Geometrical Methods for Adaptive Approximation of Image and Video Data

Laurent Demaret and Armin Iske

based on joint work with Nira Dyn and Wahid Khachabi

1 Introduction

Digital Image Compression: Basic Steps

(1) Data reduction from input image;

(2) Encoding of the reduced data at the sender;

(3) Transmission of the encoded data from the sender to the receiver;

(4) Decoding of the transmitted data at the receiver;

(5) Data reconstruction.

Original Image.

01011000110101100 10 . . .

$\longrightarrow$

Reconstruction.

Image Representation.

- A digital image I is a rectangular grid of pixels, X.
- Each pixel $x \in X$ bears a greyscale luminance $\mathrm{I}(x)$.
- We regard the image as a function, $\mathrm{I} : [X] \longrightarrow [0, 1, \dots, 2^r - 1]$, where the convex hull $[X]$ of X is the image domain.

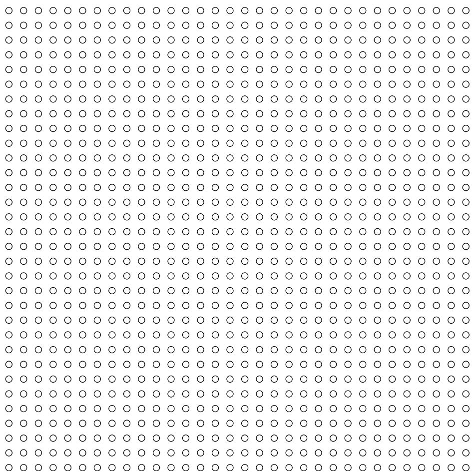


image domain $[X]$.

$$\xrightarrow{\mathrm{I}}$$

image $\mathrm{I}(X)$.

Image Approximation.

INPUT: The image $I = \{(x, I(x)) : x \in X\}$ is given by discrete pixel values in X.

OUTPUT: Reconstructed image $\tilde{I} = \{(x, \tilde{I}(x)) : x \in X\}$.

$\approx$

AIM. Increase *Peak Signal to Noise Ratio* (PSNR)

$$\text{PSNR} = 10 * \log_{10}\left(\frac{2^r \times 2^r}{\bar{\eta}^2(I, \tilde{I})}\right),$$

as much as possible, where

$$\bar{\eta}^2(I, \tilde{I}) = \frac{1}{|X|}\sum_{x \in X} |I(x) - \tilde{I}(x)|^2$$

denotes the *mean square error* (MSE).

2 Methods for Image Compression

Wavelets: The standard (EBCOT, JPEG2000)

Wavelet Image Approximation Scheme.

- The image is expanded in a fixed orthonormal basis of wavelets.
- The expansion coefficients below a given threshold are set to zero.

A mildly nonlinear approximation scheme.

Some recent highly nonlinear approximation schemes ...

... for capturing the image geometry.

- **Bandelets:** LePennec & Mallat (2005);
- **Brushlets:** Coifman & Meyer (1997);
- **Curvelets:** Candès & Donoho (2000, 2004/2005);
- **Contourlets:** Do & Vetterli (2005);
- **Directionlets:** Velisavljević, Beferull-Lozano, Vetterli & Dragotti (2006);
- **Shearlets:** Guo, Kutyniok, Labate, Lim (2006);
- **Wedgelets:** Donoho (1999); Romberg, Wakin & Baraniuk (2002);
- **The Easy Path Wavelet Transform (EPWT):** Plonka(2009), Plonka, Tenorth & I.(2010), Plonka, Tenorth & Roşca (2009);
- **Nonlinear edge-adapted multiscale decomposition:** Cohen & Matei (2001);
- **Adaptive approximation by anisotropic triangulations:**
  - **Generic triangulations and simulated annealing:** Lehner, Umlauf, Hamann (2007)
  - **Adaptive thinning algorithms:** Demaret, Dyn & I. (2006), Demaret & I. (2006)
  - **Anisotropic geodesic triangulations:** Bougleux, Peyré & L. Cohen (2009)
  - **Greedy triangle bisections:** A. Cohen, Dyn, Hecht & Mirebeau (2010)

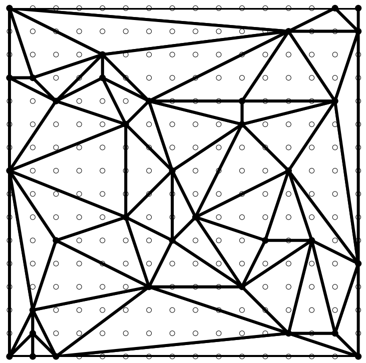
3 Linear Splines over Triangulations

Definition. A triangulation of a planar point set $Y = \{y_1, \ldots, y_N\}$ is a collection $\mathcal{T}(Y) = \{T\}_{T \in \mathcal{T}(Y)}$ of triangles in the plane, such that

(T1) the vertex set of $\mathcal{T}(Y)$ is Y;

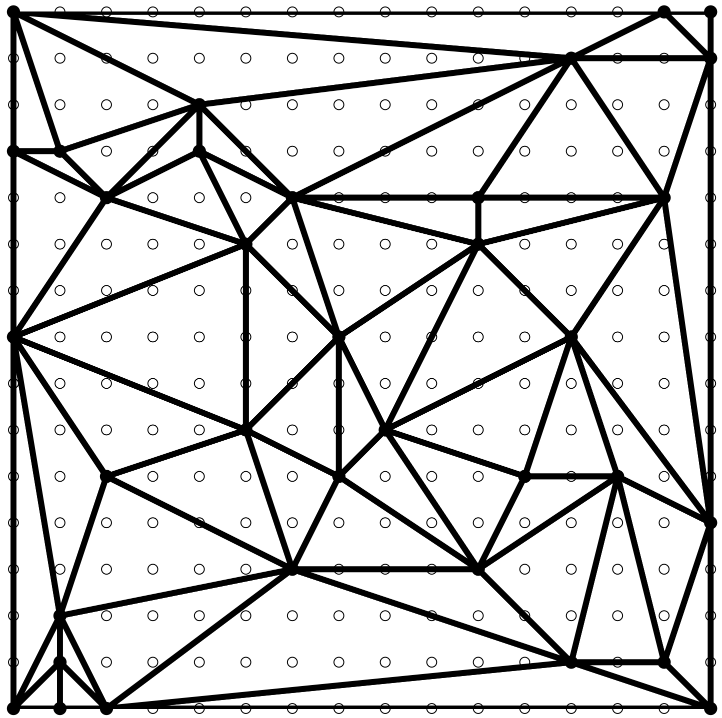
(T2) any pair of two distinct triangles in $\mathcal{T}(Y)$ intersect at most at one common vertex or along one common edge;

(T3) the convex hull $[Y]$ of Y coincides with the area covered by the union of the triangles in $\mathcal{T}(Y)$.

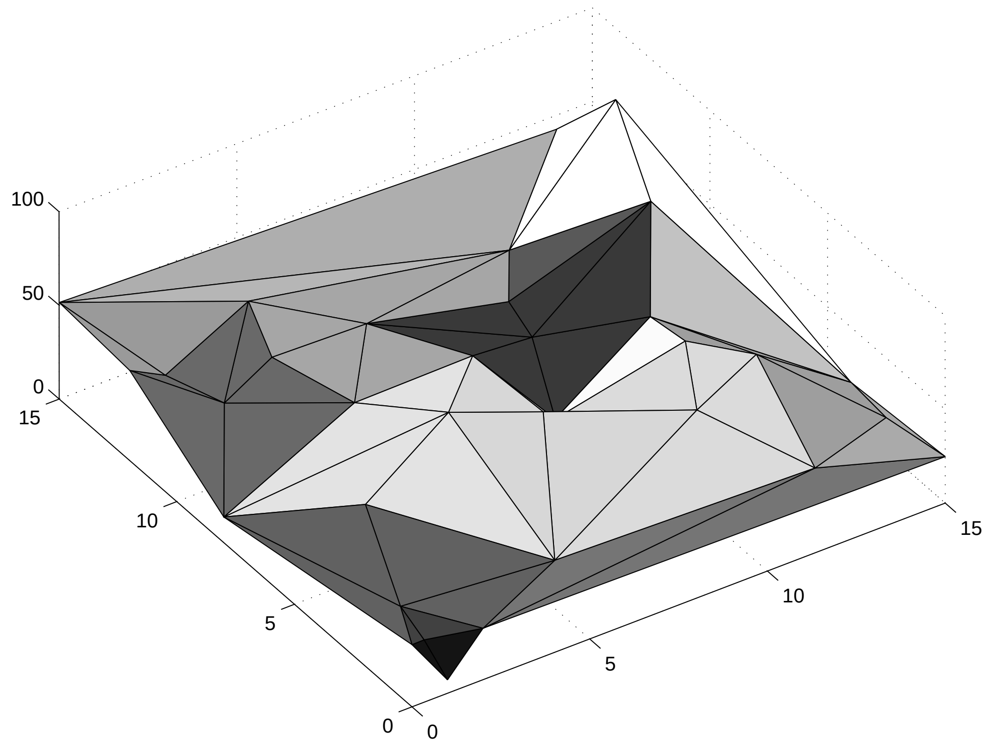


A triangulation of pixels.

Linear Splines over Triangulations.



Triangulation of pixels.



Linear spline over triangulation.

Approximation Spaces.

- Given any triangulation $\mathcal{T}(Y)$ of Y, we denote by

$$\mathcal{S}_Y = \left\{ s : s \in C([Y]) \text{ and } s\big|_T \text{ linear for all } T \in \mathcal{T}(Y) \right\},$$

  the spline space containing all continuous functions over $[Y]$ whose restriction to any triangle in $\mathcal{T}(Y)$ is linear.

- Any element in $\mathcal{S}_Y$ is referred to as a linear spline over $\mathcal{T}(Y)$.

- For given function values $\{I(y) : y \in Y\}$, there is a unique linear spline, $L(Y, I) \in \mathcal{S}_Y$, which interpolates I at the points of Y, i.e.,

$$L(Y, I)(y) = I(y), \qquad \text{for all } y \in Y.$$

Example 1: Geometrical Image `PQuad`.

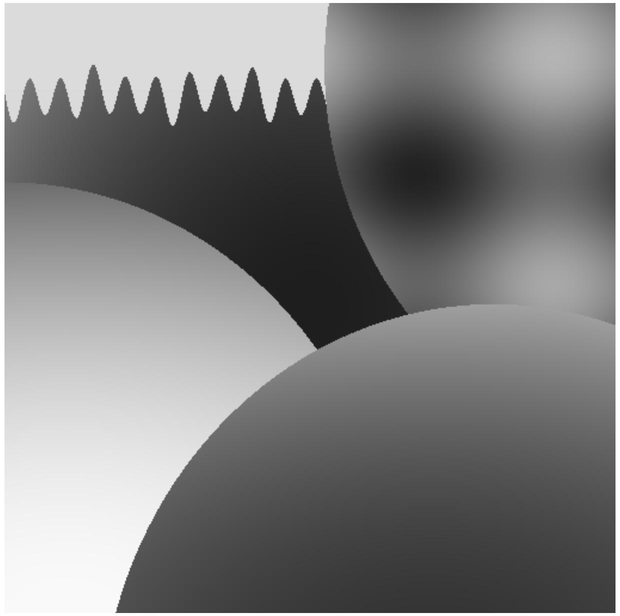
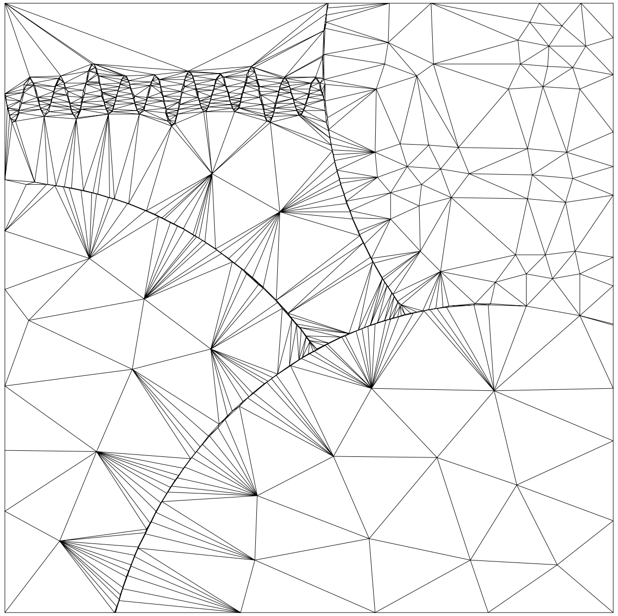
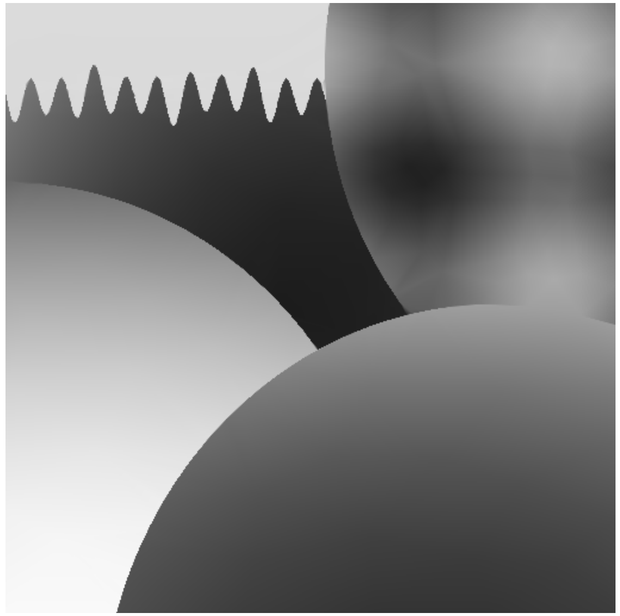


Image `PQuad` **of size** (512×512).



