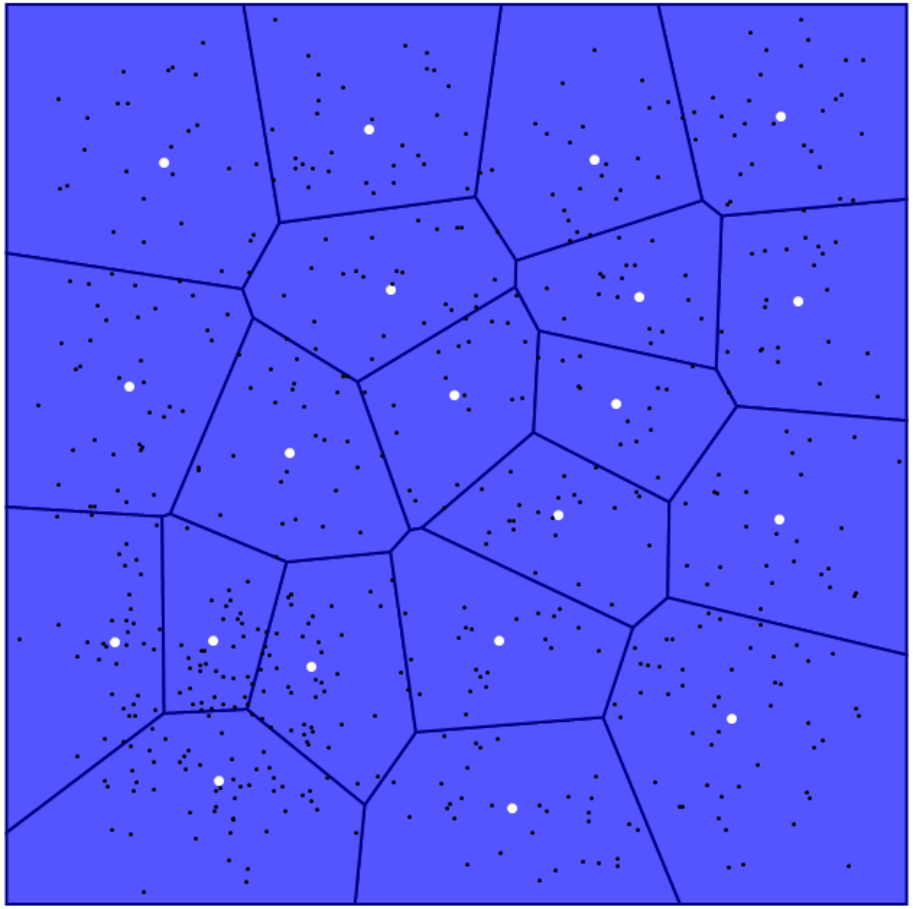


Here is a two-dimensional example:

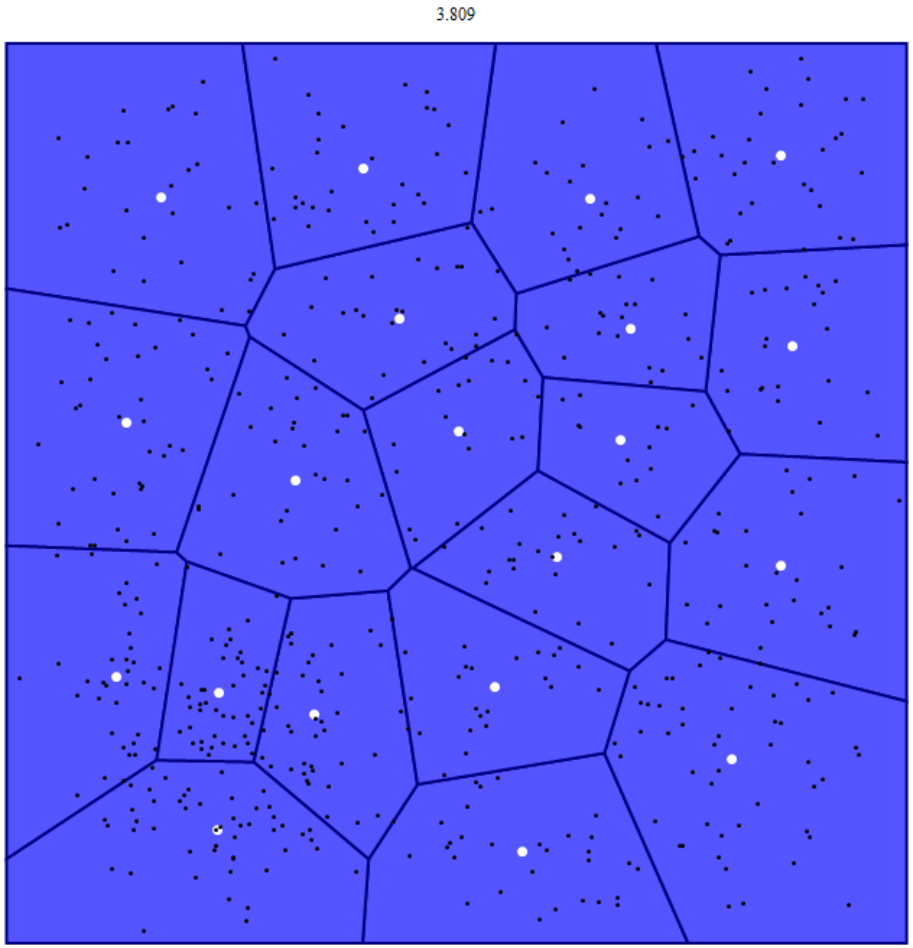


Here is a two-dimensional example:

