JBEAM: Multi-scale Curve Coding via Beamlets

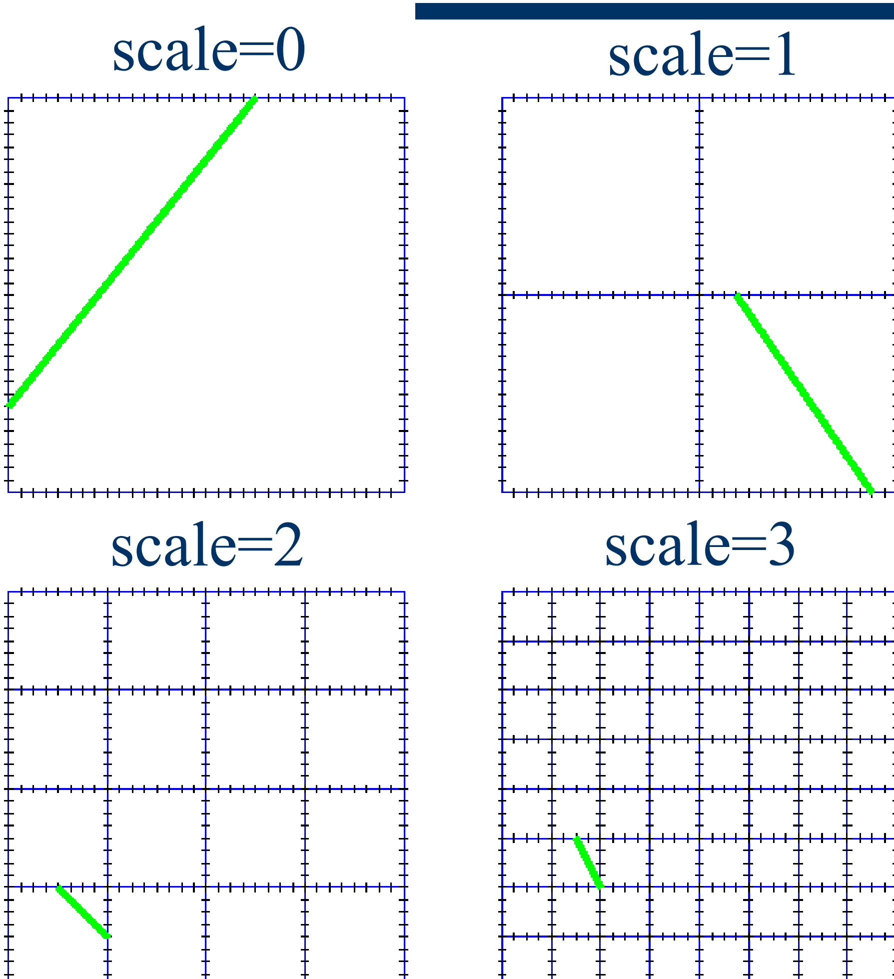
Xiaoming Huo

School of Industrial and Systems Engineering
Georgia Institute of Technology

9 / 23 / 2004

w. Jihong Chen

Beamlets & Beamlet Dictionary



Donoho, Ann. of Stat. 1999

Donoho & Huo, 2001, 2002, etc.

Objective

- Coding linear and curvilinear features
- Combine Beamlet and Zero-tree Coding
- Outperform existing industrial standards?
- Variations/Improvements/Software

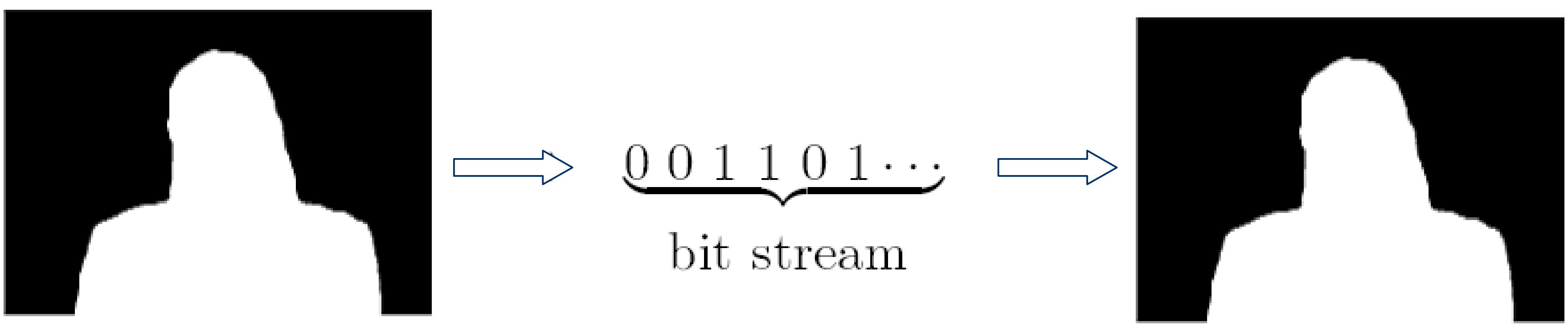
Wavelet ⟷ Image Coding
Beamlet ⟷ Curve Coding

Outline for JBEAM

- I. Motivations
- II Preparation:
  - Coding Single Beamlet
  - Beamlet representation
- III. Beamlet coder (JBEAM)
  - Rate-Distortion Optimized Representation
  - Zero-tree Coding
- IV. Simulation / Discussion
- V. Conclusion

I. Motivation (1) Shape Coding

- Video object-based coding



Recent publications in IEEE Trans. I.P.

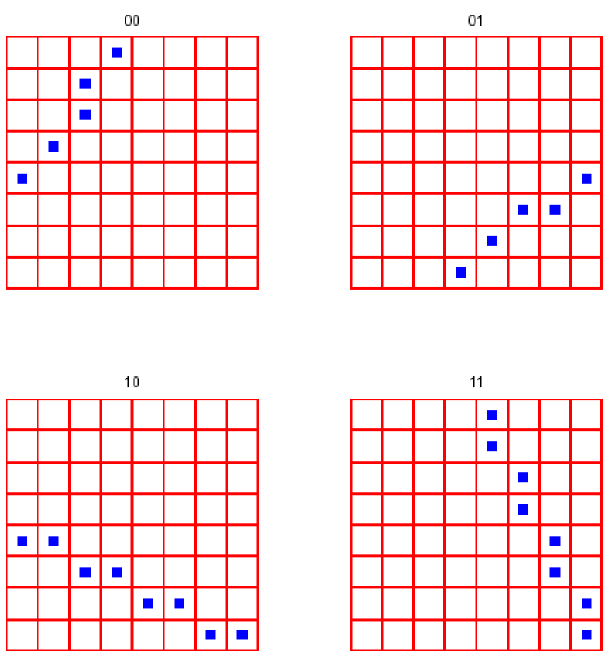
Motivation (2) Line Images

Line drawings

II. Prep.-1: Coding a Single Beamlet

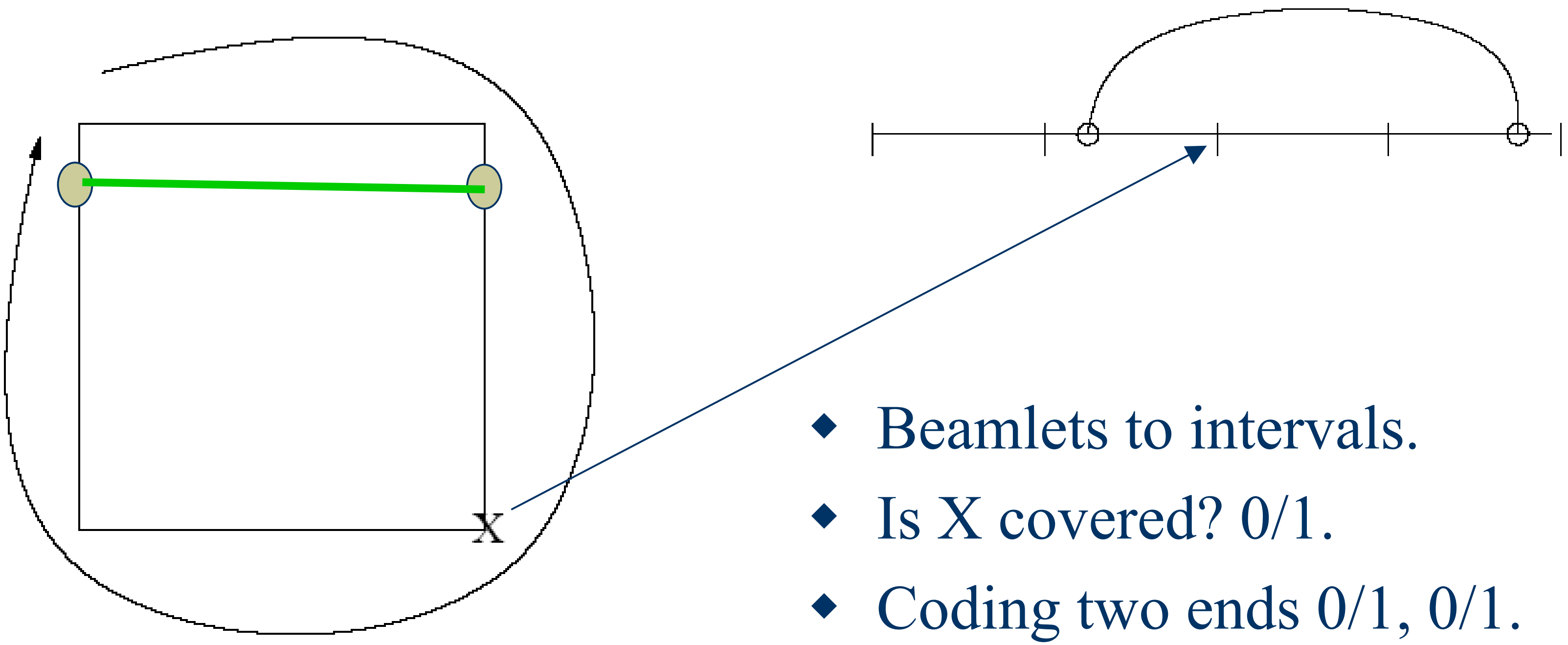
Scale, s	0	1	2	3	4
Size, 2^s	1	2	4	8	16
# beamlets	1	$\binom{4}{2} = 6$	$\binom{12}{2} = 66$	$\binom{28}{2} = 378$	$\binom{60}{2} = 1770$
# bits	(1)	(3)	7	9	11