Adaptive Triangulation with 800 vertices.



Reconstruction at PSNR 42.85 `db`.

Example 2: Geometrical Image `Game`.

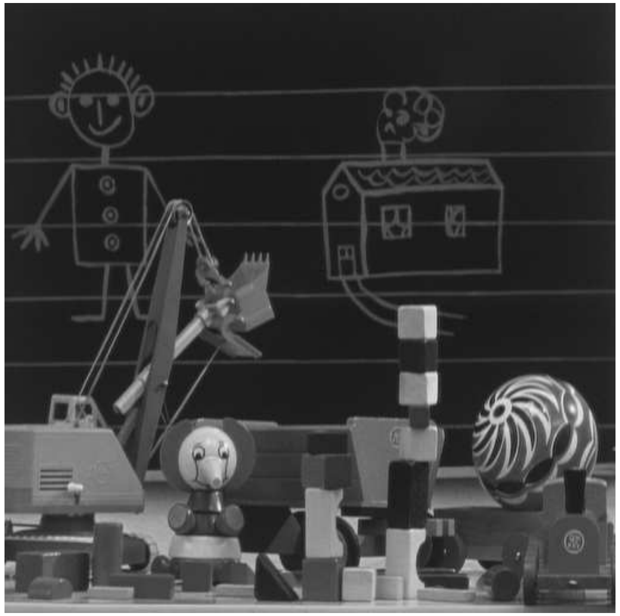
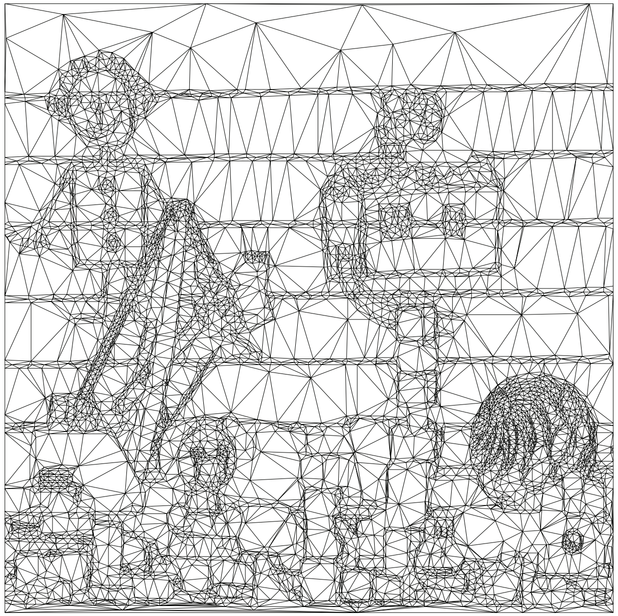
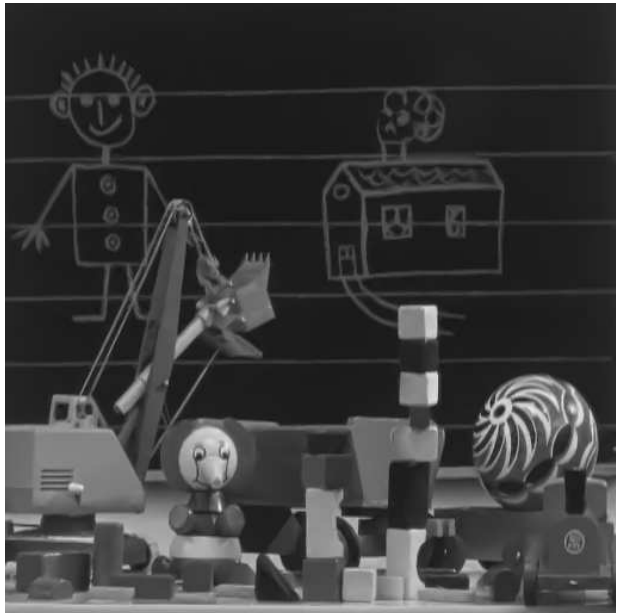


Image `Game` of size (512×512).



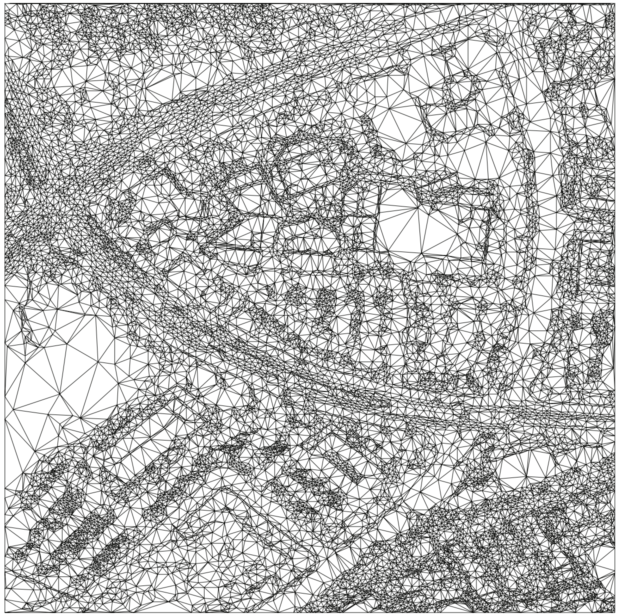
Adaptive Triangulation with 6000 vertices.



Reconstruction at PSNR 36.54 `db`.

Example 3: Multiscale Image `Aerial`.

Image `Aerial` **of size** (512×512).

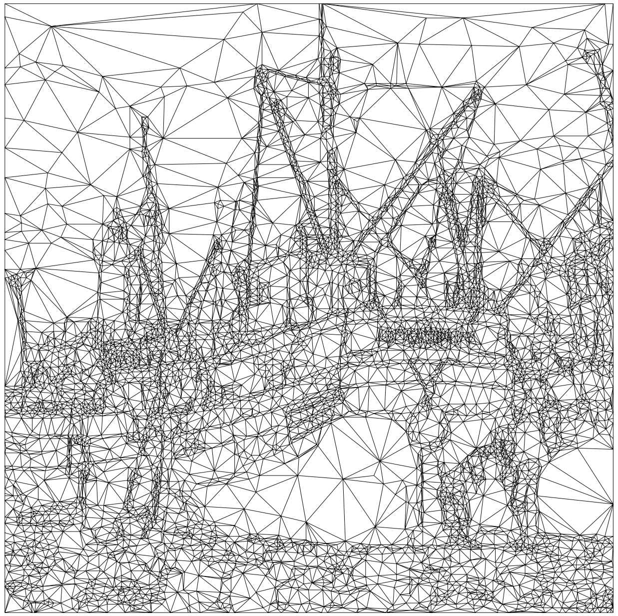


Adaptive Triangulation with 16000 vertices.

Reconstruction at PSNR 30.33 `db`.

Example 4: Multiscale Image `Boat`.

Image `Boat` **of size** (512×512).



Adaptive Triangulation with 7000 vertices.

Reconstruction at PSNR 31.83 `db`.

4 Approximation over Anisotropic Triangulations

Goal: On input image $I = \{(x, I(x)) : x \in X\}$,

- determine a *good* adaptive spline space $\mathcal{S}_Y$, where $Y \subset X$;
- determine from $\mathcal{S}_Y$ the unique best approximation $L^*(Y, I) \in \mathcal{S}_Y$ satisfying

$$\sum_{x \in X} |L^*(Y, I)(x) - I(x)|^2 = \min_{s \in \mathcal{S}_Y} \sum_{x \in X} |s(x) - I(x)|^2.$$

- Encode the linear spline $L^* \in \mathcal{S}_Y$;
- Decode $L^* \in \mathcal{S}_Y$, and so obtain the reconstructed image $\tilde{I} = \{(x, L(Y, \tilde{I})(x)) : x \in X\}$, where $L(Y, \tilde{I}) \approx L^*(Y, I)$.

OBS! Key Step: Construction of *anisotropic* triangulation $\mathcal{T}(Y)$ for $Y \subset X$.

- One possible approach is by *adaptive thinning* (AT).
- In AT, we take the *Delaunay triangulation* $\mathcal{D}(Y)$ of Y for $\mathcal{S}_Y$,

The Bramble-Hilbert Lemma.

Recall classical error estimates from finite element methods (FEM).

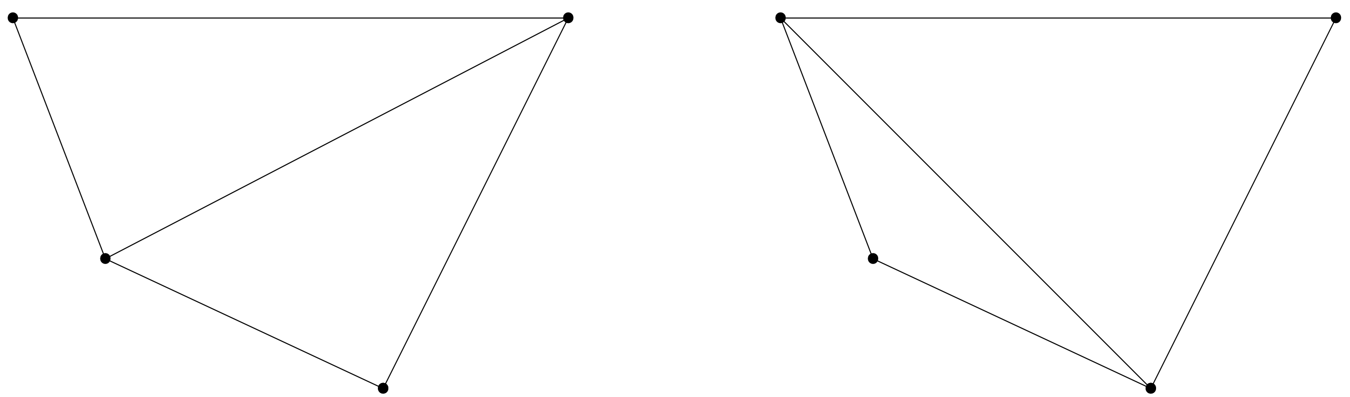
Bramble-Hilbert: For any image f from Sobolev space $W^{2,2}(T)$, $T \in \mathcal{T}(Y)$, we obtain the basic error estimate

$$\|f - \Pi_{\mathcal{S}_Y} f\|_{L^2(T)} \leq |f|_{W^{2,2}(T)}, \qquad \text{for } f \in W^{2,2}(T),$$

where $\Pi_{\mathcal{S}_Y} f$ is the orthogonal L^2-projection of f onto $\mathcal{S}_Y$. ■

5 Delaunay Triangulations

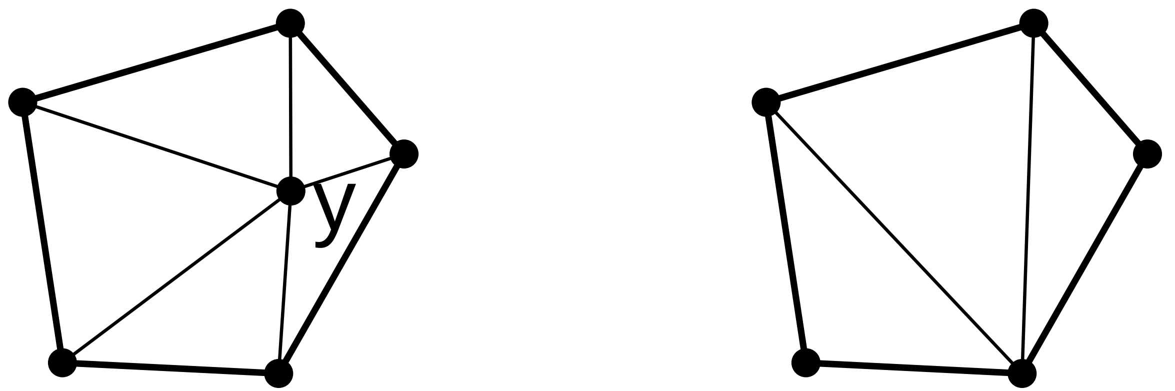
Definition. The `Delaunay triangulation` $\mathcal{D}(X)$ of a discrete planar point set X is a triangulation of X, such that the circumcircle for each of its triangles does not contain any point from X in its interior.



Two triangulations of a convex quadrilateral, $\mathcal{T}$ (left) and $\tilde{\mathcal{T}}$ (right).

Properties of Delaunay Triangulations.

- **Uniqueness.**
  Delaunay triangulation $\mathcal{D}(X)$ is *unique*, if no four points in X are co-circular.
- **Complexity.**
  For any point set X, its Delaunay triangulation $\mathcal{D}(X)$ can be computed in $\mathcal{O}(N \log N)$ steps, where $N = |X|$.
- **Local Updating.**
  For any X and $x \in X$, the Delaunay triangulation $\mathcal{D}(X \setminus x)$ of the point set $X \setminus x$ can be computed from $\mathcal{D}(X)$ by retriangulating the *cell* $\mathcal{C}(x)$ of x.

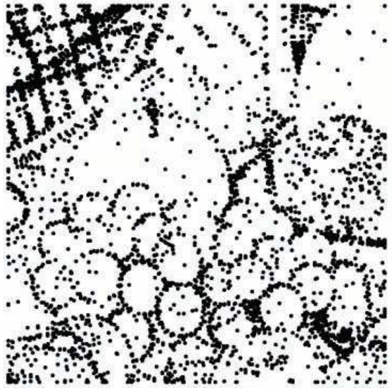


Removal of the node y, and retriangulation of its cell $\mathcal{C}(y)$.