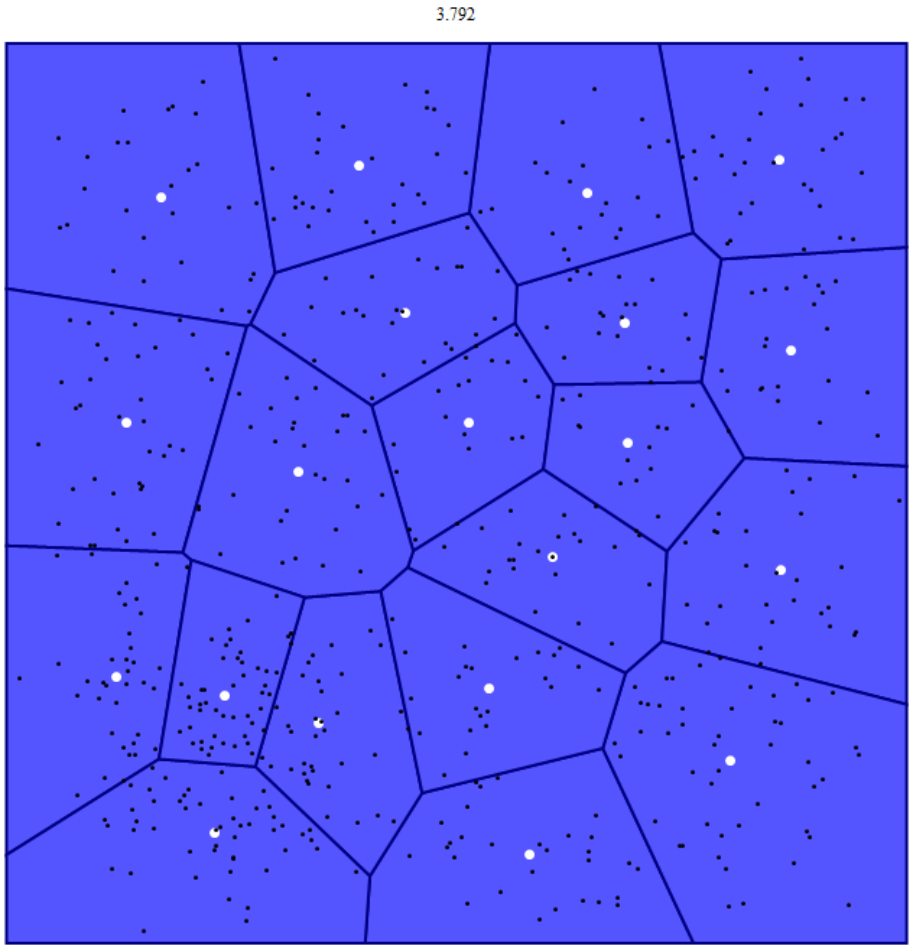
3.831



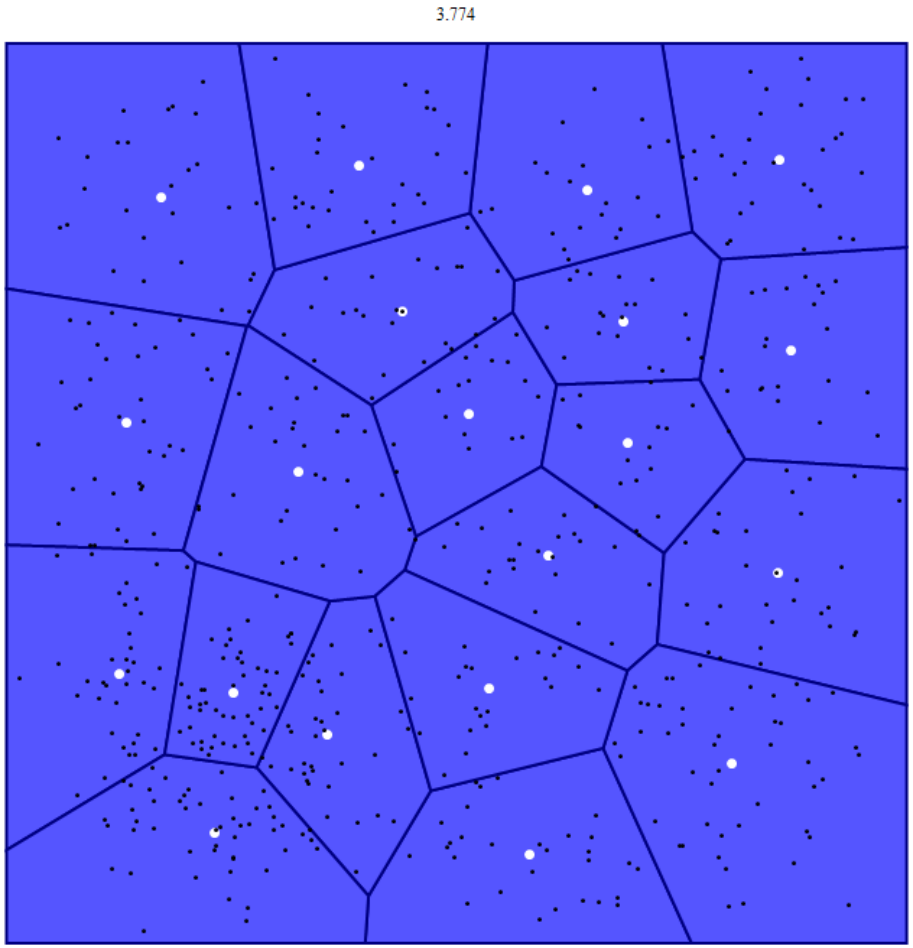
Here is a two-dimensional example:



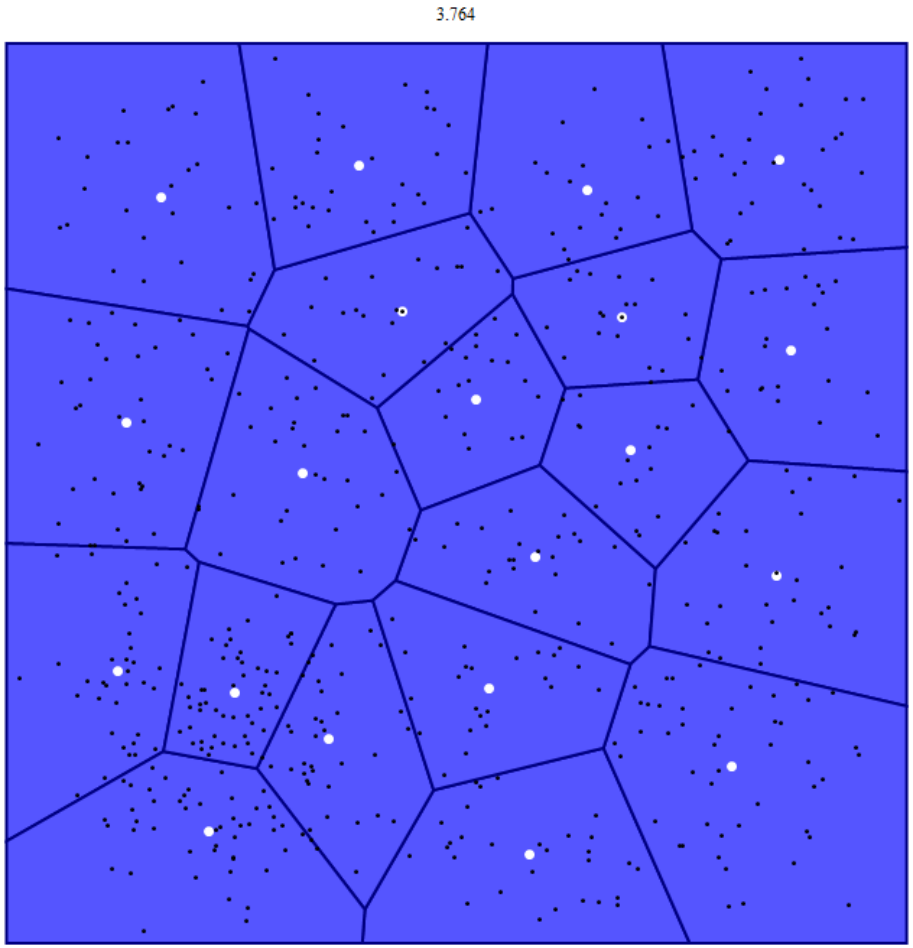
Here is a two-dimensional example:



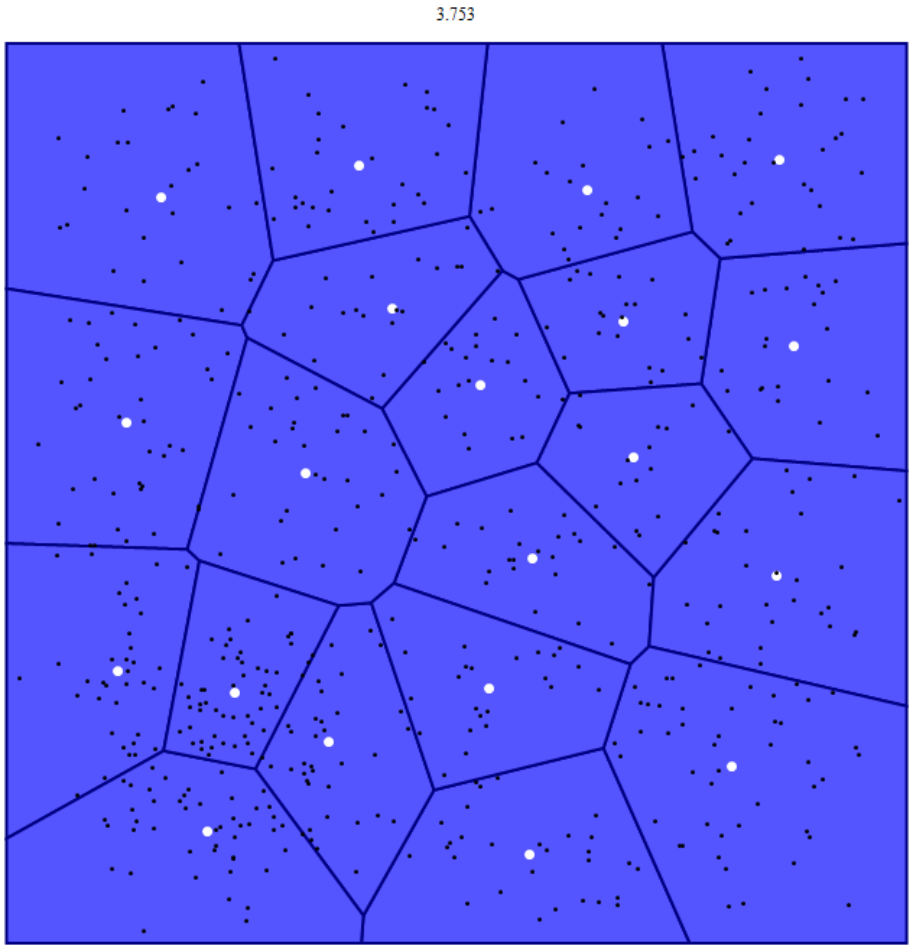
Here is a two-dimensional example:



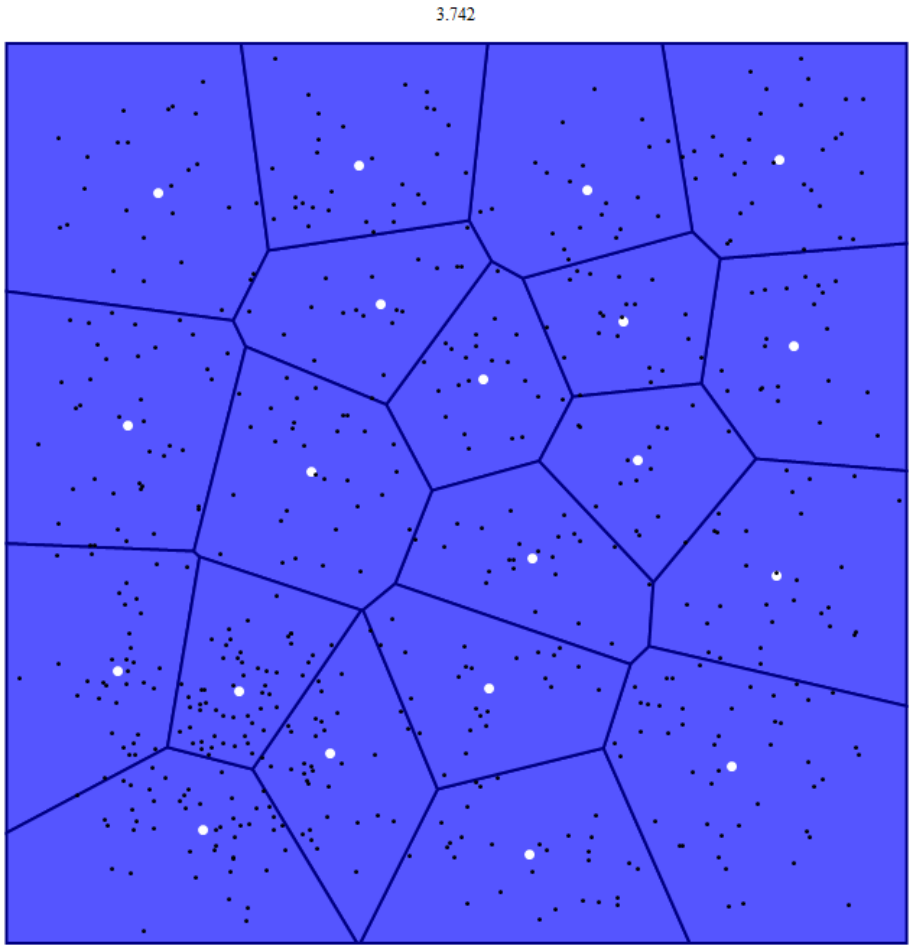
Here is a two-dimensional example:



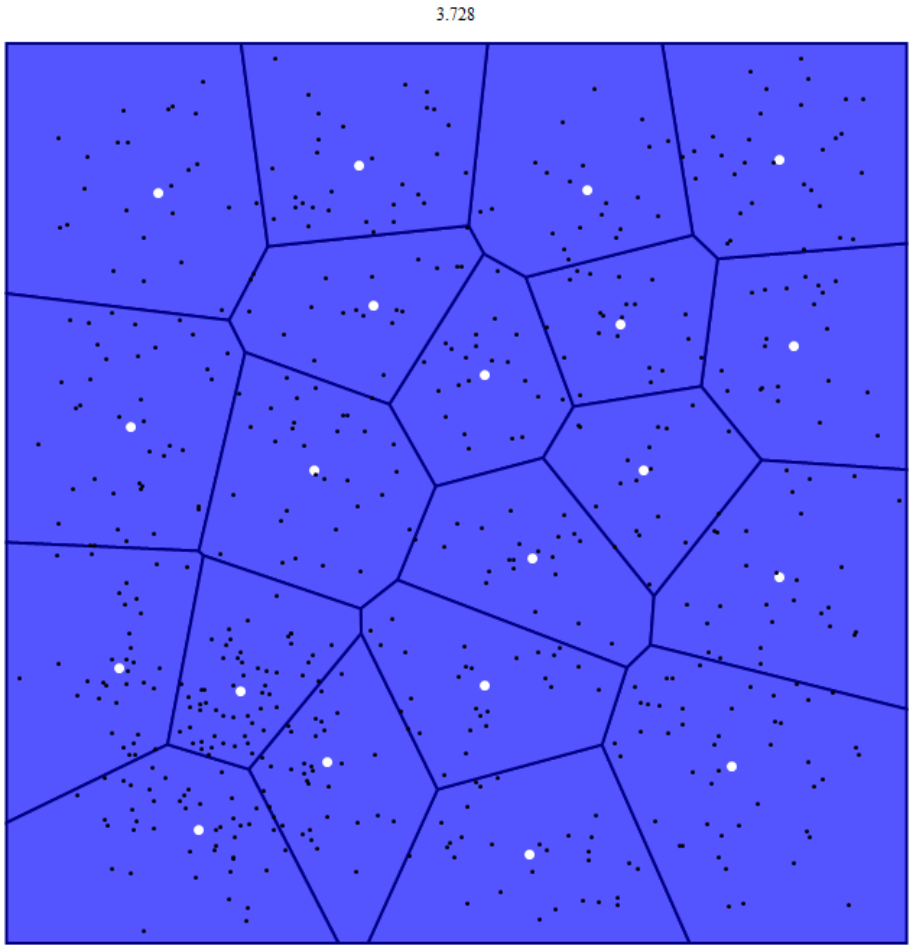
Here is a two-dimensional example:



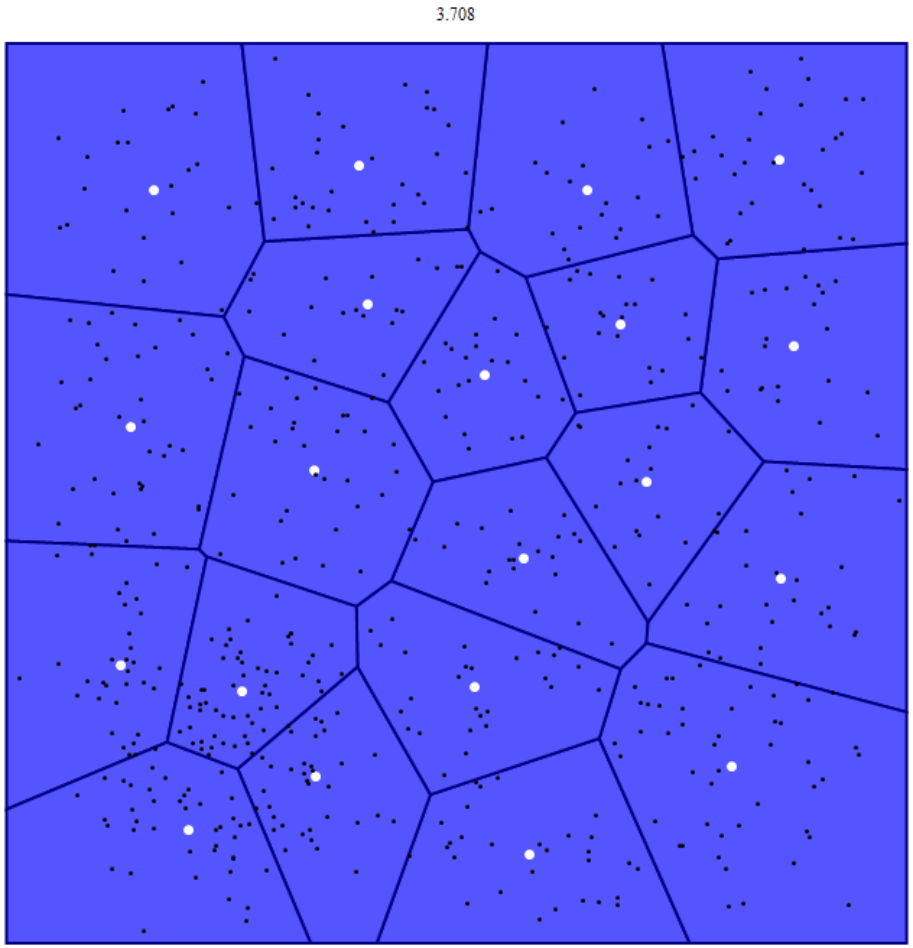
Here is a two-dimensional example:



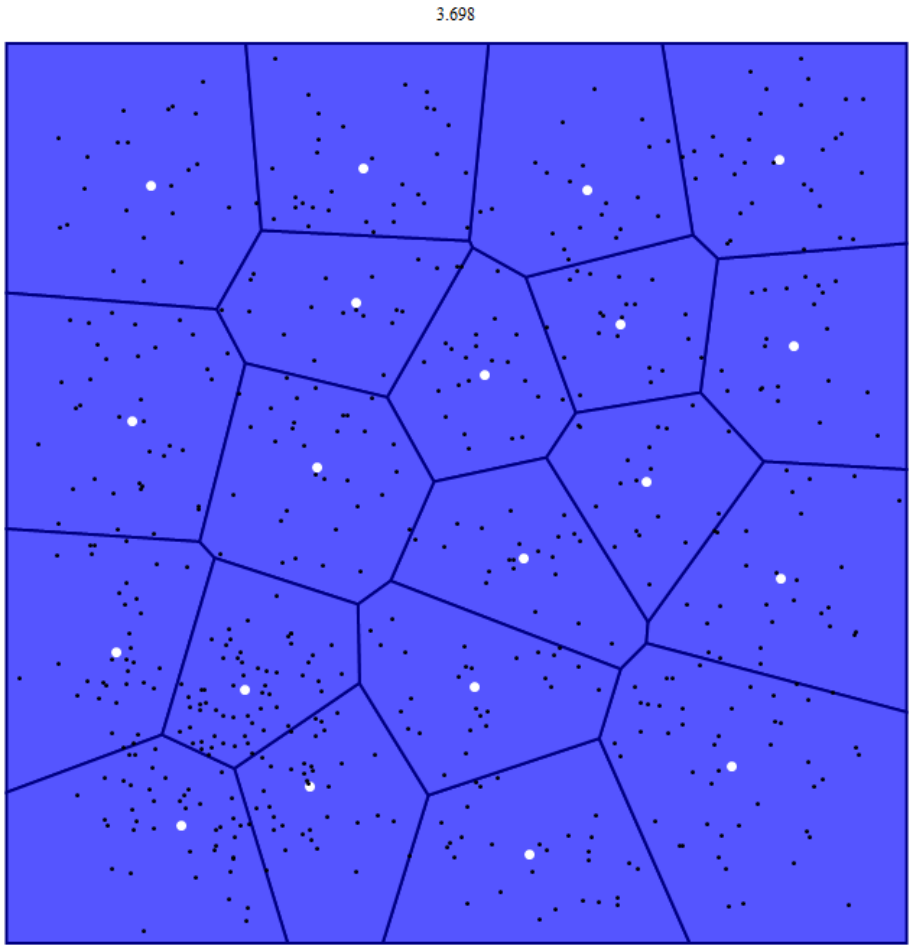
Here is a two-dimensional example:



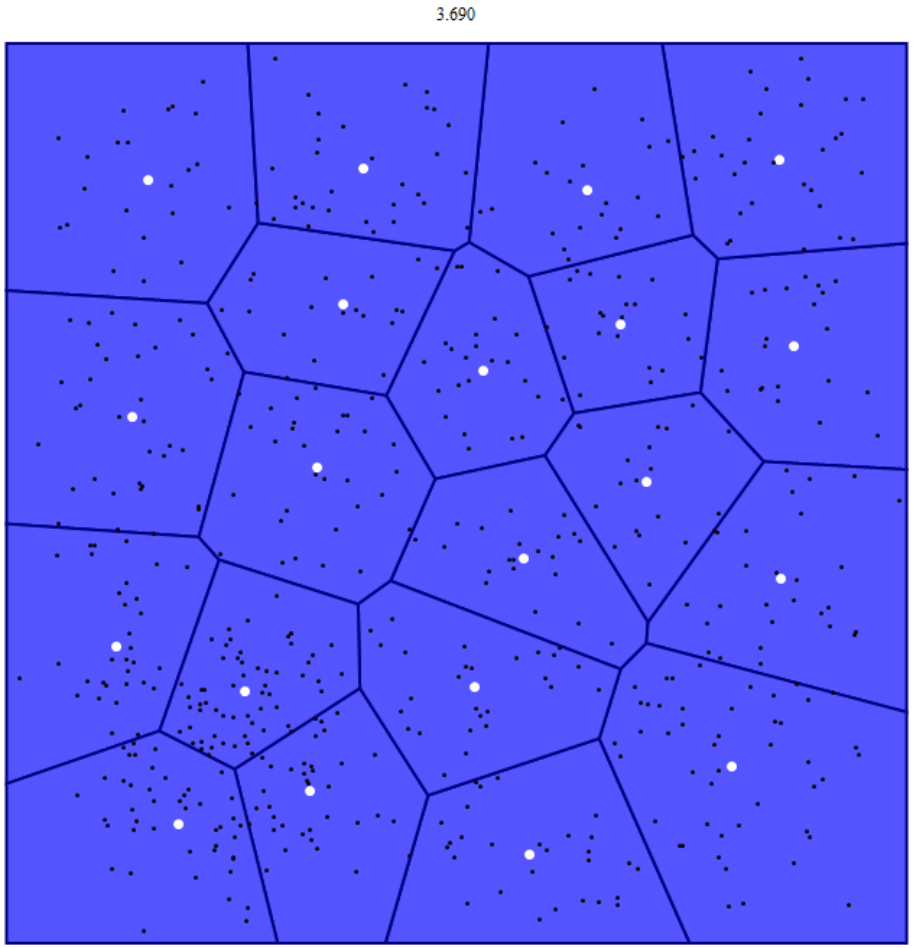
Here is a two-dimensional example:



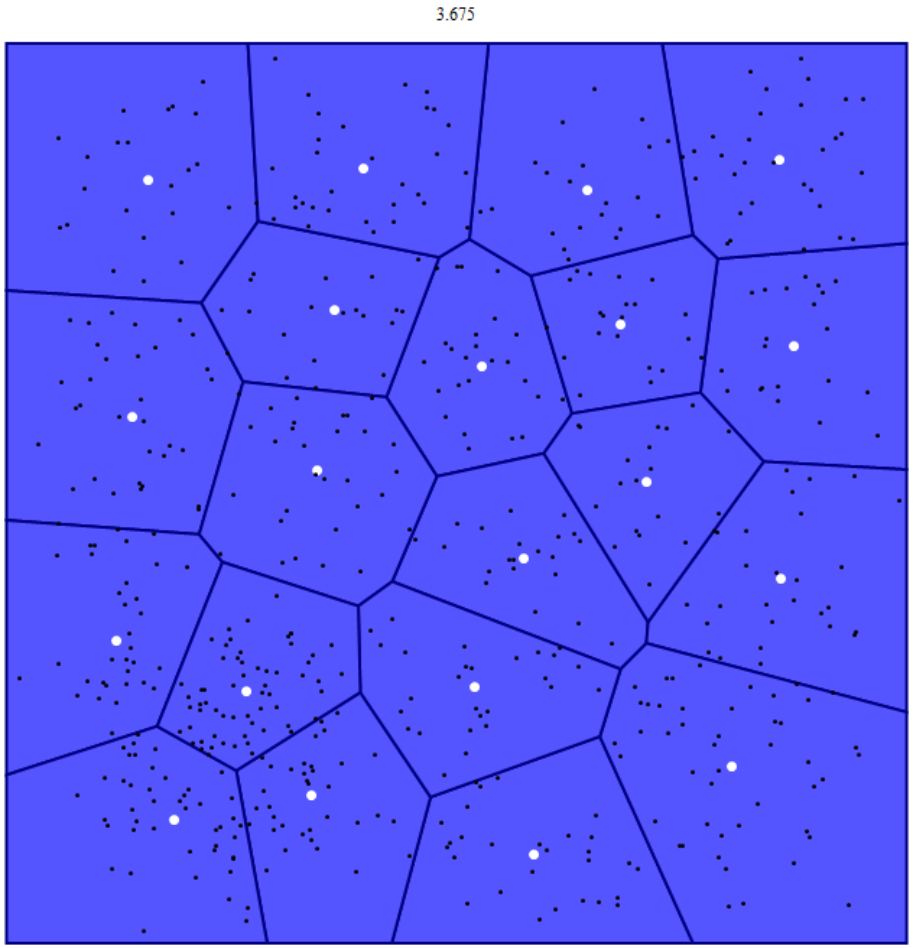
Here is a two-dimensional example:



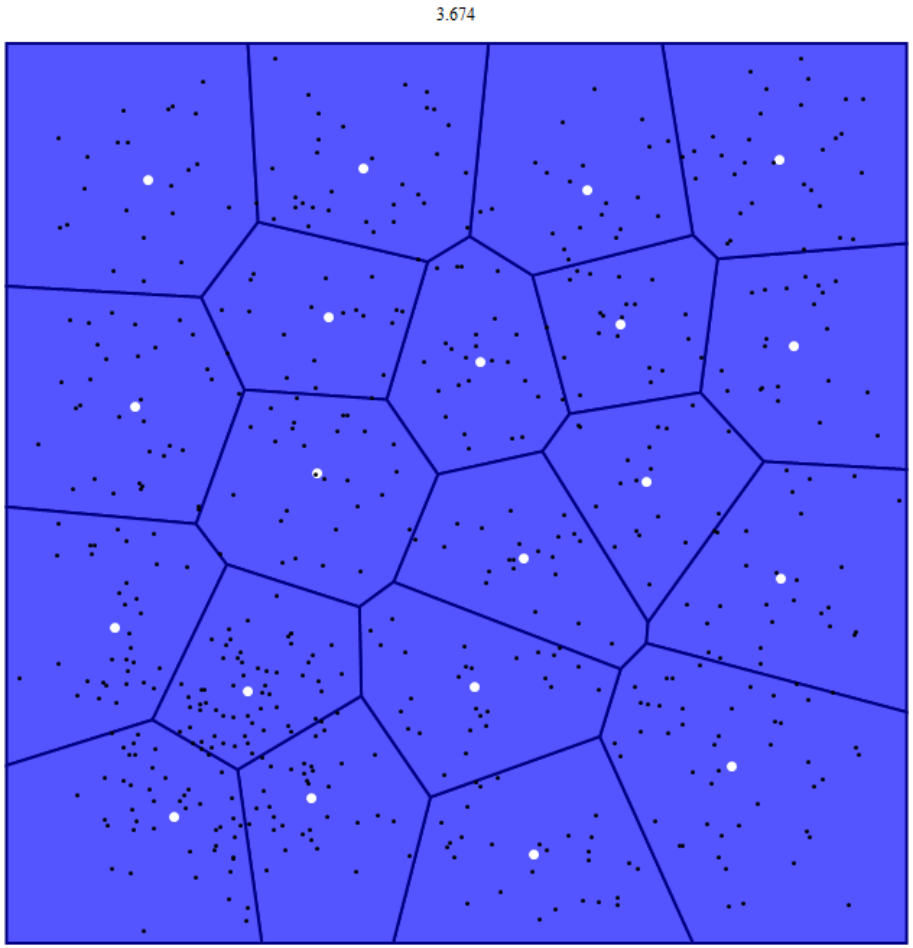
Here is a two-dimensional example:



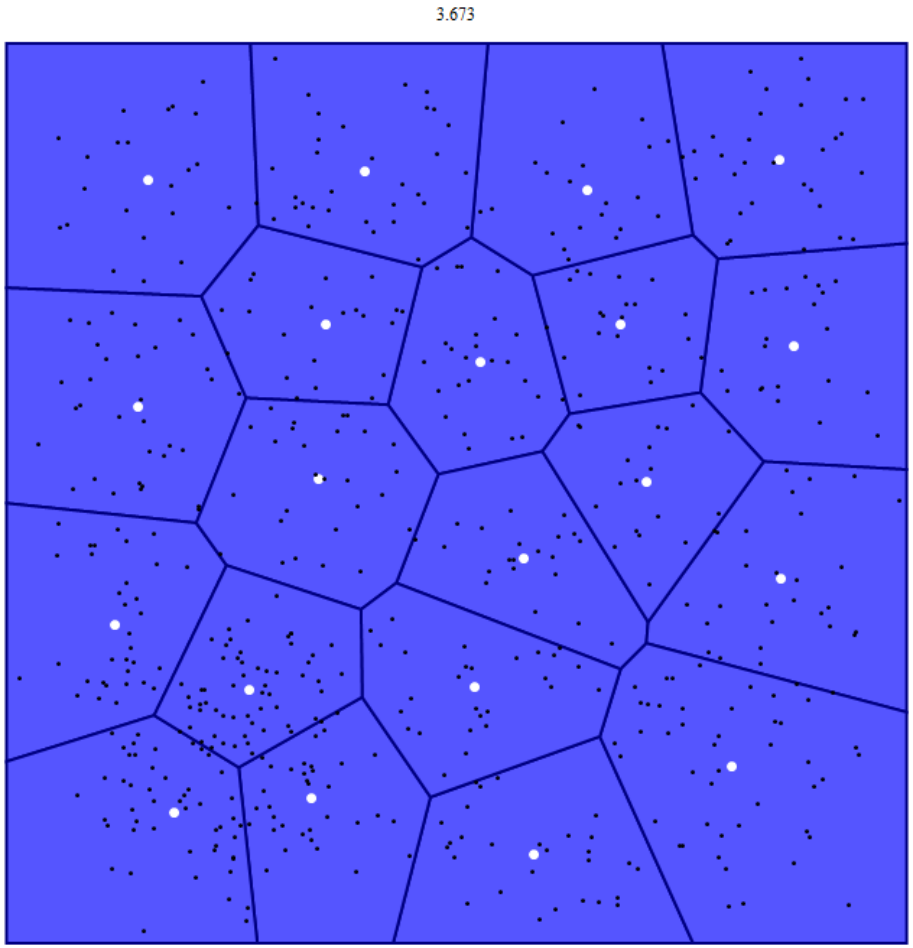
Here is a two-dimensional example:



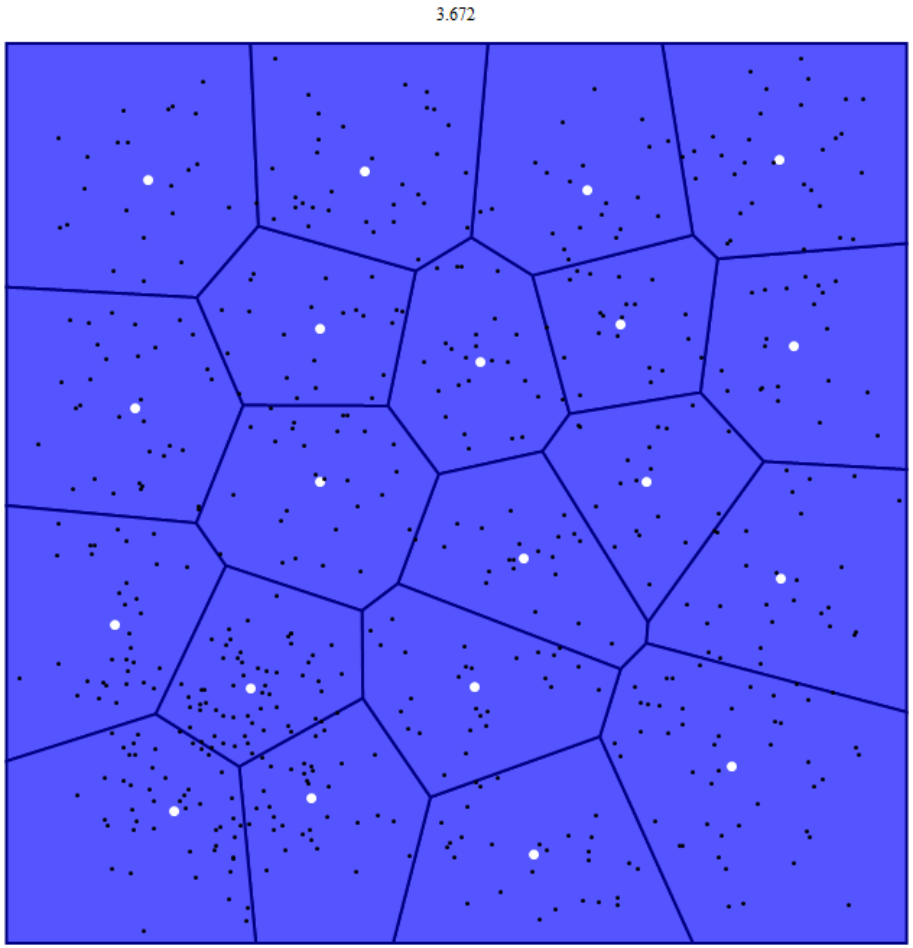
Here is a two-dimensional example:



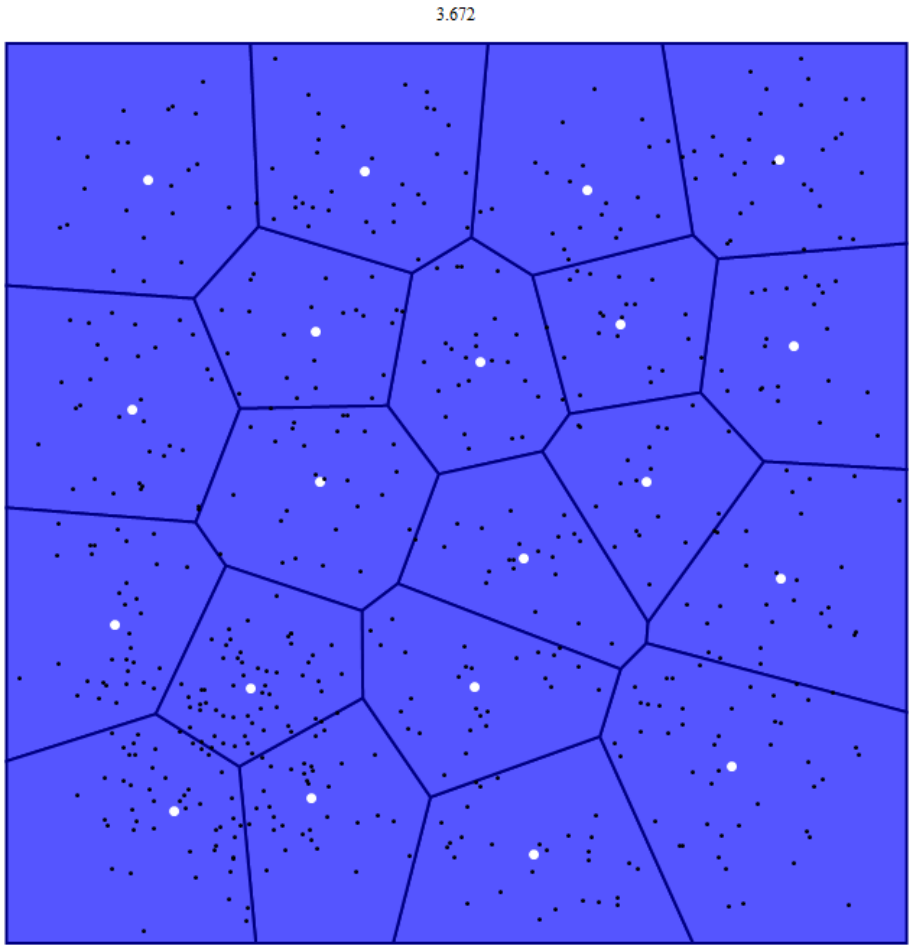
Here is a two-dimensional example:



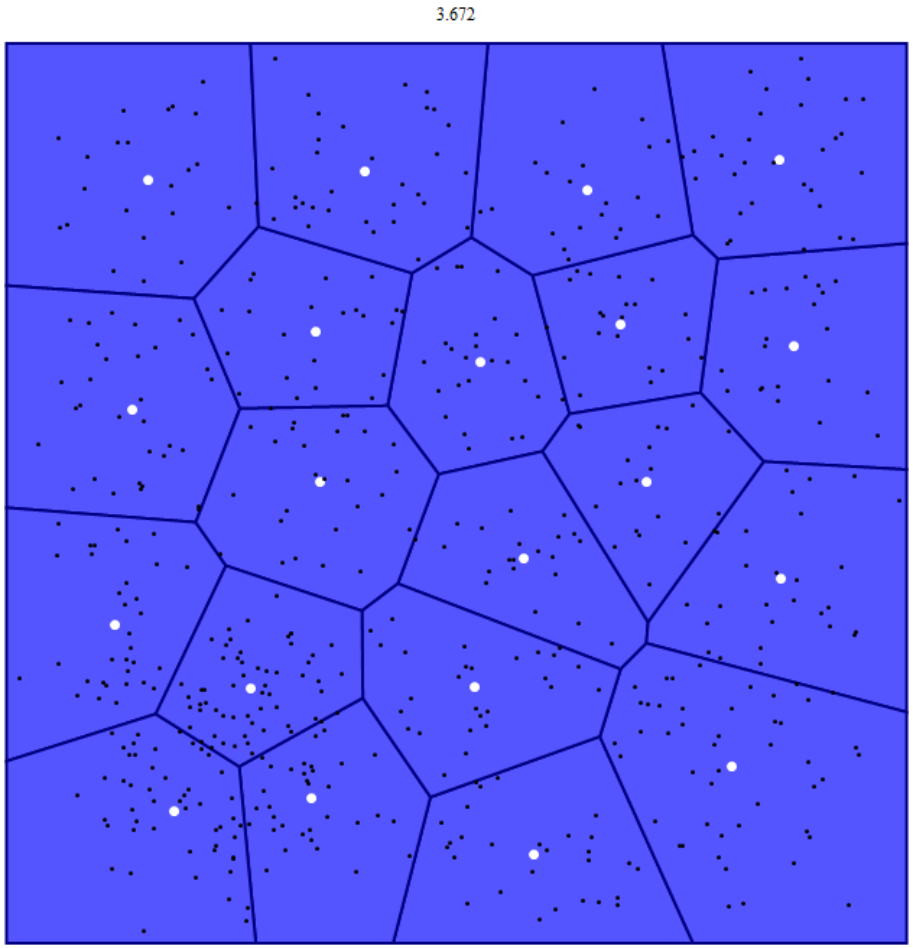
Here is a two-dimensional example:



Here is a two-dimensional example:



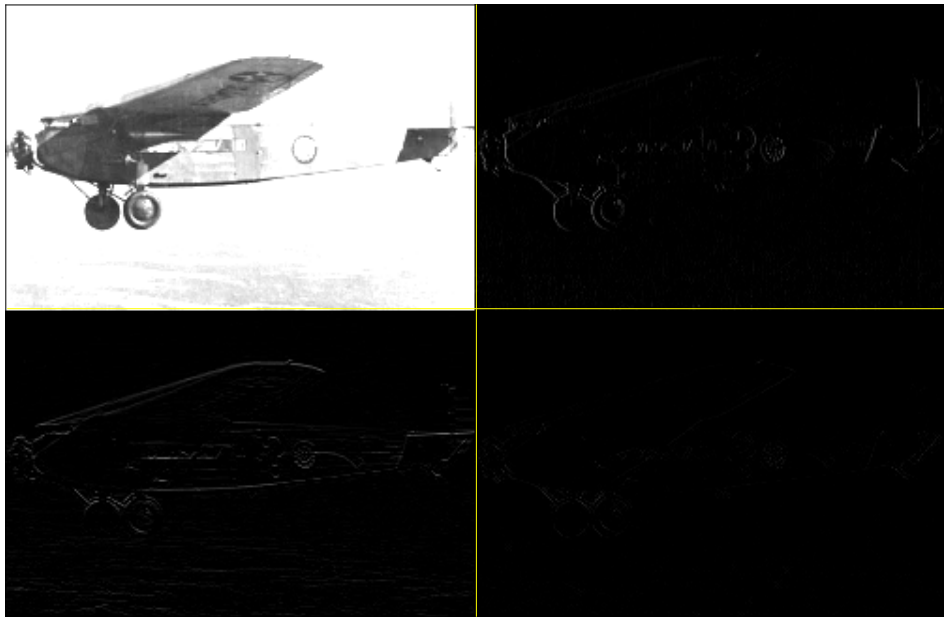
Here is a two-dimensional example:



Any Questions?

Helmut Knaust
hknaust@utep.edu

Any Questions?



Helmut Knaust
hknaust@utep.edu