Number of bits that are required



Coded beamlets with partial bit stream

Coding Scheme for a beamlet

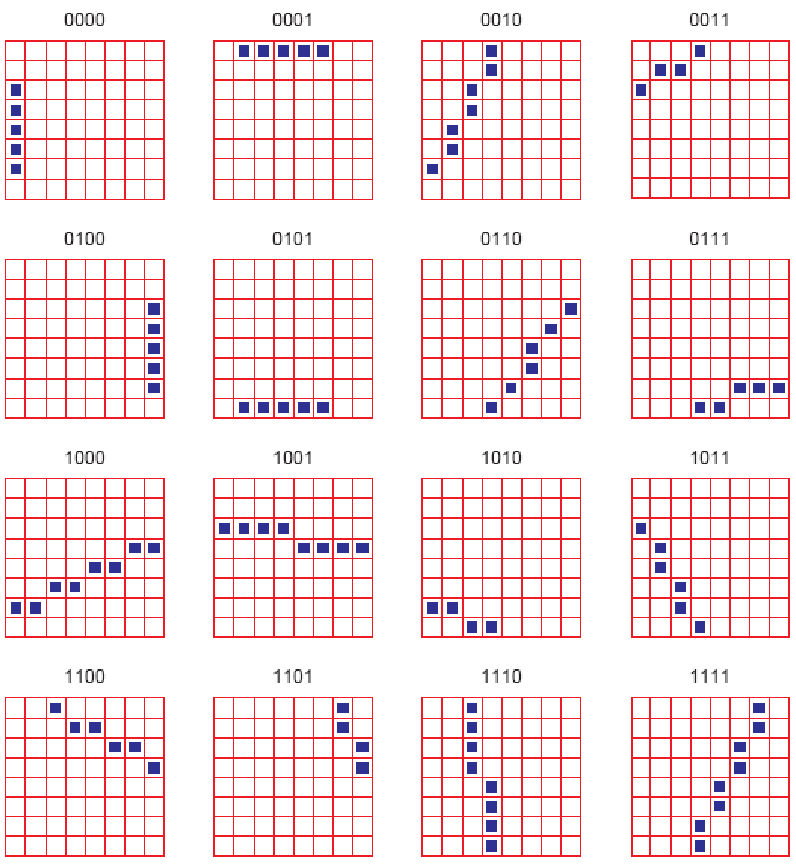


- Beamlets to intervals.
- Is X covered? 0/1.
- Coding two ends 0/1, 0/1.
- Recursion

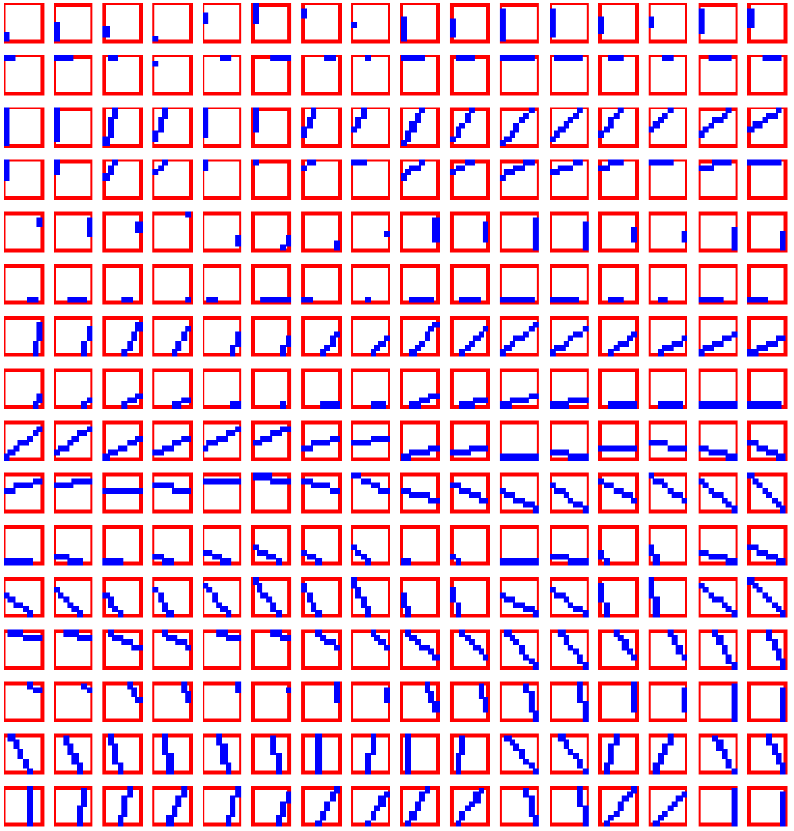
A Progressive Scheme to Code Single Beamlets

1) For a 2^s by 2^s image, it takes $2s + 3$ bits to code EXACTLY.
2) While $2k(k \leq s + 1)$ bits are available, the MAX distortion is 2^{s+1-k}, which is lower bound in IT.
3) Partial bits, partial reconstruction, as in illustration.

Illustrations

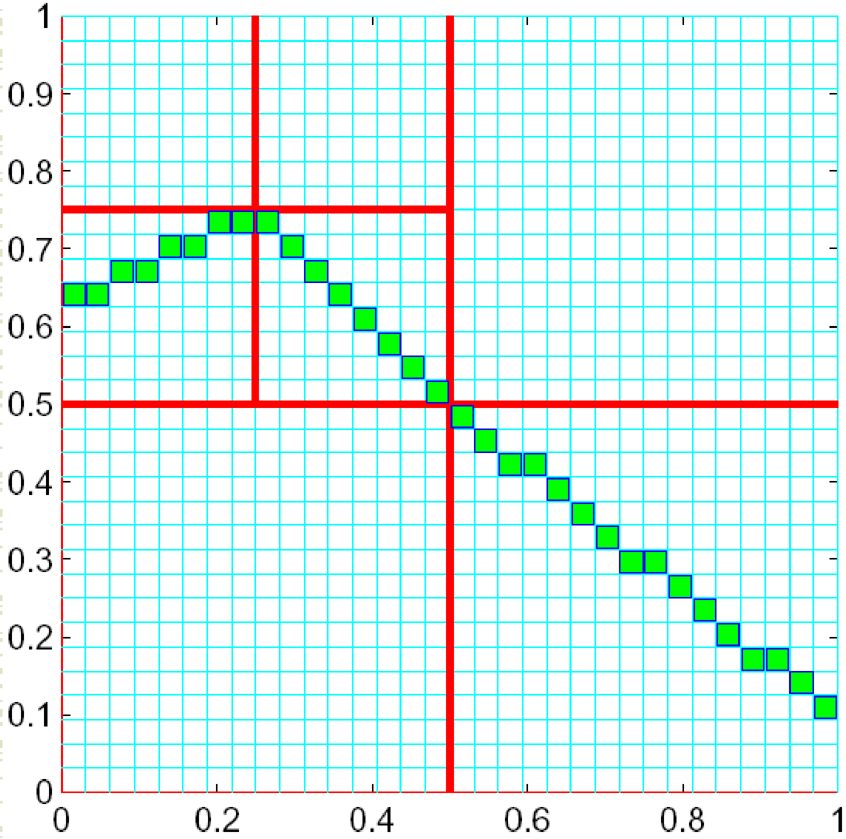


4 bits



8 bits

II. Prep.-2: Beamlet Representation



scale 0	scale 1		scale 2			
			N	N	–	–
Q	Q	N	B_2	B_3	–	–
	N	B_1	–	–	–	–
			–	–	–	–

	1st vertex	2nd vertex	binary represent
B_1	16	45	10011111011
B_2	5	16	100100000
B_3	8	24	101101010