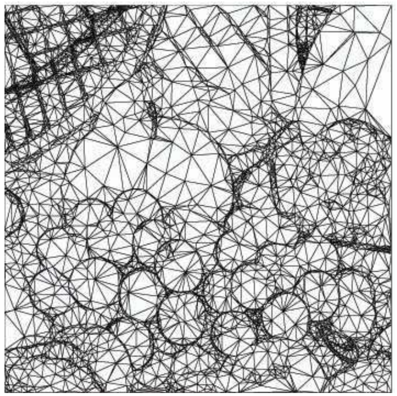
6 Adaptive Thinning

Popular Example: Test Image `Fruits`.

Original Image (512×512).



4044 significant pixels.



Delaunay Triangulation.

Image Reconstruction.

Adaptive Thinning Algorithm.

INPUT. $I = \{0, 1, \ldots, 2^r - 1\}^X$, pixels and luminances, where
X set of pixels, r number of bits for representation of luminances.

(1) Let $X_N = X$;

(2) **FOR** $k = 1, \ldots, N - n$

(2a) Find a `least significant` pixel $x \in X_{N-k+1}$;

(2b) Let $X_{N-k} = X_{N-k+1} \setminus x$;

- **OUTPUT:** Data hierarchy

$$X_n \subset X_{n+1} \subset \cdots \subset X_{N-1} \subset X_N = X$$

of nested subsets of X.

Controlling the Mean Square Error.

- For a given mean square error (MSE), $\bar{\eta}^*$, the adaptive thinning algorithm can be changed in order to terminate when for the first time, the MSE value corresponding to the current linear spline $L(X_p, I)$ is above $\bar{\eta}^*$, for some X_p in the data hierarchy, $n = p$ a posteriori.

- We take as the final approximation to the image the linear spline $L^*(X_{p+1}, I)$, and so we let $Y = X_{p+1}$.

- Observe that $L^*(X_{p+1}, I)$ satisfies

$$\sum_{x \in X} |L^*(X_{p+1}, I)(x) - I(x)|^2 / |X_{p+1}| \leq \bar{\eta}^*,$$

as desired.

7 Pixel Significance Measures

Quality Measure: Current ℓ_2-Square Error.

$$\eta(Y;X) = \sum_{x \in X} |L(I,Y)(x) - I(x)|^2, \qquad \text{for } Y \subset X.$$

Anticipated Error for the Greedy Removal of one Pixel.

$$e(y) = \eta(Y \setminus y; X), \qquad \text{for } y \in Y.$$

Definition. (Adaptive Thinning Algorithm AT).
For $Y \subset X$, a point $y^* \in Y$ is said to be `least significant` in Y, iff it satisfies

$$e(y^*) = \min_{y \in Y} e(y).$$

Aim: Compute anticipated error *locally*.

$$\begin{aligned}\eta(Y\setminus y;X) &= \eta(Y\setminus y;X\setminus\mathcal{C}(y))+\eta(Y\setminus y;X\cap\mathcal{C}(y))\\ &= \eta(Y;X\setminus\mathcal{C}(y))+\eta(Y\setminus y;X\cap\mathcal{C}(y))\\ &= \eta(Y;X)+\eta(Y\setminus y;X\cap\mathcal{C}(y))-\eta(Y;X\cap\mathcal{C}(y)).\end{aligned}$$

where $\mathcal{C}(y)$ is the cell of y in the Delaunay triangulation $\mathcal{D}(Y)$ of Y.

Therefore, minimizing $e(y)$ is equivalent to minimizing

$$e_\delta(y)=\eta(Y\setminus y;X\cap\mathcal{C}(y))-\eta(Y;X\cap\mathcal{C}(y)),\qquad \text{for } y\in Y.$$

Proposition. For $Y\subset X$, a point $y^*\in Y$ is, according to the criterion **AT**, `least significant` in Y, iff it satisfies

$$e_\delta(y^*)=\min_{y\in Y} e_\delta(y).$$

Greedy Two-Point-Removal.

Anticipated Error for the Removal of two Points.

$$e(y_1, y_2) = \eta(Y \setminus \{y_1, y_2\}; X) \qquad \text{for } y_1, y_2 \in Y,$$

can be rewritten as $e(y_1, y_2) = \eta(Y; X) + e_\delta(y_1, y_2)$, where

$$e_\delta(y_1, y_2) = \eta(Y \setminus \{y_1, y_2\}; X \cap (\mathcal{C}(y_1) \cup \mathcal{C}(y_2))) - \eta(Y; X \cap (\mathcal{C}(y_1) \cup \mathcal{C}(y_2))),$$

which can for $[y_1, y_2] \notin \mathcal{D}(Y)$ be simplified as

$$e_\delta(y_1, y_2) = e_\delta(y_1) + e_\delta(y_2).$$

Definition. (Adaptive Thinning Algorithm AT^2).
For $Y \subset X$, a point pair $y_1^*, y_2^* \in Y$ is said to be `least significant` in Y, iff

$$e_\delta(y_1^*, y_2^*) = \min_{y_1, y_2 \in Y} e_\delta(y_1, y_2).$$

8 Implementation of Adaptive Thinning.

Efficient Implementation of Algorithm AT.

Initialization.

- Compute Delaunay triangulation $\mathcal{D}(X)$;
- Compute $e_\delta(x)$ for all $x \in X$ and store nodes of $\mathcal{D}(X)$ in a `Heap`.

Removal Step. For current $Y \subset X$

- Pop root $y^* \in Y$ from `Heap`, update `Heap`;
- Remove y^* from $\mathcal{D}(Y)$ and compute $\mathcal{D}_{Y \setminus y^*}$;
- Reattach *historical points* in $\mathcal{C}(y^*) \cap (X \setminus Y)$;
- Attach y^* to new triangle in $\mathcal{C}(y^*)$;
- Update $e_\delta(y)$ for neighbours of y^* and update `Heap`.

Total Complexity. $\mathcal{O}(N \log(N))$ operations.

Efficient Implementation of Algorithm AT^2.

- Due to the representation

$$e_\delta(y_1, y_2) = e_\delta(y_1) + e_\delta(y_2), \qquad \text{for } [y_1, y_2] \notin \mathcal{D}(Y),$$

  the maintenance of significances $\{e_\delta(y_1, y_2) : \{y_1, y_2\} \subset Y\}$ can be reduced to maintenance of $\{e_\delta(y_1, y_2) : [y_1, y_2] \in \mathcal{D}(Y)\}$ and $\{e_\delta(y) : y \in Y\}$.

- For efficient implementation of **AT^2** we use two different priority queues,
  - `HeapY` for significances $e_\delta(y)$ of pixels $y \in Y$;
  - `HeapE` for significances $e_\delta(y_1; y_2)$ of edges $[y_1; y_2] \in \mathcal{D}(Y)$.

- Each priority queue, `HeapY` and `HeapE`, contains a least significant element at its head, and is updated after each pixel removal.

- The resulting algorithm has also complexity $\mathcal{O}(N \log N)$.

Further Computational Details.

- We do not remove corner points from X, so that the image domain [X] is invariant during the performance of adaptive thinning.

Uniqueness of Delaunay triangulation.

- Recall that the Delaunay triangulation $\mathcal{D}(Y)$ of $Y \subset X$, is unique, provided that no four points in Y are co-circular.
- Since neither X nor its subsets satisfy this condition, we apply an efficient method, termed *simulation of simplicity* (Edelsbrunner & Mücke, 1990), which assures uniqueness (by using lexicographical order of vertices).
- Unlike in previous perturbation methods, the simulation of simplicity method allows us to work with integer arithmetic rather than with floating point arithmetic.

9 Local Optimization by Exchange

Definition: For any $Y \subset X$, let $Z = X \setminus Y$. A point pair $(y, z) \in Y \times Z$ satisfying

$$\eta((Y \cup z) \setminus y; X) < \eta(Y; X)$$

is said to be `exchangeable`. A subset $Y \subset X$ is said to be `locally optimal` in X, iff there is no exchangeable point pair $(y, z) \in Y \times Z$.

Algorithm (Exchange)

INPUT: $Y \subset X$;

(1) Let $Z = X \setminus Y$;

(2) **WHILE (**Y not locally optimal in X**)**

  (2a) Locate an exchangeable pair $(y, z) \in Y \times Z$;

  (2b) Let $Y = (Y \setminus y) \cup z$ and $Z = (Z \setminus z) \cup y$;

OUTPUT: $Y \subset X$, locally optimal in X.

Characterization of Exchangeable Point Pairs.

Let $Z = X \setminus Y$, for any $Y \subset X$, and recall

$$\eta(Y \setminus y; X) = \eta(Y; X) + e_\delta(y; Y), \qquad \text{for } y \in Y,$$

where $e_\delta(y; Y) = \eta(Y \setminus y; X \cap \mathcal{C}(y; Y)) - \eta(Y; X \cap \mathcal{C}(y; Y))$.

Letting first $Y = Y \cup z$, and then $y = z$, this implies

$$\begin{aligned} \eta((Y \cup z) \setminus y; X) &= \eta(Y \cup z; X) + e_\delta(y; Y \cup z) \\ \eta(Y; X) &= \eta(Y \cup z; X) + e_\delta(z, Y \cup z). \end{aligned}$$

Therefore, $(y, z) \in Y \times Z$ are exchangeable, iff

$$e_\delta(z; Y \cup z) > e_\delta(y; Y \cup z),$$

which simplifies to

$$e_\delta(z; Y \cup z) > e_\delta(y; Y),$$

in case $\mathcal{C}(y; Y) = \mathcal{C}(y; Y \cup z)$, i.e., $[y; z] \notin \mathcal{D}(Y \cup z)$.

Efficient Implementation of Exchange.

- Due to the swapping criterion

$$e_\delta(z; Y \cup z) > e_\delta(y; Y), \qquad \text{for } [y; z] \notin \mathcal{D}(Y \cup z),$$

  the localization of exchangeable point pairs can efficiently be accomplished by maintenance of three different priority queues,

  - `HeapY` for significances $e_\delta(y; Y)$ of pixels $y \in Y$;
  - `HeapZ` for significances $e_\delta(z; Y \cup z)$ of pixels $z \in Z$;
  - `HeapE` for significances $\sigma(y, z) = e_\delta(z; Y \cup z) - e_\delta(y; Y \cup z)$ of edges $[y; z] \in \mathcal{D}(Y \cup z)$.

- The priority queue `HeapY` contains a least significant element at its head; the head of `HeapZ` and `HeapE` contains a most significant element.
- Each of the three priority queues is updated after each pixel exchange.
- The resulting complexity for *one* pixel exchange is $\mathcal{O}(\log N)$.

10 Image Compression

- Our compression method replaces the image I by its linear spline approximation $L^*(Y, I)$, where $Y \subset X$ are the significant pixels.
- In order to code $L^*(Y, I)$, we code the information

$$\{(y, I^*(y)) : y \in Y\}.$$

Quantization.

- Apply uniform quantization to the *optimal* luminances $I^*(y) = L^*(Y, I)(y)$,
- so obtain quantized symbols $\{Q(I^*(y)) : y \in Y\}$,
- corresponding to quantized luminances $\{\tilde{I}(y)) : y \in Y\}$.

11 Theoretical Coding Costs

OBSERVE! Due to the uniqueness of the Delaunay triangulation, no connectivity coding is required!

- We are only concerned with coding the elements of the set

$$\{(y, Q(I^*(y))) : y \in Y\} \in \mathcal{I}_n^s,$$

where, with $n = |Y|$,

$$\mathcal{I}_n^s = \left\{\{0, 1, \ldots, 2^s - 1\}^Z : Z \subset X \text{ and } |Z| = n\right\}.$$

- The number of elements in $\mathcal{I}_n^s$ is $\binom{|X|}{n} \times 2^{s \times n}$.
- If we assume that every element of $\mathcal{I}_n^s$ has the same probability of occurrence, then the theoretical coding cost is

$$\log_2 \left(\binom{|X|}{n} \right) + s \times n.$$

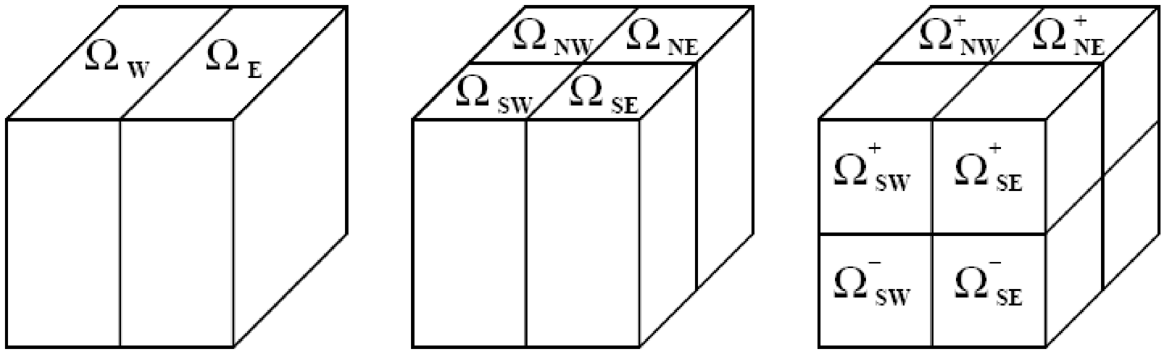
12 Scattered Data Coding

OBSERVE! We can reduce the theoretical coding costs by taking advantage of the *geometric* structure of the image as follows.

The elements of $\{(i, j, Q(I^*(i, j))) : (i, j) \in Y\}$ are coded by decomposing their bounding cell

$$\Omega = [0..2^p - 1] \times [0..2^q - 1] \times [0..2^s - 1]$$

recursively, where $[0..2^s - 1]$ is the range for the quantized symbols.



Splitting of the cell Ω into eight subcells in three stages.

(1) Coding of Scattered Pixels.

- Coding of pixels in Y relies on a recursive splitting of the pixel domain $\Omega = [X]$.
- For the sake of simplicity, let us assume that Ω is a square domain of the form $\Omega = [0, 2^q - 1] \times [0, 2^q - 1]$.
- In the splitting, a square subdomain $\omega \subset \Omega$ (initially $\omega = \Omega$) is split horizontally into two rectangular subdomains of equal size. A rectangular subdomain is split vertically into two square subdomains of equal size.
- The splitting terminates at subdomains which are either *empty*, i.e., not containing any pixel from Y, or *atomic*, i.e., of size 1×1.

(1) Coding of Scattered Pixels.

- This recursive splitting can be represented by a binary tree, whose nodes correspond to the subdomains. The root of the tree corresponds to Ω, and its leaves correspond to empty or atomic subdomains.

- In each node of the tree, with a corresponding subdomain ω, we store the number $|\omega|$ of pixels from Y contained in ω, i.e., $|\omega| = |Y \cap \omega|$.

- Note that for a parent node ω, and its two children nodes, ω_1 and ω_2, we have the relation $|\omega| = |\omega_1| + |\omega_2|$. This relation allows a non-redundant representation of the binary tree.

- The bitstream, representing the tree, is constructed by a Huffman code.

(2) Coding of Quantized Symbols.

- To code the quantized symbols in Q_Y, we first split the image domain Ω into a small number of square subdomains of equal size.
- For each subdomain, the pixels from Y contained in it are ordered linearly, such that close pixels in the image domain are close in this ordering.
- The quantized symbol of any pixel in this ordering is coded relative to the quantized symbol of its predecessor, except for that of the first pixel.
- The coding is done by using a Huffman code.

13 Image Reconstruction at the Decoder

Reconstruction of the compressed image from information

$$\{(y, Q(I^*(y))) : y \in Y\}$$

in four steps:

(1) Compute Delaunay triangulation $\mathcal{D}(Y)$ of Y;

(2) Construct unique linear spline $L(Y, \tilde{I}) \in \mathcal{S}_Y$ satisfying

$$L(Y, \tilde{I})(y) = \tilde{I}(y), \qquad \text{for all } y \in Y,$$

from quantized luminance values $\{\tilde{I}(y) : y \in Y\}$;

(3) Obtain reconstructed image by

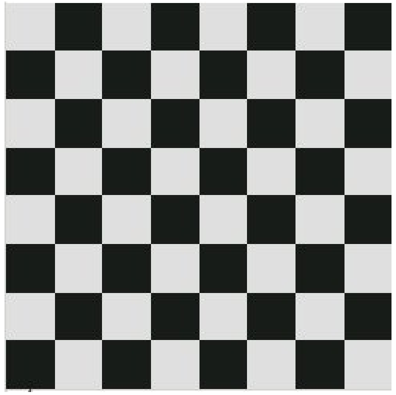
$$\tilde{I} = \{(x, L(Y, \tilde{I})(x)) : x \in X\}.$$

14 First Comparisons with JPEG2000

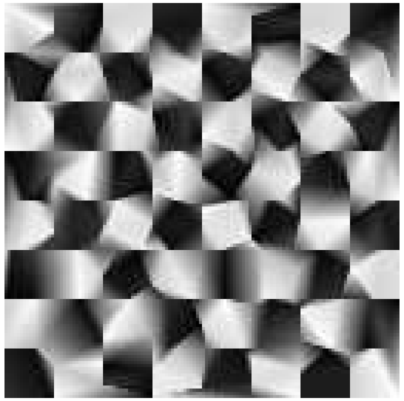
Preliminary Remarks.

- We compare the performance of our compression method **AT^2** with that of EBCOT, which is the basic algorithm in JPEG2000.
- In each comparison, the compression rate, in *bits per pixel* (bpp), is fixed.
- The quality of the resulting reconstructions is measured by their PSNR values, and by their visual quality.

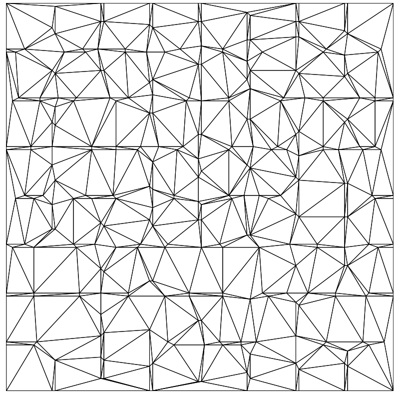
Geometric Test Image `Chessboard`. AT versus AT^2.



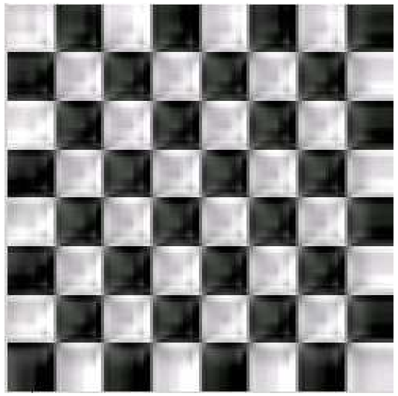
Original Image.



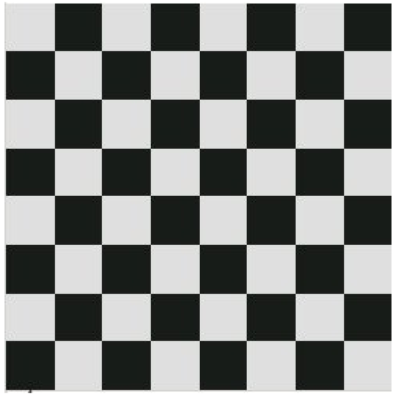
AT



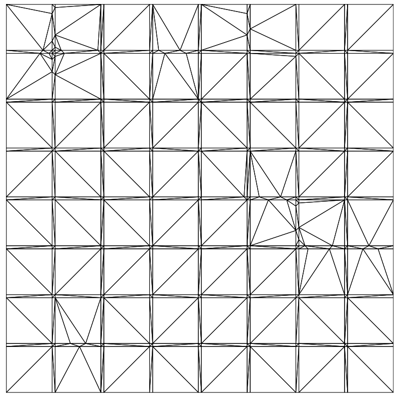
Delaunay triangulation.



JPEG2000

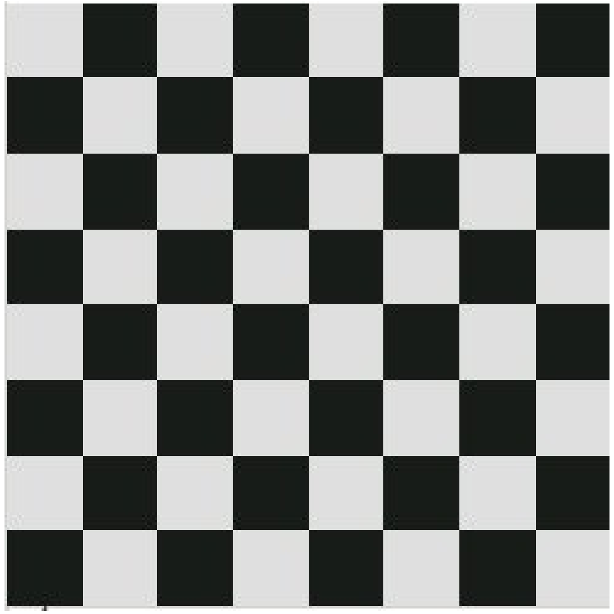


AT^2

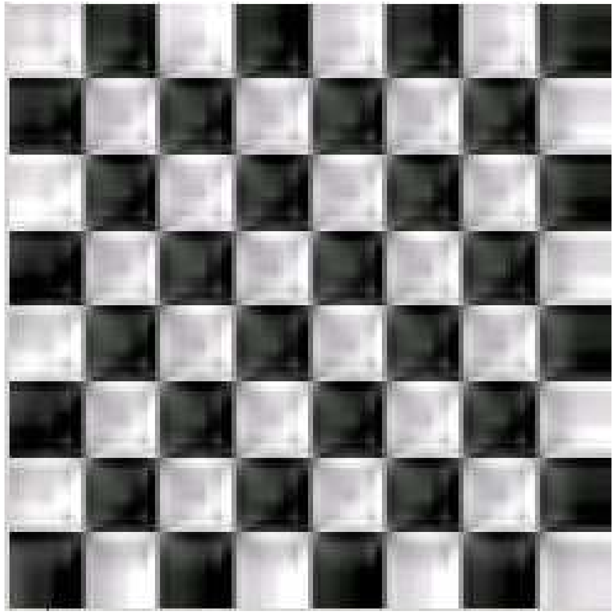


Delaunay triangulation.

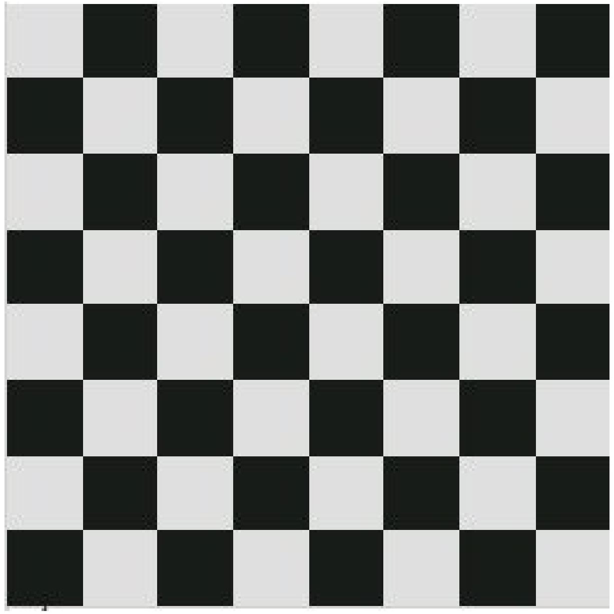
Geometric Test Image `Chessboard`.



Original Image
`Chessboard`
of size (128×128).

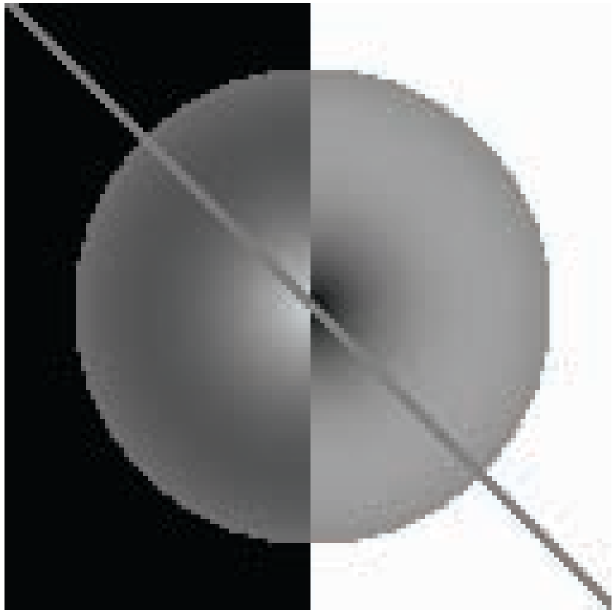


Reconstruction by
JPEG2000 at 0.23 bpp
PSNR 18.68 db.

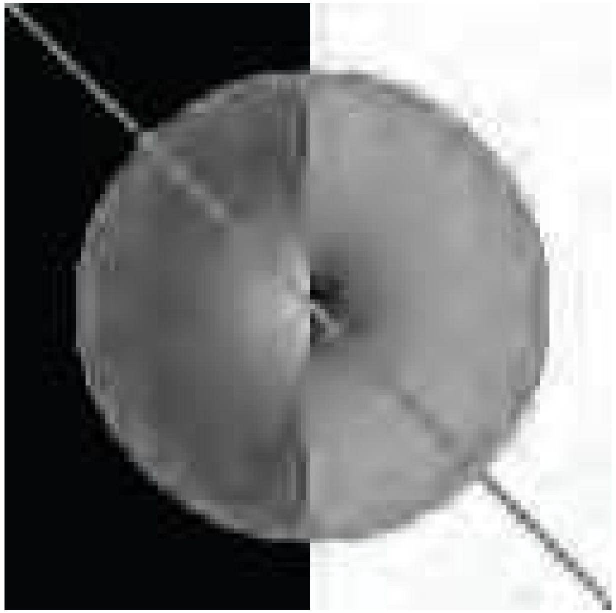


Reconstruction by
AT^2 at 0.23 bpp
PSNR 45.15 db.

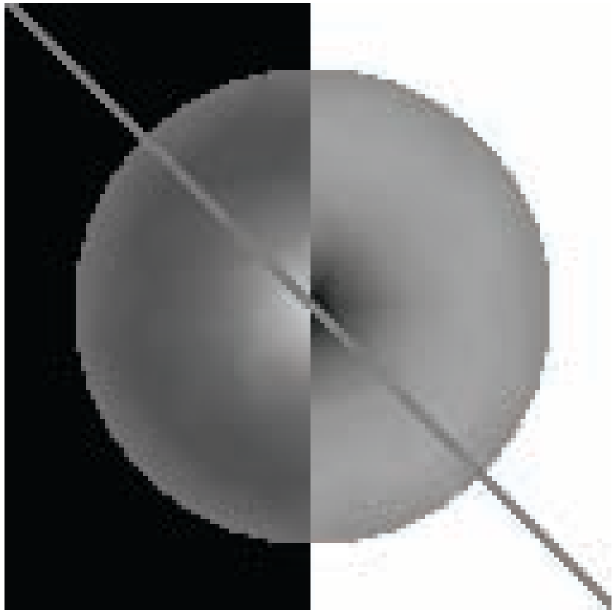
Geometric Real Image Reflex.



Original Image
Reflex
of size (128×128).



Reconstruction by
JPEG2000 at 0.251 bpp
PSNR 28.74 db.



Reconstruction by
AT^2 **at 0.251** bpp
PSNR 42.86 db.