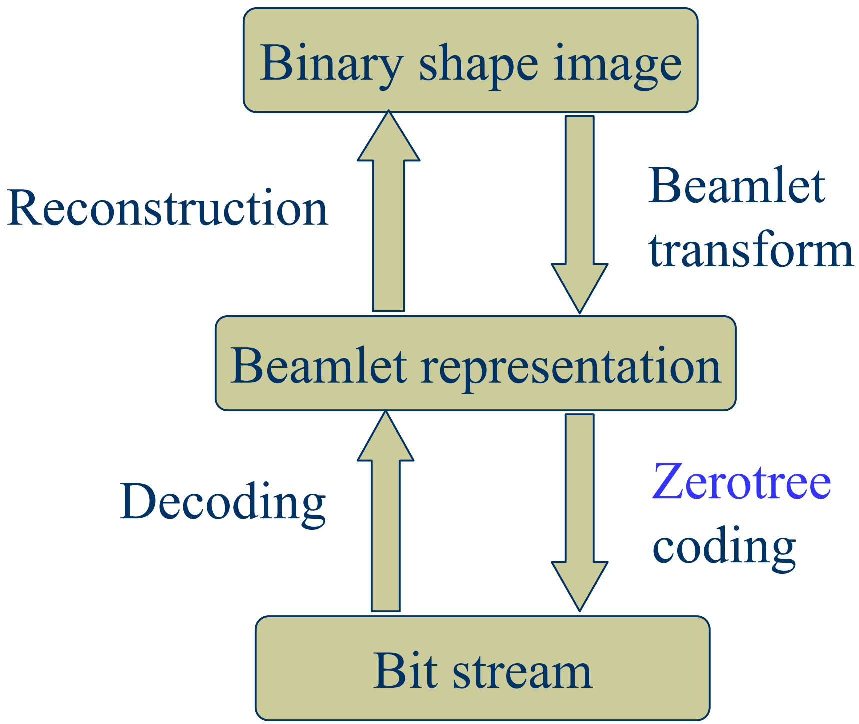
A Re-list of Beamlet Representation

$$
\begin{array}{lcll}
l_0 & \longrightarrow & \mathrm{Q} & \\
l_1 & \longrightarrow & \mathrm{Q} & \\
 & & \mathrm{N} & \\
 & & \mathrm{N} & \\
 & & \mathrm{B}_1 & 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
l_2 & \longrightarrow & \mathrm{N} & \\
 & & \mathrm{N} & \\
 & & \mathrm{B}_2 & 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\
 & & \mathrm{B}_3 & 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0
\end{array}
$$

Bits, locations of beamlets

Geometry

III. JBEAM
Overview of Beamlet Coder



Entropy Coding

Zerotree algorithm (Shapiro,93; Said & Pearlman)

- The bit-stream includes information in order of importance: Msb first and Lsb last.

Rate-Distortion-Optimized Representation

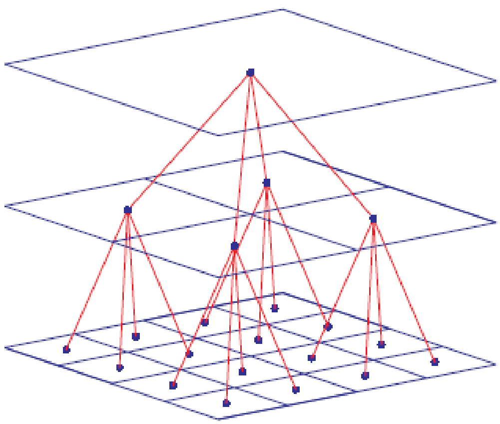
$$\min_{\mathcal{P}} R(\mathcal{P}), \qquad \text{subject to } D(y, \mathcal{P}) \leq \overline{D},$$

- Use Lagrange multiplier to solve

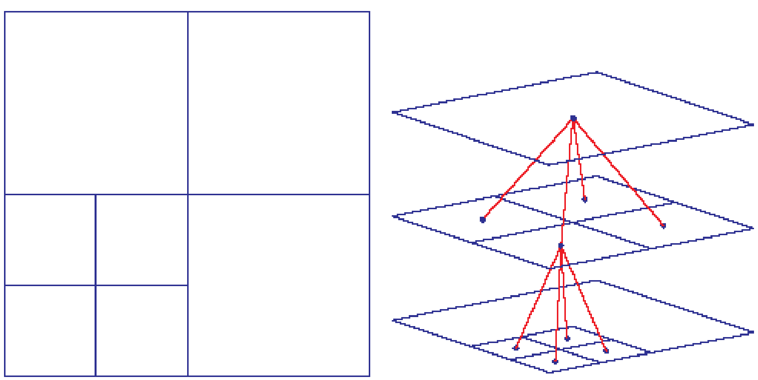
$$\min_{\mathcal{P}} R(\mathcal{P}) + \tau D(y, \mathcal{P}).$$

- Bottom-up Tree pruning algorithm, fast

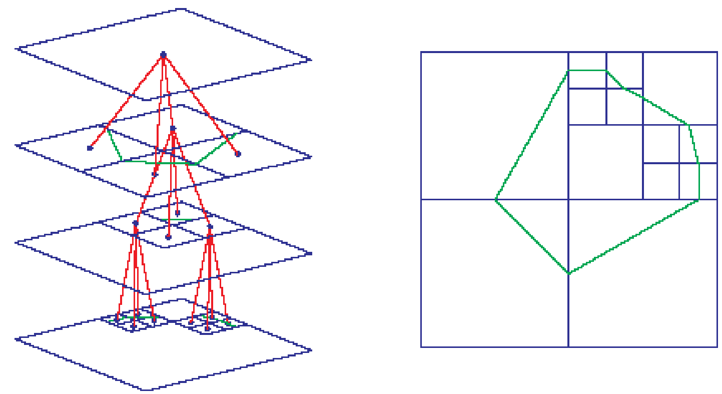
Pruning a Quad-tree



1. A complete quad-tree



2. An admissible sub-tree



3. An admissible sub-tree with beamlet decoration

Distortion

- Two components in distortion:
  - Sum of square of Euclidean distance from image to beamlet representation (~ Hausdorff distance)
  - Degree of overlapping

$$d_S(y, b) = L_1(y \cap S, b) + \lambda \cdot L_2(y \cap S, b)$$

Zero-Tree Coding

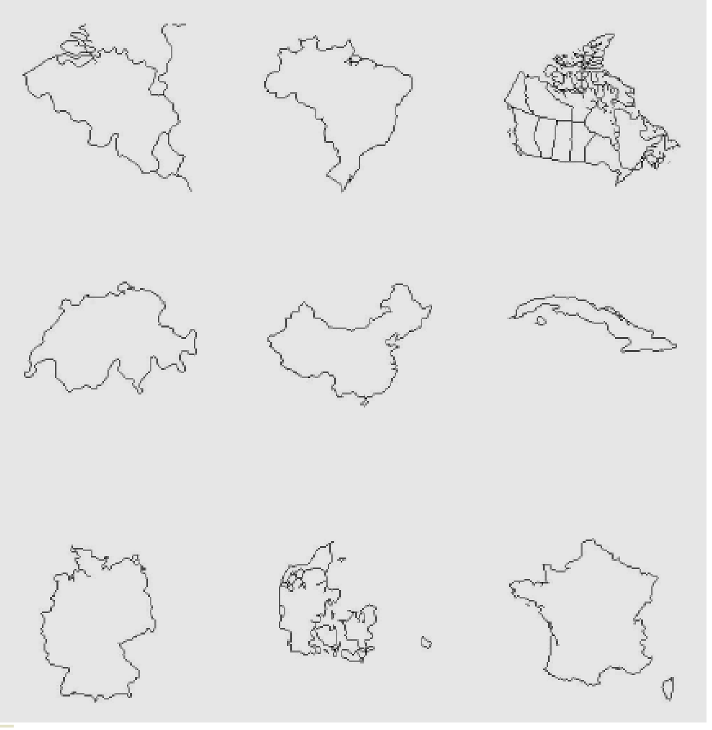
$$
\begin{array}{lll}
l_0 & \rightarrow & \mathrm{Q} \\
l_1 & \rightarrow & \mathrm{Q} \\
 & & \mathrm{N} \\
 & & \mathrm{N} \\
 & & \mathrm{B}_1 \quad 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
l_2 & \rightarrow & \mathrm{N} \\
 & & \mathrm{N} \\
 & & \mathrm{B}_2 \quad 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\
 & & \mathrm{B}_3 \quad 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0
\end{array}
$$

- Representation
- Symbol Stream
- Rule that generates the symbol stream

Q, Q, N, N, B, 1, 0, 0, N, N, B, B, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0.

Q, Q, N, N, B, $B_1(1)$, $B_1(2)$, $B_1(3)$, N, N, B, B, $B_1(4)$, $B_1(5)$, $B_2(1)$, $B_2(2)$, $B_2(3)$, $B_3(1)$, $B_3(2)$, $B_3(3)$, ... $B_1(2J)$, $B_1(2J+1)$, $B_2(2J-2)$, $B_2(2J-1)$, $B_3(2J-2)$, $B_3^2(2J-1)$.