15 More Recent Comparisons with JPEG2000

Current Version (AT2009):

- L. Demaret, A. Iske, W. Khachabi (2009)
  Contextual image compression from adaptive sparse data representations.
  In: *Signal Processing with Adaptive Sparse Structured Representations*.
  Workshop Proceedings, Saint-Malo (France), 6.-9. April 2009.

Previous Version (AT2006):

- L. Demaret, A. Iske (2006)
  Adaptive image approximation by linear splines over locally optimal Delaunay triangulations.
  IEEE Signal Processing Letters 13(5), 281-284.

- L. Demaret, N. Dyn, A. Iske (2006)
  Image compression by linear splines over adaptive triangulations.
  Signal Processing **86**(7), July 2006, 1604–1616.

Comparison between JPEG2000 and AT2009.

Original Image

`Cameraman`

of size (256×256).

Reconstruction by

JPEG2000 at 3.247 kB

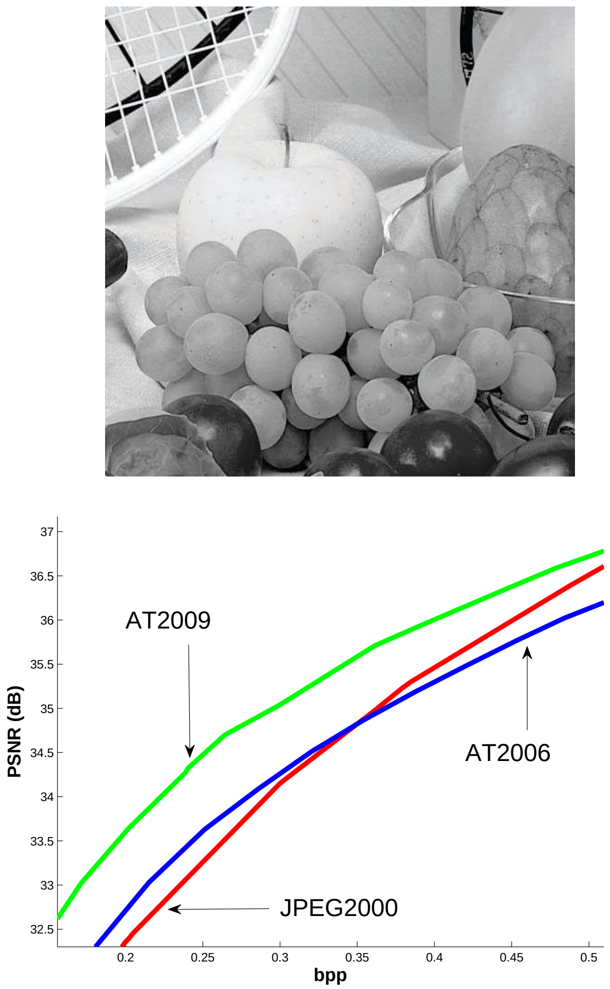
PSNR 29.84 db.

Reconstruction by

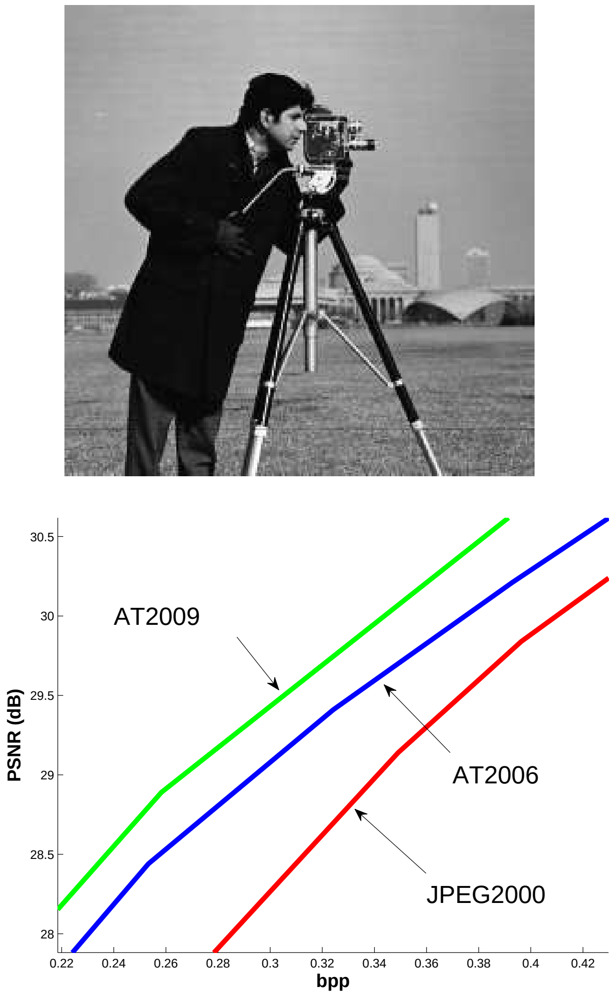
AT2009 at 3.233 kB

PSNR 30.66 db.

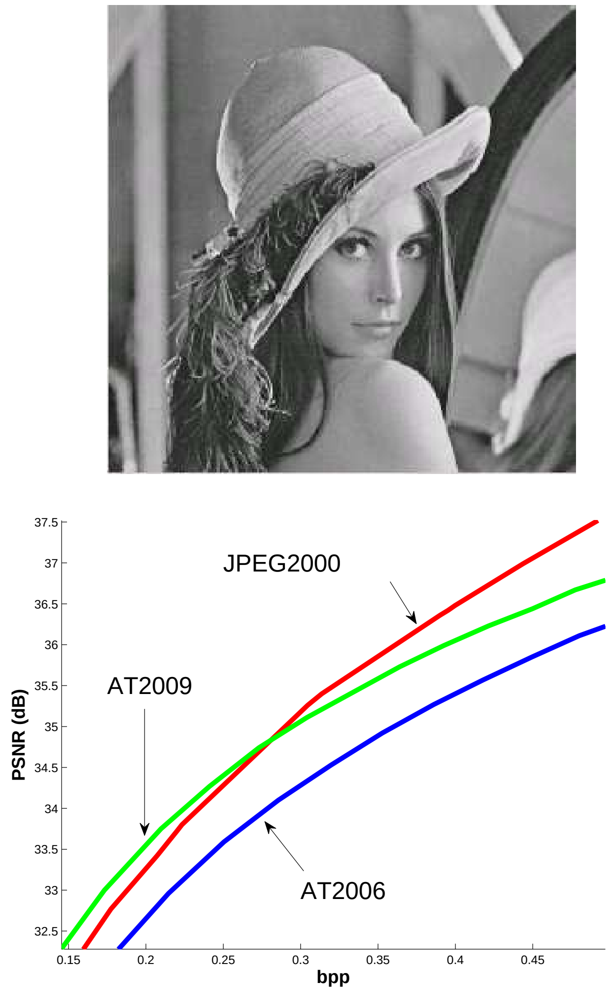
Rate-Distortion Curves for JPEG2000 and AT.



Fruits



Cameraman



Lena

Asymptotic Behaviour of N-term Approximations.

Theorem (BIRMAN & SOLOMJAK 1967)**:** Let $\alpha \in (0, 2]$ and $p \geq 1$ satisfy $\alpha > 2/p - 1$. Then, for any $f \in W^{\alpha,p}([0,1]^2)$ we have

$$E_N(f) = \mathcal{O}(N^{-\alpha}) \qquad \text{for } N \to \infty$$

where

$$E_N(f) = \inf \left\{ \|f - \hat{f}(\mathcal{Q}_N)\|^2_{L^2([0,1]^2)} : \mathcal{Q}_N \in \mathfrak{Q} \text{ with } |\mathcal{Q}_N| = N \right\}. \qquad \blacksquare$$

Corollary (DEMARET & I. 2010)**:** Let $\alpha \in (0, 2]$ and $p \geq 1$ satisfy $\alpha > 2/p - 1$. Then, for any $f \in W^{\alpha,p}([0,1]^2)$ we have

$$E_N(f) = \mathcal{O}(N^{-\alpha}) \qquad \text{for } N \to \infty$$

where

$$E_N(f) = \inf \left\{ \|f - \hat{f}(\mathcal{D}_N)\|^2_{L^2([0,1]^2)} : \mathcal{D}_N \in \mathfrak{D} \text{ with } |\mathcal{D}_N| = N \right\}. \qquad \blacksquare$$

Video Compression: Test Case Suzie.

Video Compression: Preliminary Remarks.

- Natural videos are composed of a superposition of moving objects ...
- ... usually resulting from anisotropic motions;
- a video may be regarded as a sequence of consecutive natural still images ...
- ... or — a video may be regarded as a 3d scalar field;
- it is desirable to work with sparse representations of video data;
- ...
- **Adaptive Thinning (AT)** extracts significant video pixels ...
- ... to obtain a sparse representation of the video ...
- ... relying on linear splines over anisotropic tetrahedralizations.

Representation of Video Data.

- A digital video V is a rectangular 3d grid of pixels, X.
- Each pixel $x \in X$ bears a greyscale luminance $V(x)$.
- We regard the video as a trivariate function,

$$V : [X] \longrightarrow \{0, 1, \ldots, 2^r - 1\}$$

where the convex hull $[X]$ of X is the video domain.

INPUT: The video is given by its restriction to the pixels in X,

$$V\big|_X = \{(x, V(x)) : x \in X\}.$$

GOAL: Approximation of V from discrete data $V\big|_X$.

Linear Splines over Tetrahedralizations.

- Given any tetrahedralizations $\mathcal{T}(Y)$ of Y, we denote by

$$\mathcal{S}_Y = \left\{ s : s \in C([Y]) \text{ and } s\big|_T \text{ linear for all } T \in \mathcal{T}(Y) \right\},$$

  the spline space containing all continuous functions over $[Y]$ whose restriction to any tetrahedron in $\mathcal{T}(Y)$ is linear.

- Any element in $\mathcal{S}_Y$ is referred to as a linear spline over $\mathcal{T}(Y)$.

- For given function values $\{V(y) : y \in Y\}$, there is a unique linear spline, $L(Y, V) \in \mathcal{S}_Y$, which interpolates V at the points of Y, i.e.,

$$L(Y, V)(y) = V(y), \qquad \text{for all } y \in Y.$$

Basic Features of Delaunay Tetrahedralizations.

- **Uniqueness.**

Delaunay tetrahedralization $\mathcal{D}(X)$ is *unique*,
if no five points in X are co-spherical.

- **Complexity.**

For any point set X, its Delaunay tetrahedralization $\mathcal{D}(X)$
can be computed in $\mathcal{O}(N \log N)$ steps, where $N = |X|$.

- **Local Updating.**

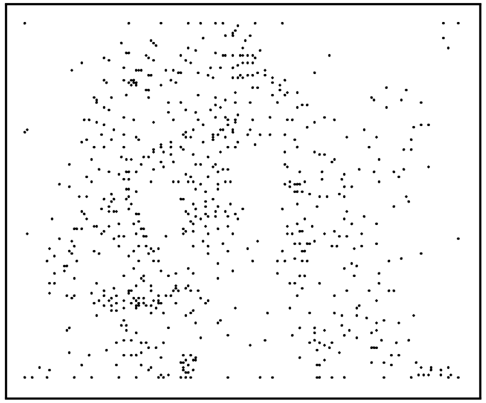
For any X and $x \in X$, the Delaunay tetrahedralization $\mathcal{D}(X \setminus x)$ of the point set $X \setminus x$ can be computed from $\mathcal{D}(X)$ by re-tetrahedralization of the *cell* $\mathcal{C}(x)$ of x.

Removal of the node x and re-tetrahedralization of its cell $\mathcal{C}(x)$.

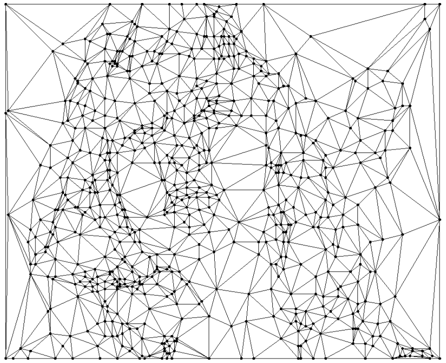
Numerical Simulation for Test Case `Suzie`.

Test Case `Suzie`: Frame 0000.

Original Frame `Suzie`.



708 significant pixels.

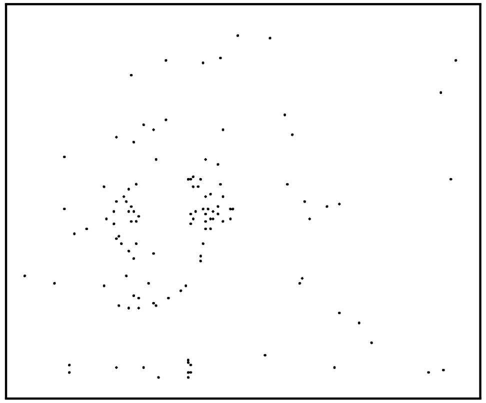


Delaunay tetrahedralization.

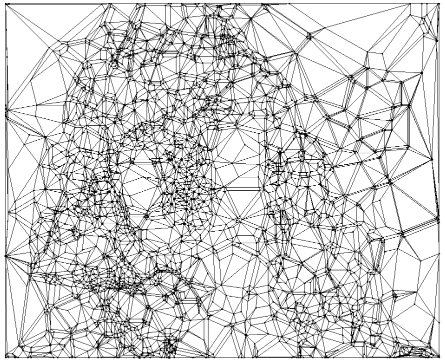
Reconstruction by AT at 34.58 dB.

Test Case `Suzie`: Frame 0001.

Original Frame `Suzie`.



118 significant pixels.

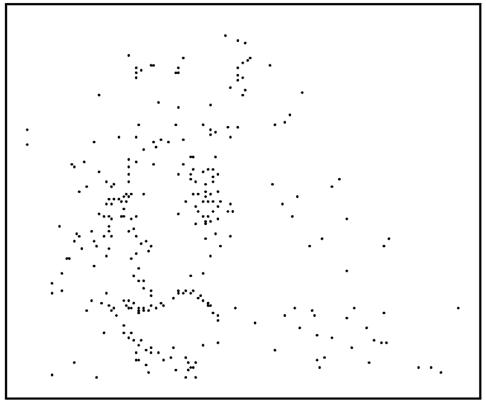


Delaunay tetrahedralization.

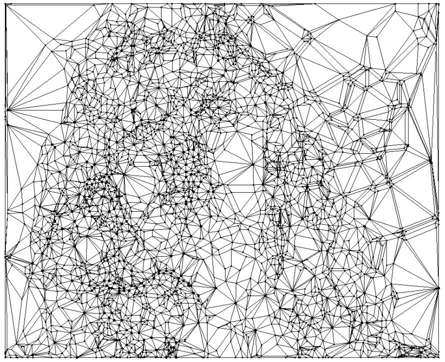
Reconstruction by AT at 35.15 dB.

Test Case `Suzie`: Frame 0002.

Original Frame `Suzie`.



287 significant pixels.

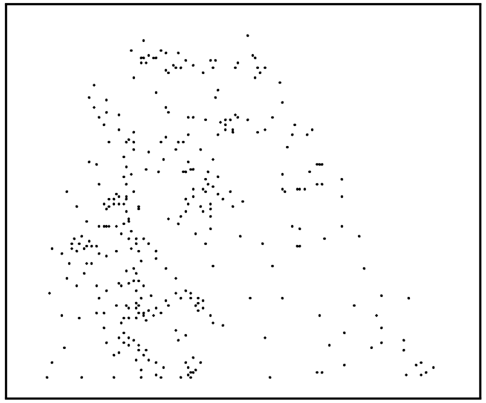


Delaunay tetrahedralization.

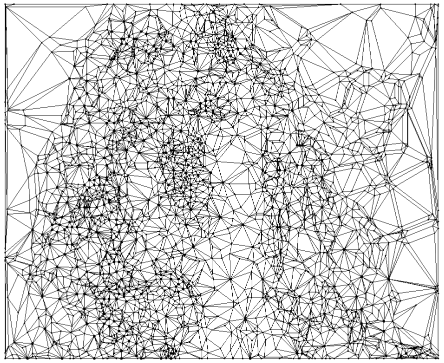
Reconstruction by AT at 35.18 dB.

Test Case `Suzie`: Frame 0003.

Original Frame `Suzie`.



338 significant pixels.

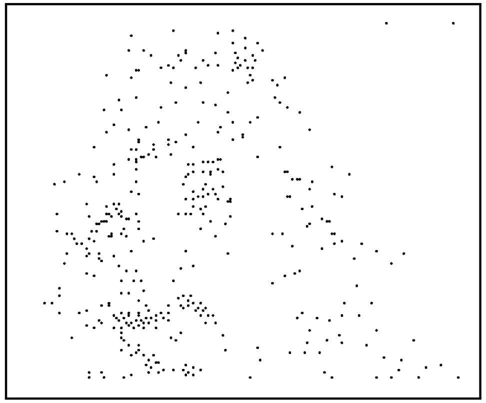


Delaunay tetrahedralization.

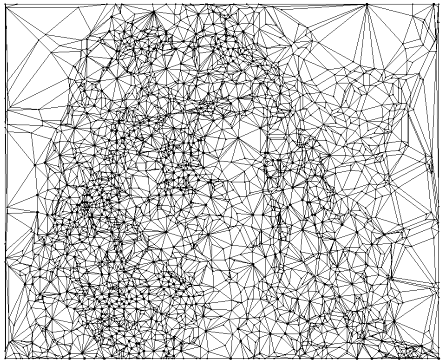
Reconstruction by AT at 34.91 dB.

Test Case `Suzie`: Frame 0004.

Original Frame `Suzie`.



398 significant pixels.

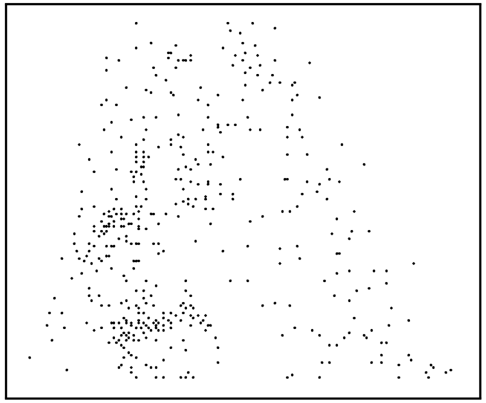


Delaunay tetrahedralization.

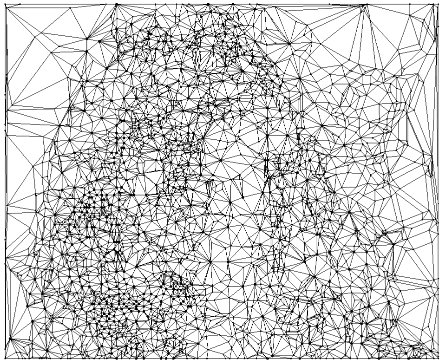
Reconstruction by AT at 34.98 dB.

Test Case `Suzie`: Frame 0005.

Original Frame `Suzie`.



448 significant pixels.

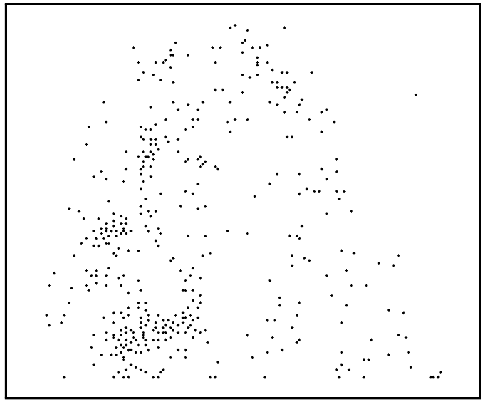


Delaunay tetrahedralization.

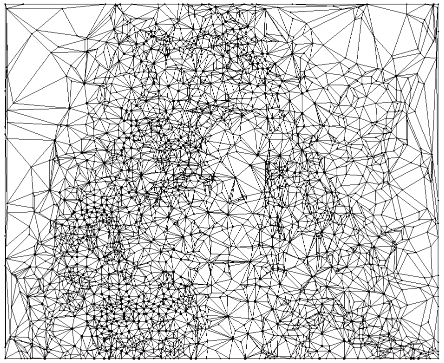
Reconstruction by AT at 34.99 dB.

Test Case Suzie: Frame 0006.

Original Frame Suzie.



424 significant pixels.

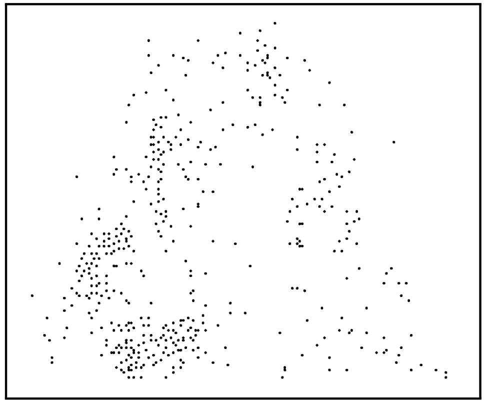


Delaunay tetrahedralization.

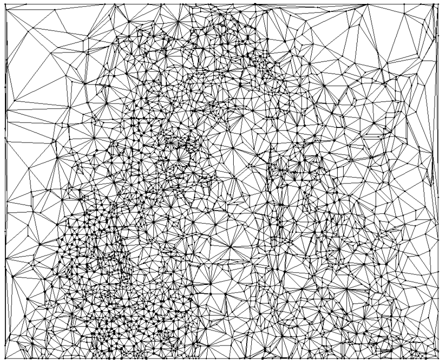
Reconstruction by AT at 34.96 dB.

Test Case `Suzie`: Frame 0007.

Original Frame `Suzie`.



460 significant pixels.

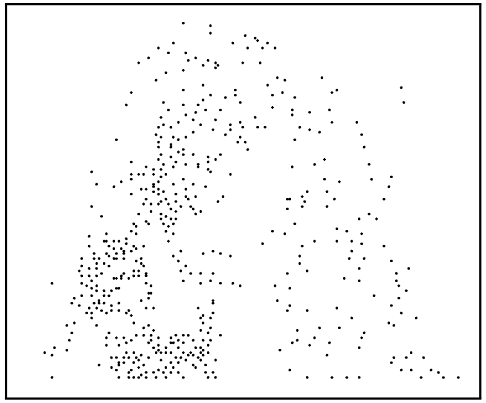


Delaunay tetrahedralization.

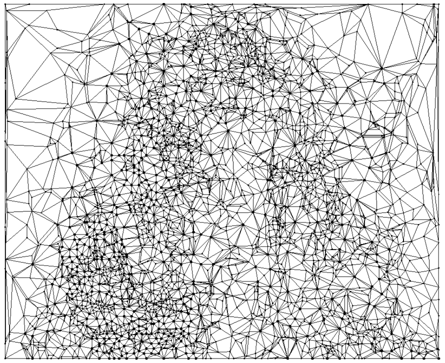
Reconstruction by AT at 34.92 dB.

Test Case Suzie: Frame 0008.

Original Frame Suzie.



534 significant pixels.

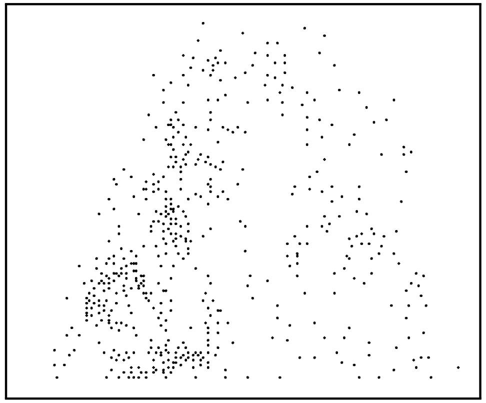


Delaunay tetrahedralization.

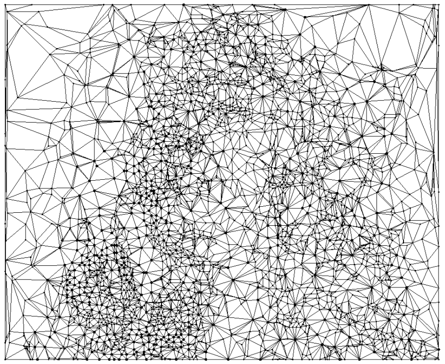
Reconstruction by AT at 35.11 dB.

Test Case `Suzie`: Frame 0009.

Original Frame `Suzie`.



523 significant pixels.

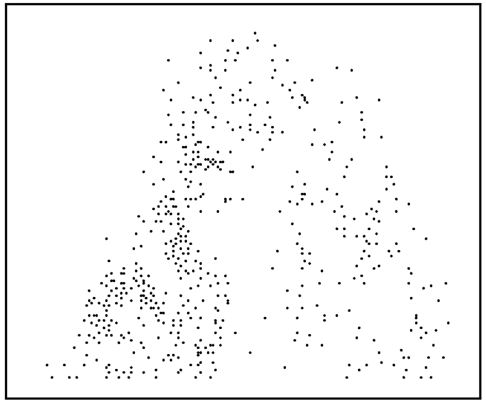


Delaunay tetrahedralization.

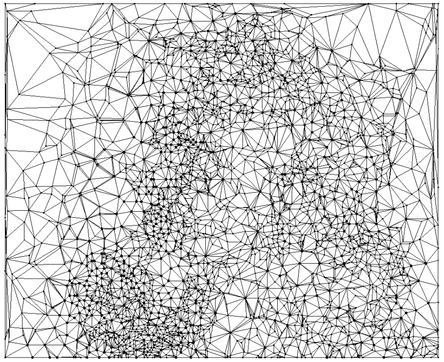
Reconstruction by AT at 34.82 dB.

Test Case `Suzie`: Frame 0010.

Original Frame `Suzie`.



539 significant pixels.

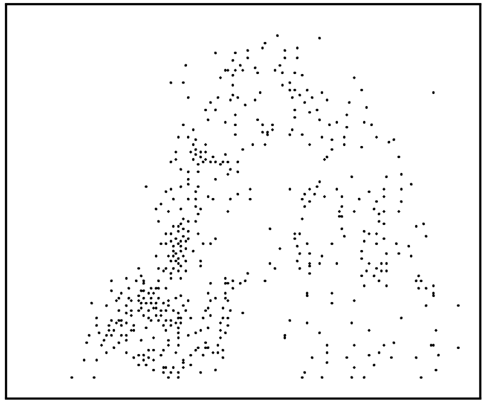


Delaunay tetrahedralization.

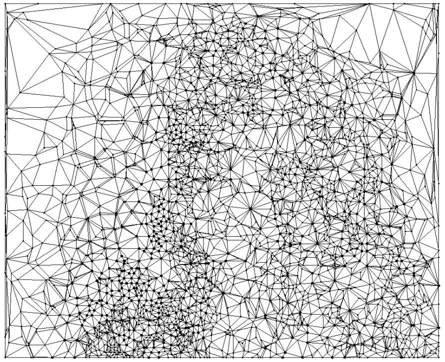
Reconstruction by AT at 34.89 dB.

Test Case Suzie: Frame 0011.

Original Frame Suzie.



534 significant pixels.

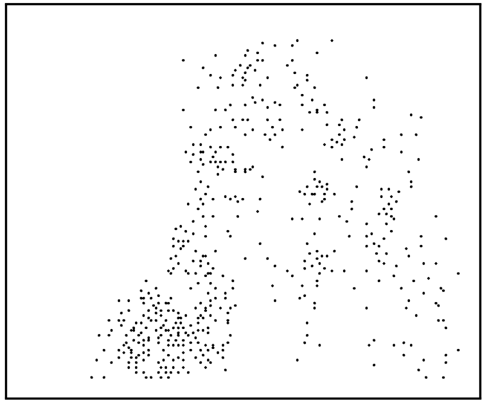


Delaunay tetrahedralization.