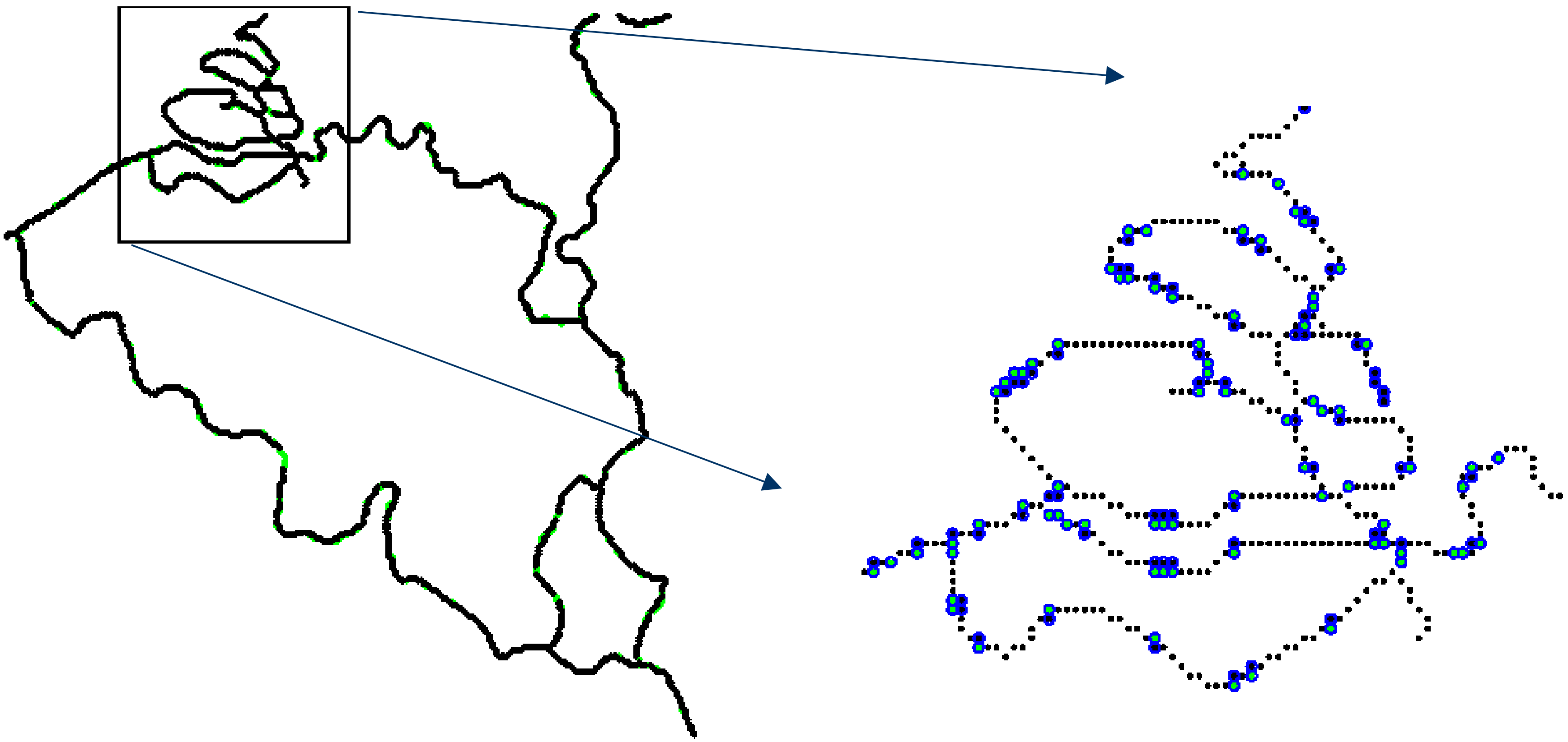
IV. Simulation Results: Comparison with JBIG



Country	Pixels	JBIG 2 (lossless)	JBEAM (lossless)	JBEAM (lossy)
Belgium	1233	7888	6114	3098
Belize	822	5768	4391	2021
Canada	2911	15224	14555	7366
Switzerland	845	6088	4171	2135
China	813	5792	4609	2041
Cuba	658	4360	3303	1542
Germany	871	5792	4916	2138
Denmark	1350	7992	6780	3735
France	920	6448	5300	2339

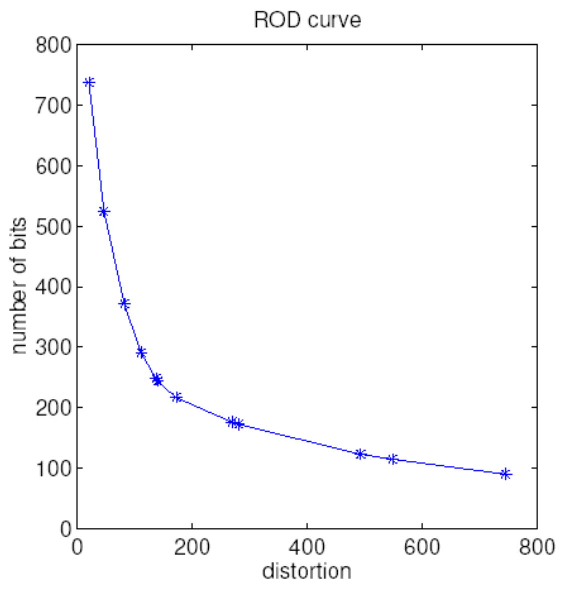
No CAE and PWC…

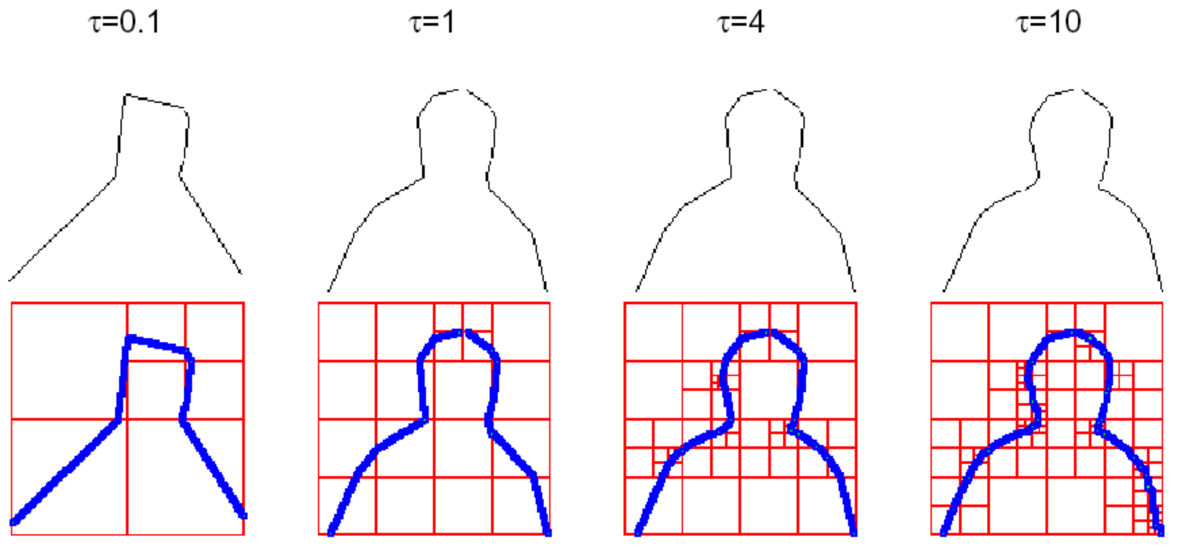
Effects of Distortion



Simulation: Rate-distortion tradeoff

min $D+\tau R$





- Distortion decreases.
- Rate increases.

Simulation: Progressive Reconstruction

Rate & Distortion Analysis

- Polygonal, or model with finite number of parameters, an oracle can do:

$$D(R) \sim C_1 2^{-C_2 R}$$

- JBEAM:

$$D(R) \sim C_3 2^{-C_4 \sqrt{R}}$$

- Horizon model ($\mathbf{C}^2$)

$$D(R) \sim \log_2^2(R) \cdot R^{-2} \quad \text{versus} \quad D(R) \sim R^{-2}$$

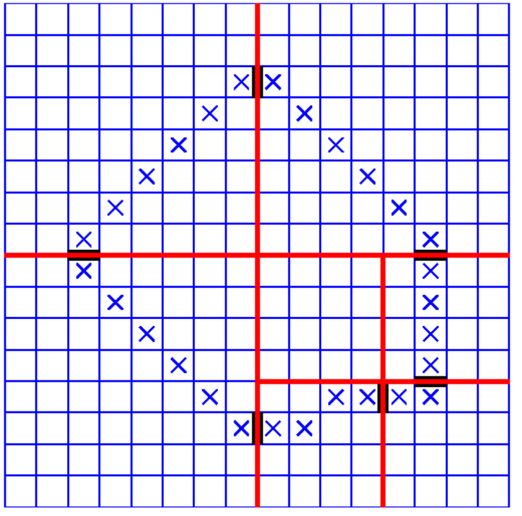
JBEAM Best

Discussion

- Multiscale curve representation: Strip Trees, 70’s; and many more; no beamlets, no zero-tree, no R-D optimization
- Recent works in image compression, e.g. wedgeprint, bandelets, curvelets, ridgelets, contourlets, etc
- Partial progressivity versus full progressivity?

Discussion (2)

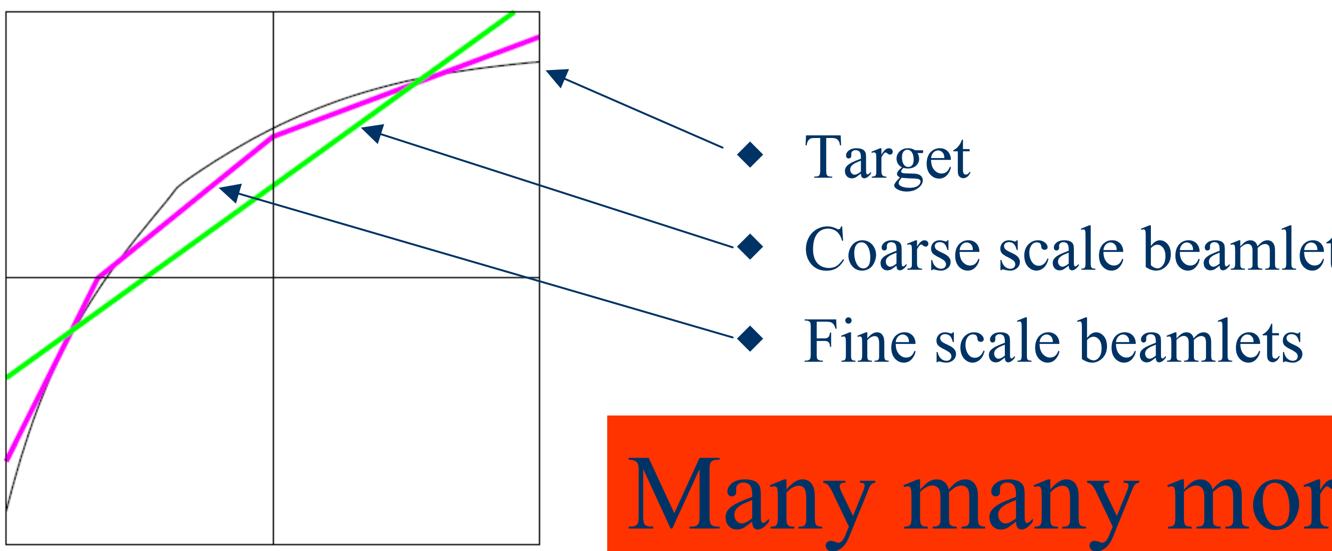
- Shape coding versus curve coding



- Hidden Markov Model

Discussion (3)

- More innovative way to take advantage of SPIHT.



Many many more…

V. Conclusion

- Combining zero-tree algorithm and beamlet.
- Numerical experiments show promises.
- First multiscale curve coder.
- Software, more research.

Beamlets: Multiscale,
Curves: Geometry.

“Connect-the-dots” Problems and Dynamic Programming Solutions

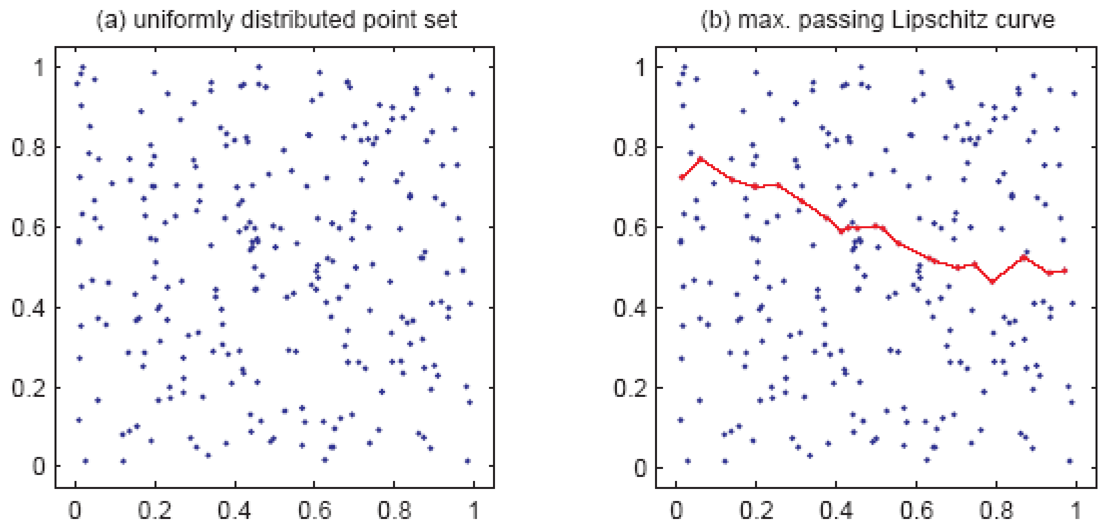
Xiaoming Huo
School of Industrial and Systems Engineering
Georgia Institute of Technology

Joint work with David L. Donoho, Craig Tovey, and Ery Arias-Castro

9 / 23 / 2004

What is a Connect-The-Dots (CTD) Problem?

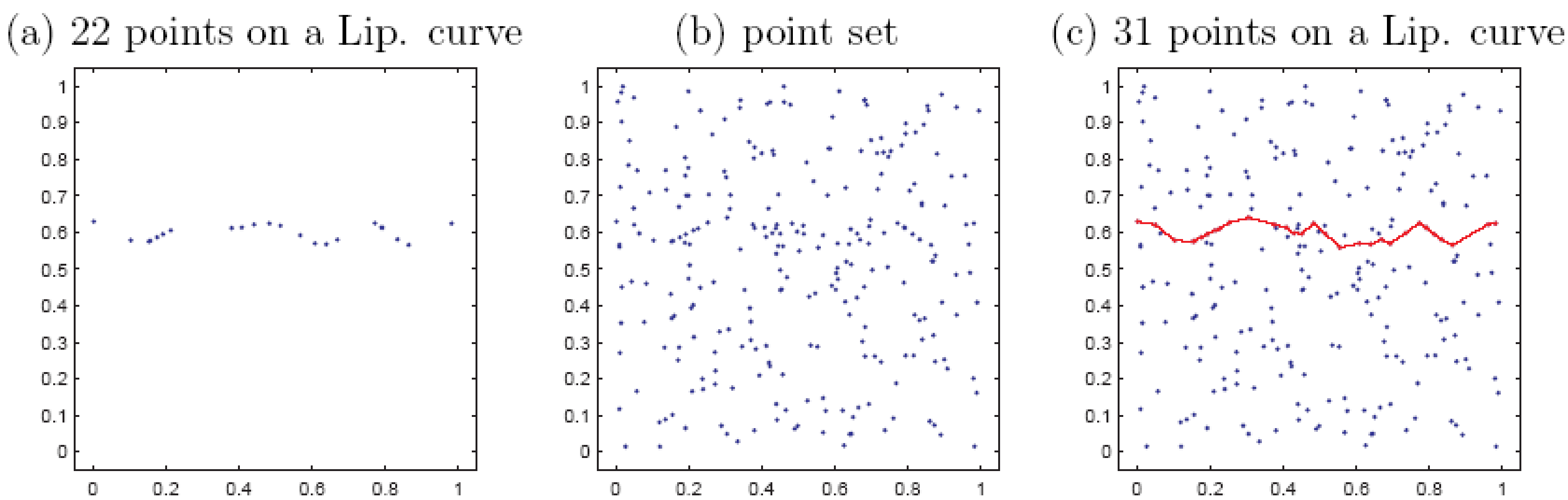
- Size of maximum subset of dots on a function from a given functional class.