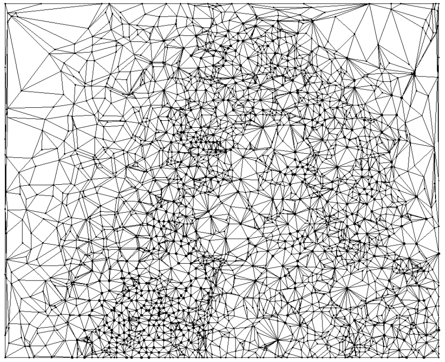
Reconstruction by AT at 34.95 dB.

Test Case `Suzie`: Frame 0012.

Original Frame `Suzie`.



513 significant pixels.

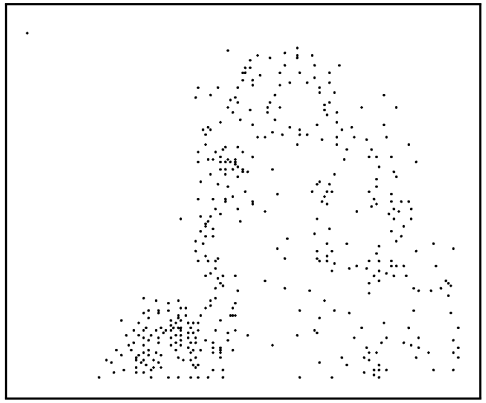


Delaunay tetrahedralization.

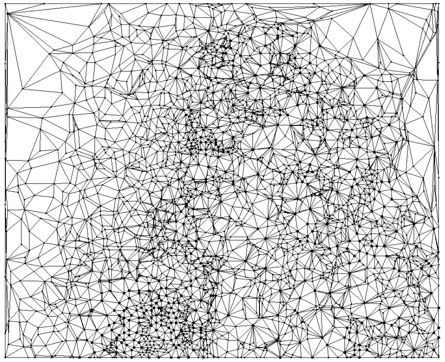
Reconstruction by AT at 35.34 dB.

Test Case `Suzie`: Frame 0013.

Original Frame `Suzie`.



432 significant pixels.



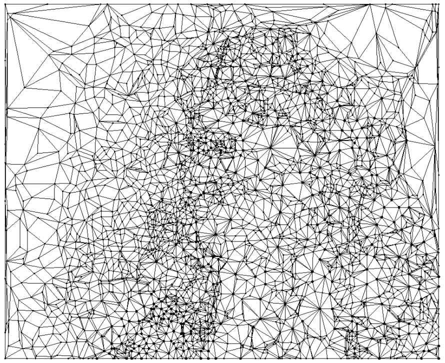
Delaunay tetrahedralization.

Reconstruction by AT at 35.30 dB.

Test Case `Suzie`: Frame 0014.

Original Frame `Suzie`.

364 significant pixels.

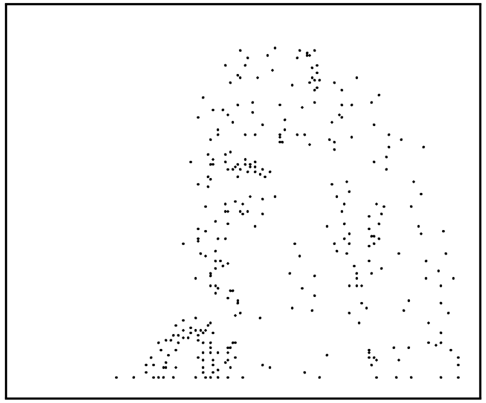


Delaunay tetrahedralization.

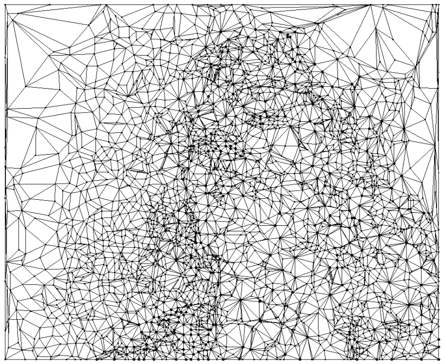
Reconstruction by AT at 35.49 dB.

Test Case `Suzie`: Frame 0015.

Original Frame `Suzie`.



311 significant pixels.



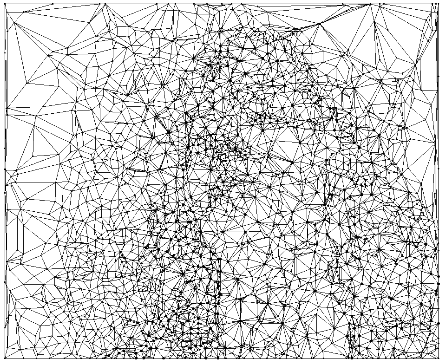
Delaunay tetrahedralization.

Reconstruction by AT at 35.68 dB.

Test Case `Suzie`: Frame 0016.

Original Frame `Suzie`.

285 significant pixels.

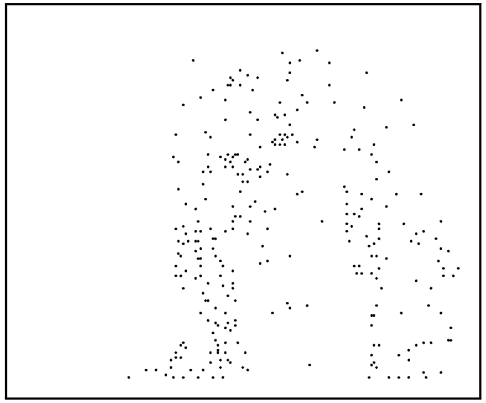


Delaunay tetrahedralization.

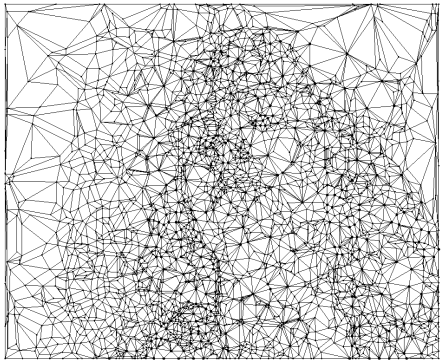
Reconstruction by AT at 35.82 dB.

Test Case Suzie: Frame 0017.

Original Frame Suzie.



293 significant pixels.

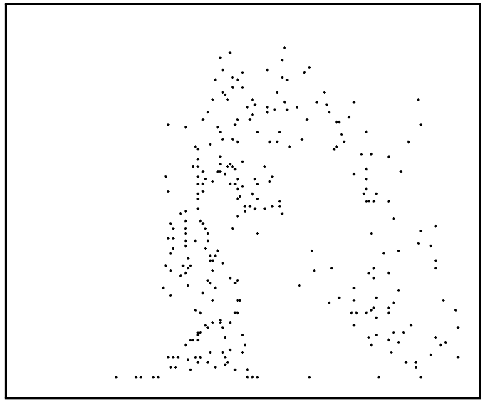


Delaunay tetrahedralization.

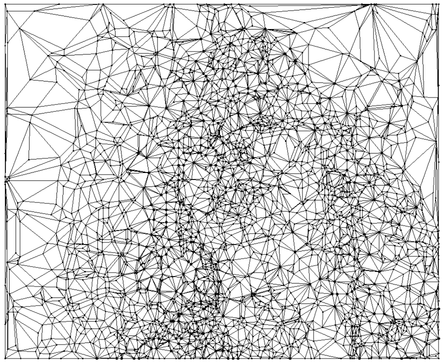
Reconstruction by AT at 36.32 dB.

Test Case `Suzie`: Frame 0018.

Original Frame `Suzie`.



289 significant pixels.

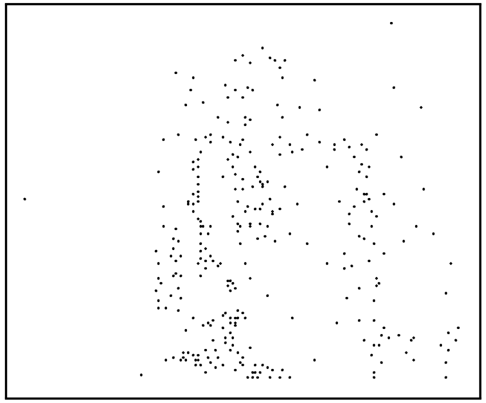


Delaunay tetrahedralization.

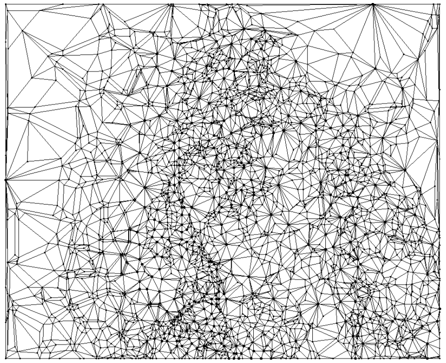
Reconstruction by AT at 36.08 dB.

Test Case `Suzie`: Frame 0019.

Original Frame `Suzie`.



307 significant pixels.

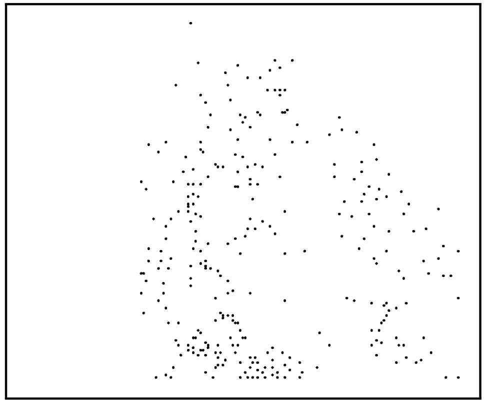


Delaunay tetrahedralization.

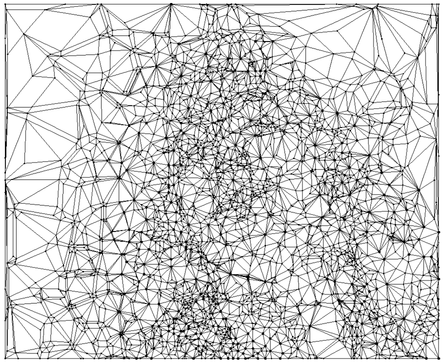
Reconstruction by AT at 36.25 dB.

Test Case Suzie: Frame 0020.

Original Frame Suzie.



292 significant pixels.

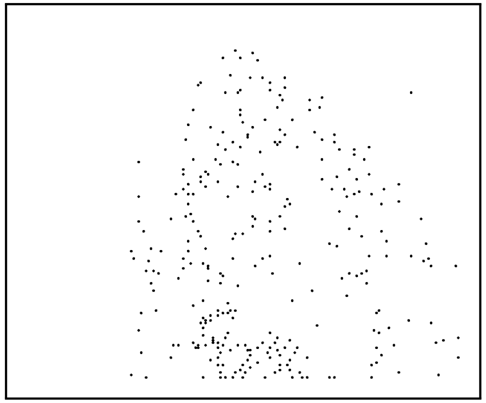


Delaunay tetrahedralization.

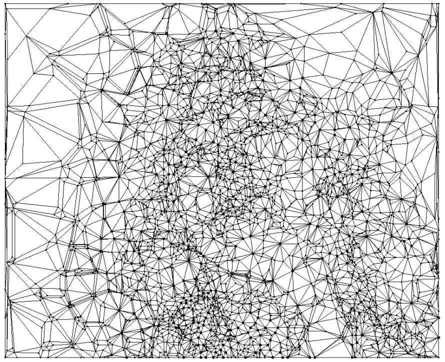
Reconstruction by AT at 36.26 dB.

Test Case `Suzie`: Frame 0021.

Original Frame `Suzie`.



293 significant pixels.

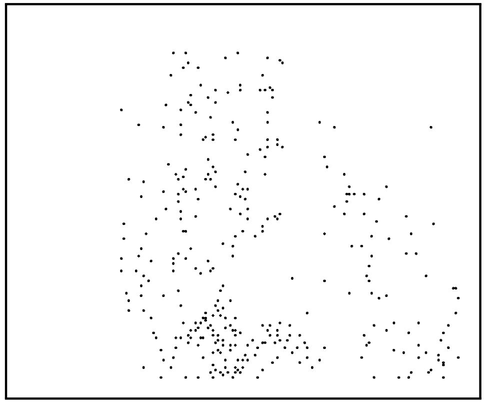


Delaunay tetrahedralization.

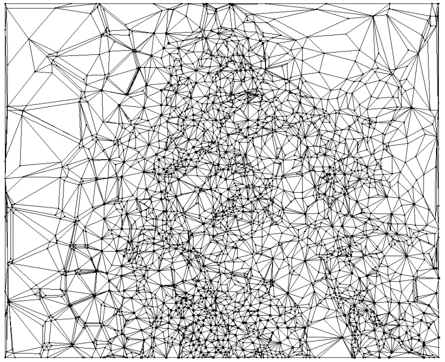
Reconstruction by AT at 36.02 dB.

Test Case `Suzie`: Frame 0022.

Original Frame `Suzie`.



326 significant pixels.

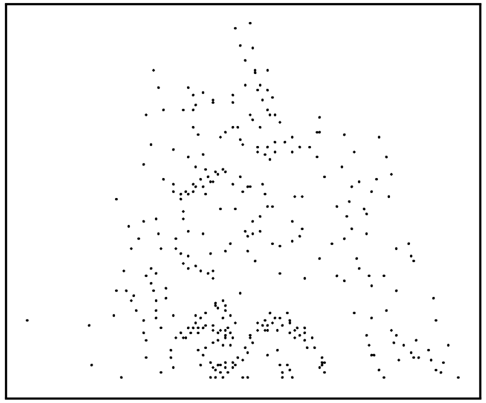


Delaunay tetrahedralization.