Maximum number of dots on a Lipschitz graph

Why CTD Problem

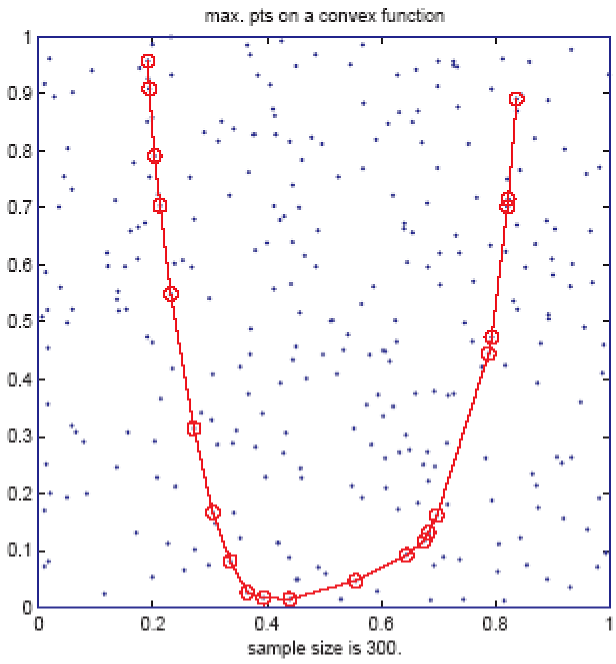
- Test the Uniformality



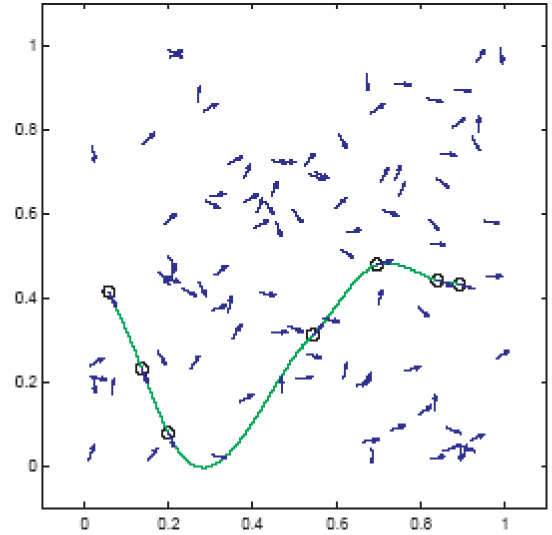
N = 256, C = 1

Other Classes

- Convex functions

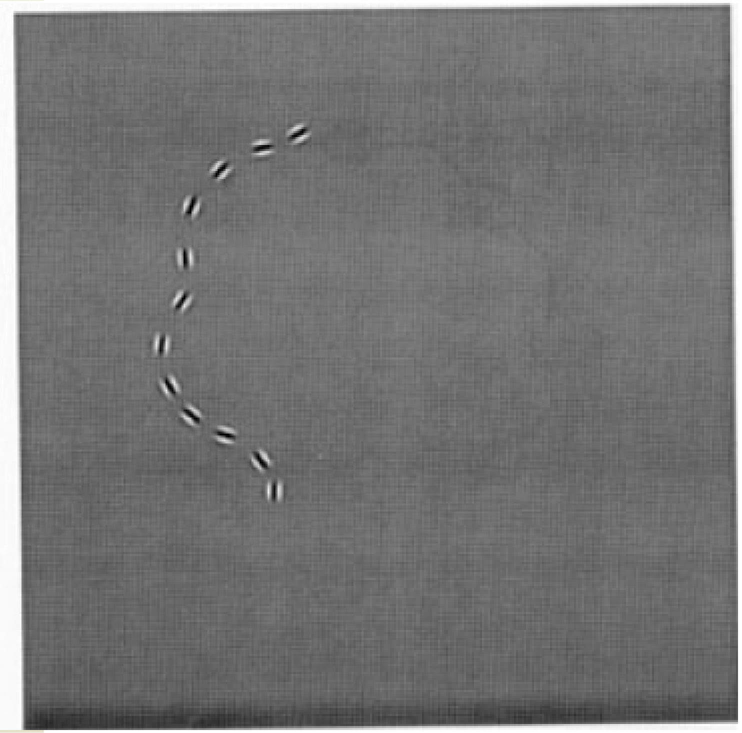


- Vectors; Connect-the-darts



Applications (1)

- Illusory contour (Psychophysics)



This talk…

- Different formulation, due to different motivations – CTD problems
- Efficient *dynamic programming* algorithms to solve these problems.
- Simulations / Insights / Software.
- Main objective: Fundamental understanding (asymptotic distributions).

Outline

- I. Classes of problems and *dynamic programming* approaches
- II. Complexity
- III. Theoretical Insights
- IV. Software
- V. Conclusion

I. Classes of Problems & DP Approaches

- Increasing, Unimodal, Lipschitz
- Extra variable: convex, Holder-2
- High-dimensional cases
  - Increasing
  - Lipschitz
  - Holder-2
- Connect-the-Darts

I. Increasing Functions

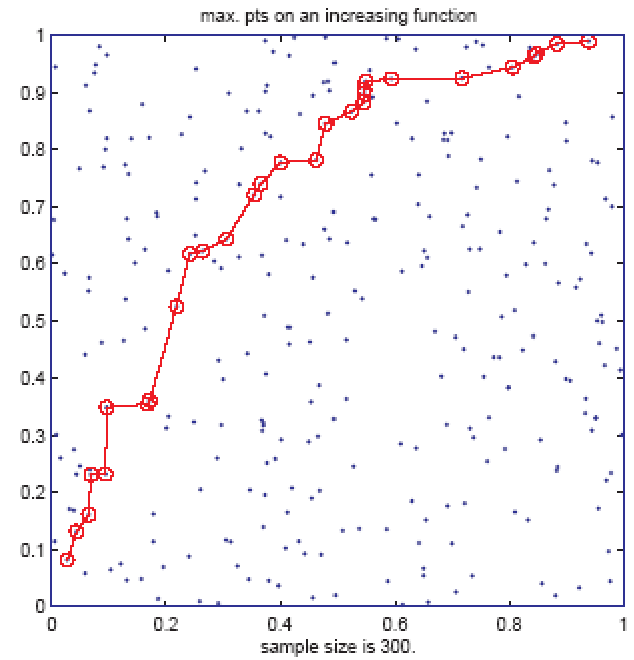
- Incr. Function

$$x < x' \Rightarrow f(x) \leq f(x').$$

- M(i) = maximum number of point on an incr. func. up to point i.

$$M(i) = 1 + \max_{(j<i, y_j \leq y_i)} M(j)$$

- *DP* approach

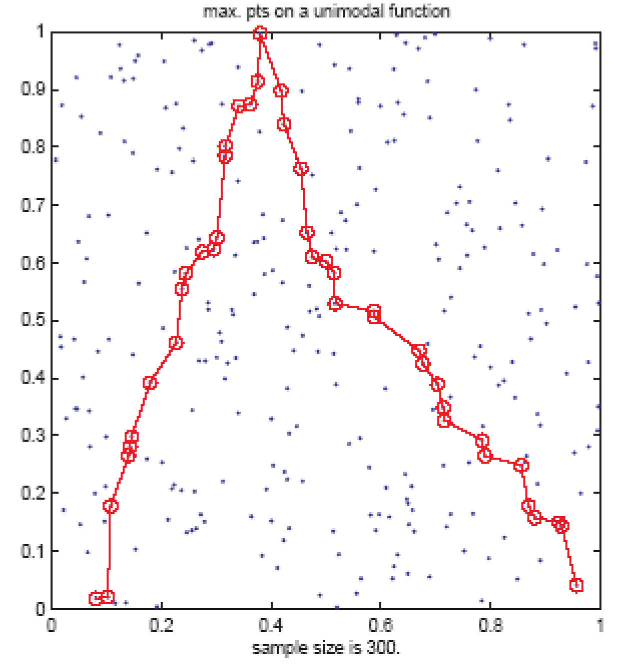


Fast Algorithm for Incr. Functions

- There is an O(n log(n)) time and O(n) space algorithm to solve CTD problem with increasing funtions, Fredman 1975.
- While the data are ordinal, there is an O(n log log(n)) time and O(n) space algorithm, Bespamyatnikh and Segal (2000).
- We will extend it to high dimensions.

Unimodal

- Non-decreasing up to a point, then non-increasing
- Solve it by running the algorithm for Incr. twice.
- Fast algorithms have the same complexity



Lipschitz

- Lipschitz function

$$|f(x)-f(x')|\leq C|x-x'|$$

- Convert it into Incr. function problem.

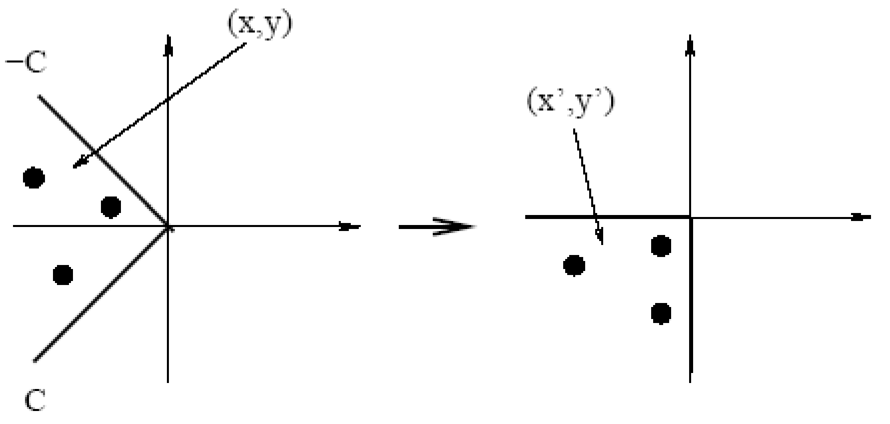
$$(x_i',y_i')=(x_i,y_i)\begin{pmatrix} \frac{1}{2}\sqrt{1+C^2} & \frac{1}{2}\sqrt{1+C^2} \\ -\frac{1}{2C}\sqrt{1+C^2} & \frac{1}{2C}\sqrt{1+C^2} \end{pmatrix}.$$

$$|y_1-y_2|\leq C|x_1-x_2| \longrightarrow x_1'\leq x_2' \quad \text{and} \quad y_1'\leq y_2'.$$

Lipschitz Increasing

Lipschitz (2)

- Illustration of the previous transform



- Solving CTD for Lipschitz functions is as hard as for increasing functions.

Convex

- Convex function

$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x'),$$

- DP with additional variable
  - Mi(s) = max. number of points up to i-th point with left slope s. (s is an additional variable)

Convex (2)

- DP algorithm:

$$\forall x_k < x_i : M_i(s_{ki}) = 1 + \max_{x_j < x_i} M_j(s_{ji}) \mathbf{1}\left(s_{ji} \leq s_{ki}\right),$$

Where

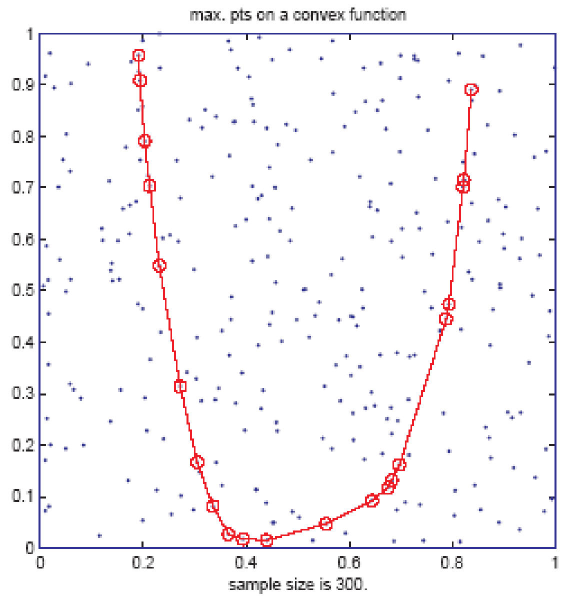
$$s_{ji} \equiv \frac{y_i - y_j}{x_i - x_j} : x_j < x_i$$

- Hence

$$L_N(\text{Conv.}) = \max_i \max_s M_i(s)$$

Convex (3)

- There is an O(n n log(n)) algorithm to solve the CTD for convex functions.



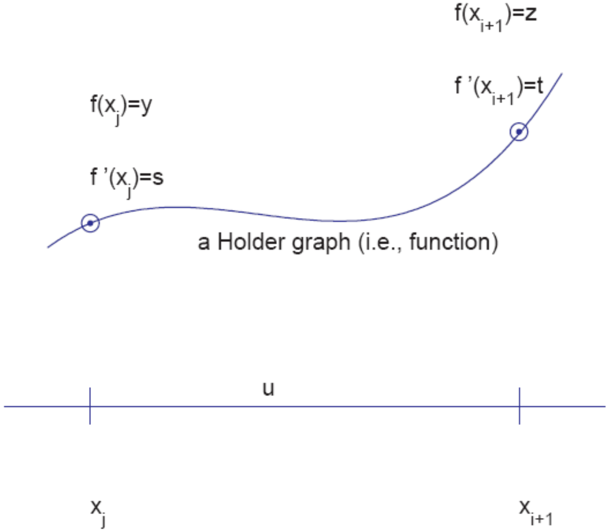
Holder-2

- Holder-2 function

$$\mathcal{G}_2 = \{f : [0,1] \to [0,1] : |f'(t) - f'(s)| \le C \cdot |t - s|\}$$

- DP approach

$$V_{i+1}(z,t) = 1_{z=y_{i+1}} + \max_{(x_j, y, s) \in R_{i+1}(z,t)} V_j(y, s)$$



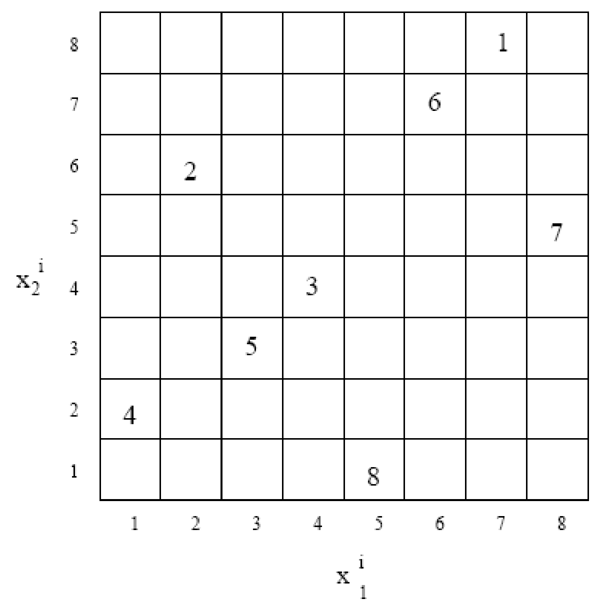
Holder-2 (2)

- Essence of the derivation
  - Solving a variational problem
  - Computing influential intervals
- Complexity: unclear, but empirically, it seems polynomial.

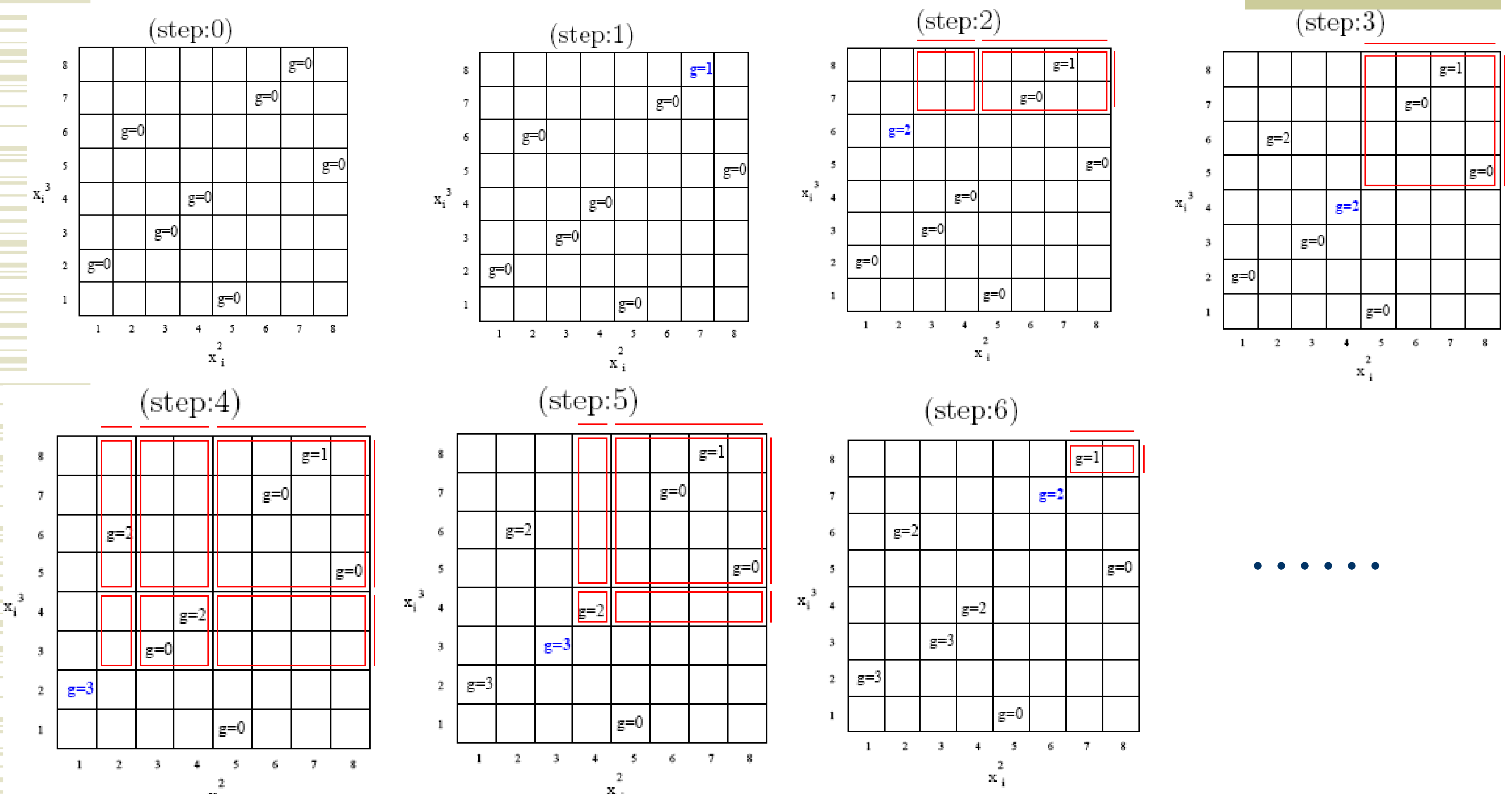
Sample size (N)	Run time (sec.)	Ave. no. of breakpoints	Max. no. of breakpoints
16	0.125	16.2	31
32	0.734	20.8	36
64	4.437	32.2	53
128	28.28	53.6	99
256	175.2	80.3	168

High-dimensional

- Longest increasing subsequence in k-dimension.
- We designed an $O(N \log^K N)$ algorithm.
- Example when k=3, N=8.

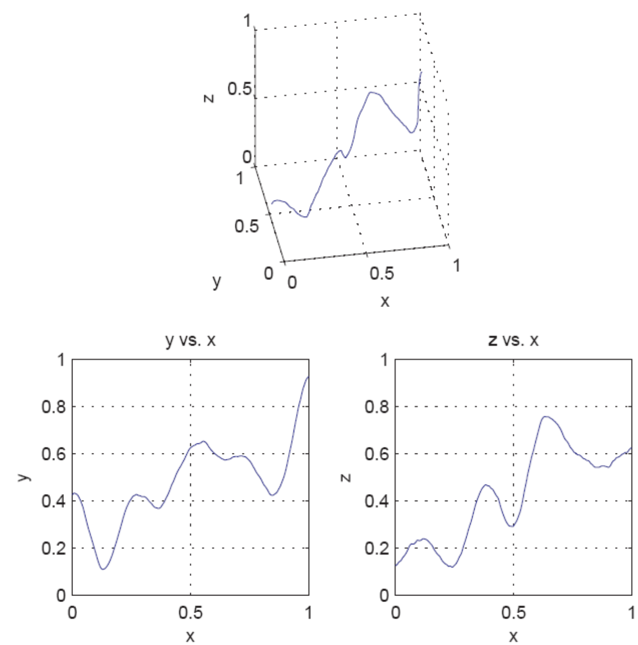


Illustrate k-D increasing subsequences



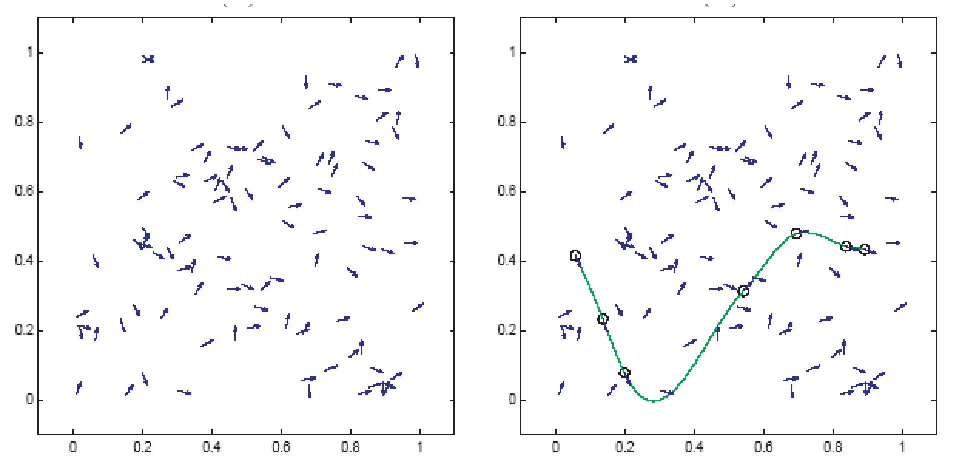
High-Dimensional Lipschitz and Holder

- Modified DP approach works for high dimensional Lipschitz functions
- Modified DP approach works for high dimensional Holder-2 functions



Connect-the-Darts

- Test of Randomness



- Similar approach as in Holder-2
- Knowing the angle `saves' many computing, connection in psychophysics.