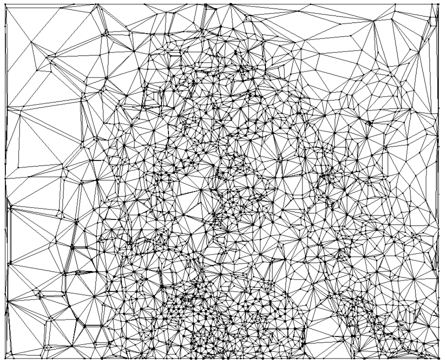
Reconstruction by AT at 36.06 dB.

Test Case `Suzie`: Frame 0023.

Original Frame `Suzie`.



341 significant pixels.

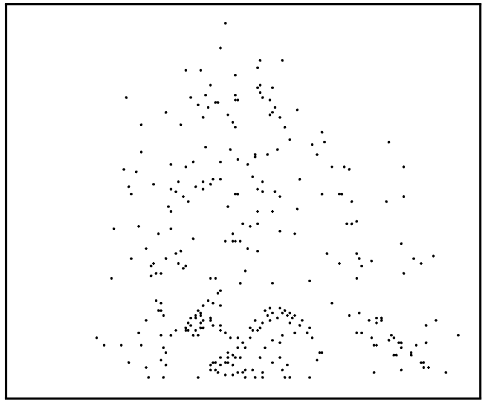


Delaunay tetrahedralization.

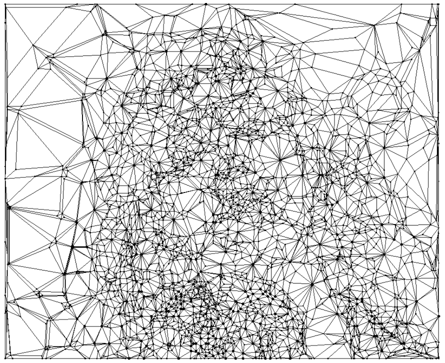
Reconstruction by AT at 36.08 dB.

Test Case `Suzie`: Frame 0024.

Original Frame `Suzie`.



311 significant pixels.

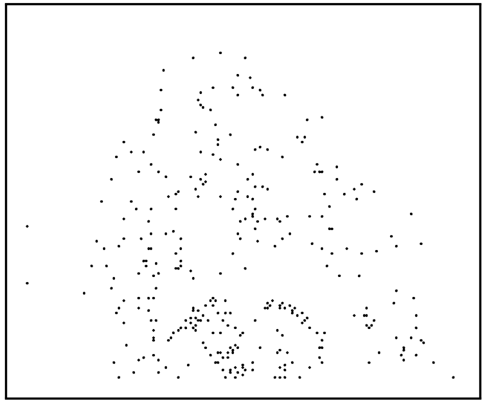


Delaunay tetrahedralization.

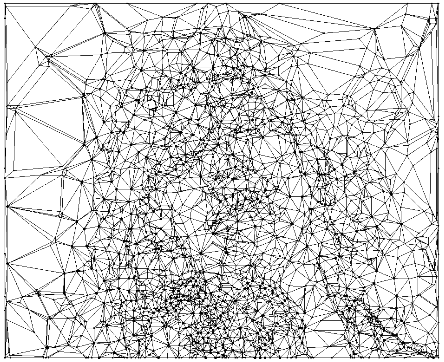
Reconstruction by AT at 36.24 dB.

Test Case `Suzie`: Frame 0025.

Original Frame `Suzie`.



321 significant pixels.

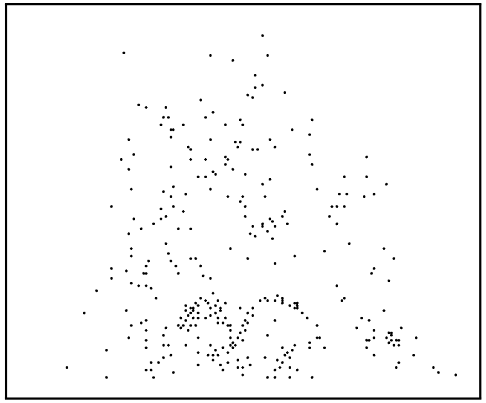


Delaunay tetrahedralization.

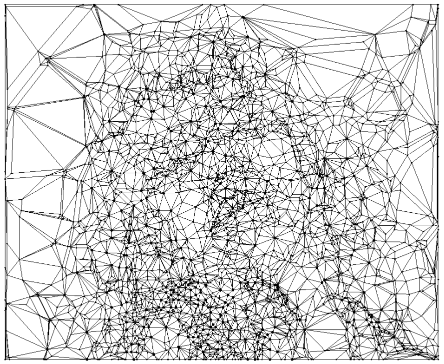
Reconstruction by AT at 36.16 dB.

Test Case `Suzie`: Frame 0026.

Original Frame `Suzie`.



320 significant pixels.

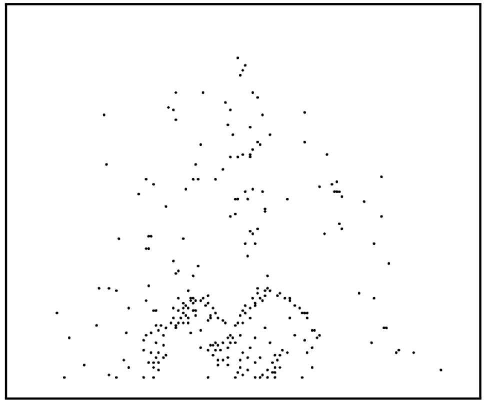


Delaunay tetrahedralization.

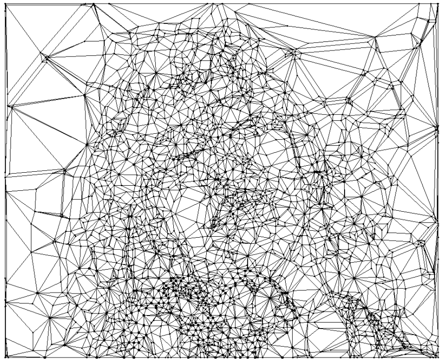
Reconstruction by AT at 35.95 dB.

Test Case Suzie: Frame 0027.

Original Frame Suzie.



273 significant pixels.

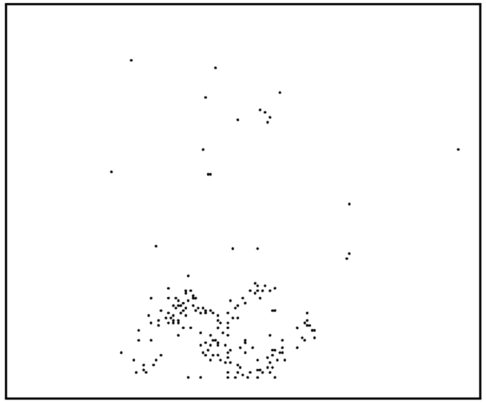


Delaunay tetrahedralization.

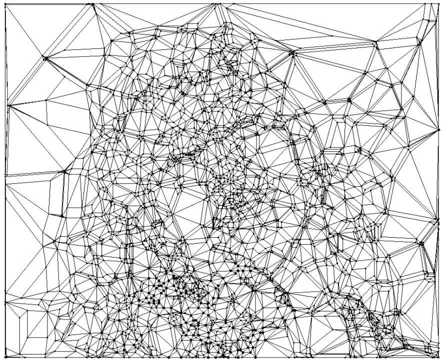
Reconstruction by AT at 35.60 dB.

Test Case Suzie: Frame 0028.

Original Frame Suzie.



179 significant pixels.

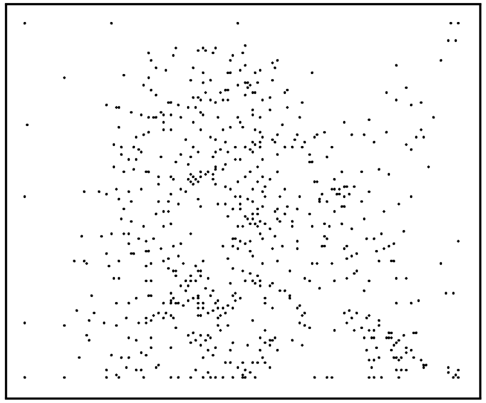


Delaunay tetrahedralization.

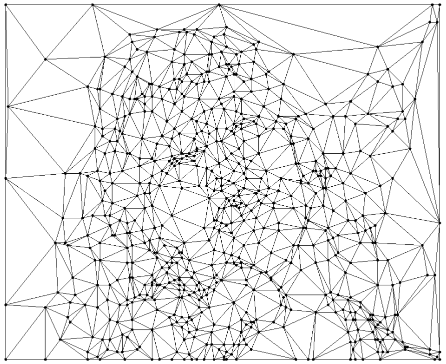
Reconstruction by AT at 35.48 dB.

Test Case Suzie: Frame 0029.

Original Frame Suzie.



669 significant pixels.



Delaunay tetrahedralization.

Reconstruction by AT at 35.00 dB.

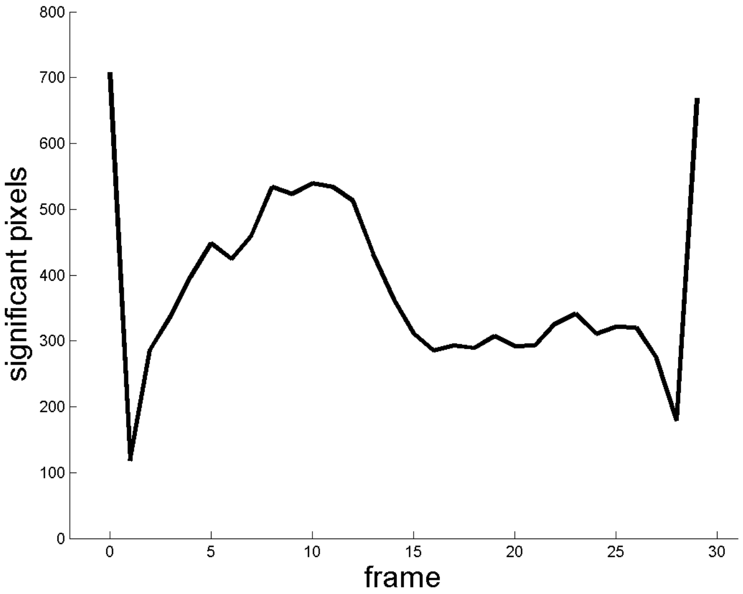
Performance Check: Data Size and Approximation.

Number of significant pixels:

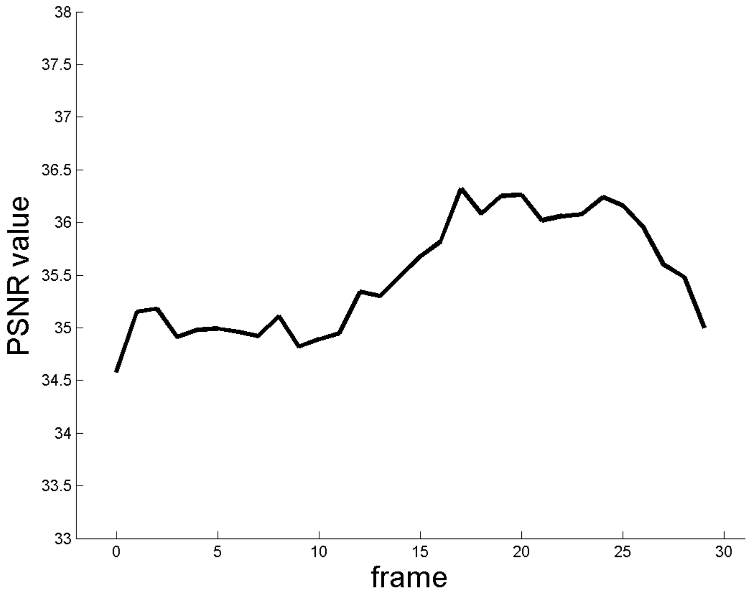
Total: 11,430; minimal: 118; maximal: 708; average: 381 pixels.

PSNR value:

Overall: 35.45 dB; minimal: 34.58 dB; maximal: 36.32 dB; average: 35.49 dB.



Number of significant pixels.



PSNR values.

Relevant Literature.

- L. Demaret and A. Iske (2010) Anisotropic triangulation methods in image approximation. In: *Approximation Algorithms for Complex Systems*, E.H. Georgoulis, A. Iske, and J. Levesley (eds.), Springer, Berlin, 47–68.
- L. Demaret, A. Iske, and W. Khachabi (2010) Sparse representation of video data by adaptive tetrahedralisations. In: *Locally Adaptive Filters in Signal and Image Processing*, L. Florack, R. Duits, G. Jongbloed, M.-C. van Lieshout, L. Davies (eds.), 197–220.
- L. Demaret, A. Iske, and W. Khachabi (2009) Contextual image compression from adaptive sparse data representatons. Workshop Proceedings *Signal Processing with Adaptive Sparse Structured Representations*, 6.-9. April 2009 - Saint-Malo (France).
- L. Demaret and A. Iske (2006) Adaptive image approximation by linear splines over locally optimal Delaunay triangulations. *IEEE Signal Processing Letters* **13**(5), 281–284.
- L. Demaret, N. Dyn, and A. Iske (2006) Image compression by linear splines over adaptive triangulations. *Signal Processing* **86**(7), July 2006, 281–284.
- L. Demaret, N. Dyn, M.S. Floater, and A. Iske (2005) Adaptive thinning for terrain modelling and image compression. *Advances in Multiresolution for Geometric Modelling*, N.A. Dodgson, M.S. Floater, and M.A. Sabin (eds.), Springer, 321–340.