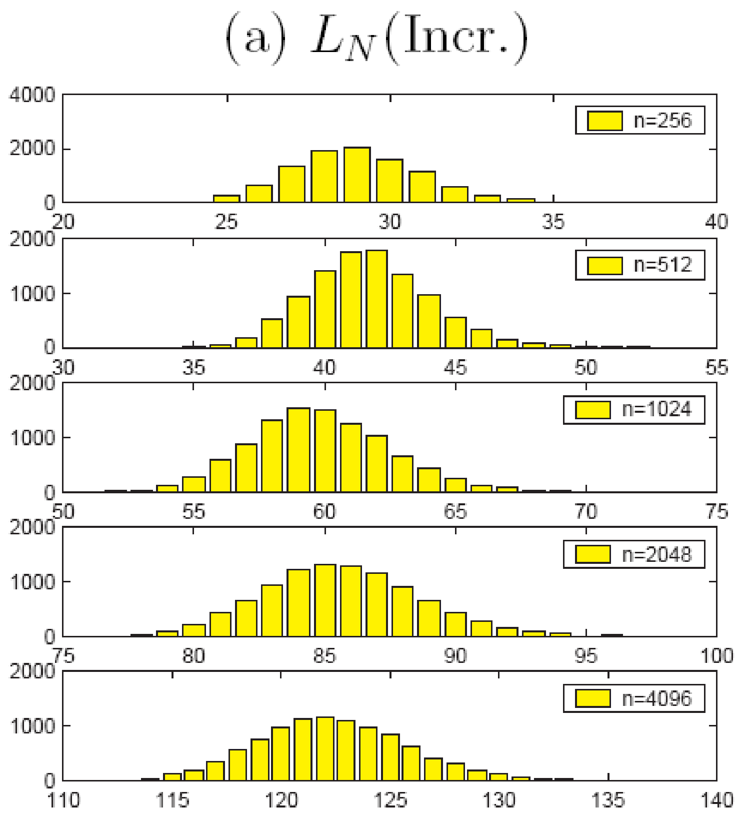
II. Complexity

Case Description	Complexity
Increasing	$O(N\log(N))$
Unimodal	$O(N\log(N))$
Lipschitz 1-graph in $\mathcal{R}^2$	$O(N\log(N))$
Convex	$O(N^2\log(N))$
Increasing in $\mathcal{R}^d$	$O(N(\log(N))^{d-1})$
Lipschitz 1-graph in $\mathcal{R}^d$, $d > 2$	$O(N^2 d)$
Vector Field under **(C1)**	$O(N^2)$
Finite Length Curve	NP-hard

III. Insights

- Limit distributions
- Asymptotic rates
- Concentration of measures
- Typical cases

III-1. Limit Distributions

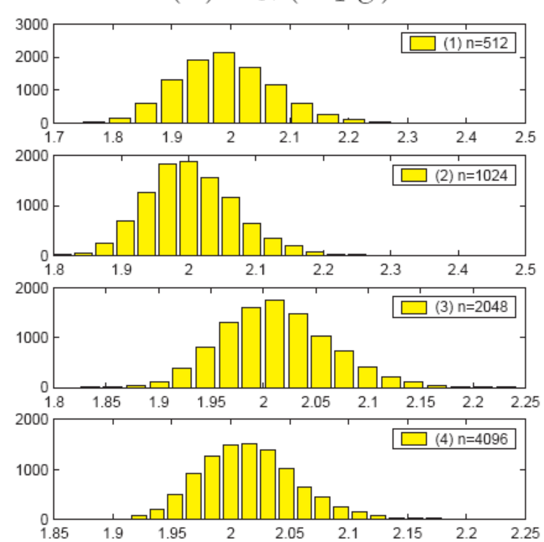


- Consistent with known theory in geometric probability.
- Tracy & Widom (2001) law.

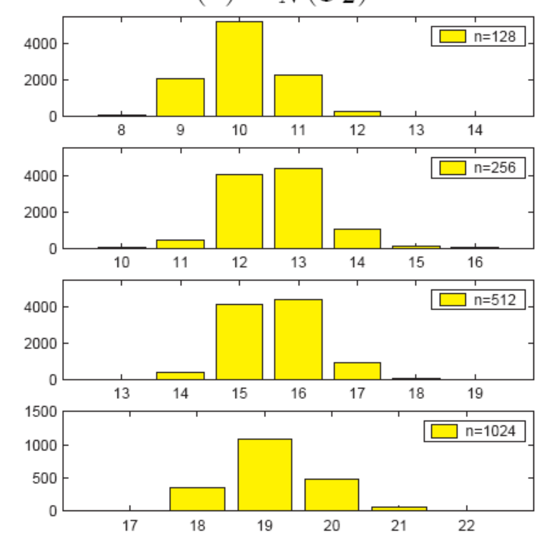
Limit Distributions (2)

- New cases
  - Lipschitz
  - Holder-2
  - High dimension
  - Connect the darts

(b) $L_N(\mathrm{Lip}_C)$



(c) $L_N(\mathcal{G}_2)$



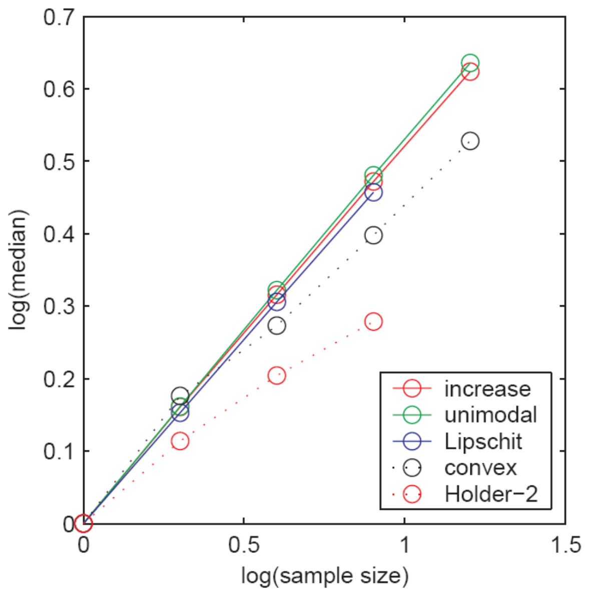
III-2. Asymp. Rates

- For increasing functions, we have

$$\lim_{N\to\infty} \frac{L_N(\mathcal{F})}{2\sqrt{N}} = 1, \quad \text{w.p. } 1$$

- In general, we have

$$L_N(\mathcal{F}) \asymp \text{Constant} \cdot N^{\rho}$$



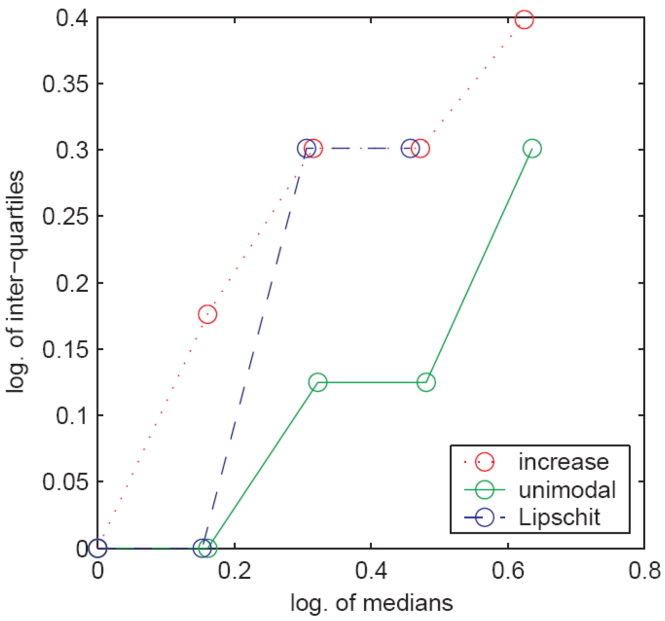
Case Description	Slope	Predicted Rate
increasing	0.5182	1/2
unimodal	0.5282	1/2
Lipschitz in $\mathcal{R}^2$	0.5065	1/2
convex	0.4387	1/3
Hölder-2	0.3087	1/3

III-3. Concentration of Measure

- What is concentration of measure?

  Log(IQR) < .5 log(median).

- The square root seems to be true in all cases.

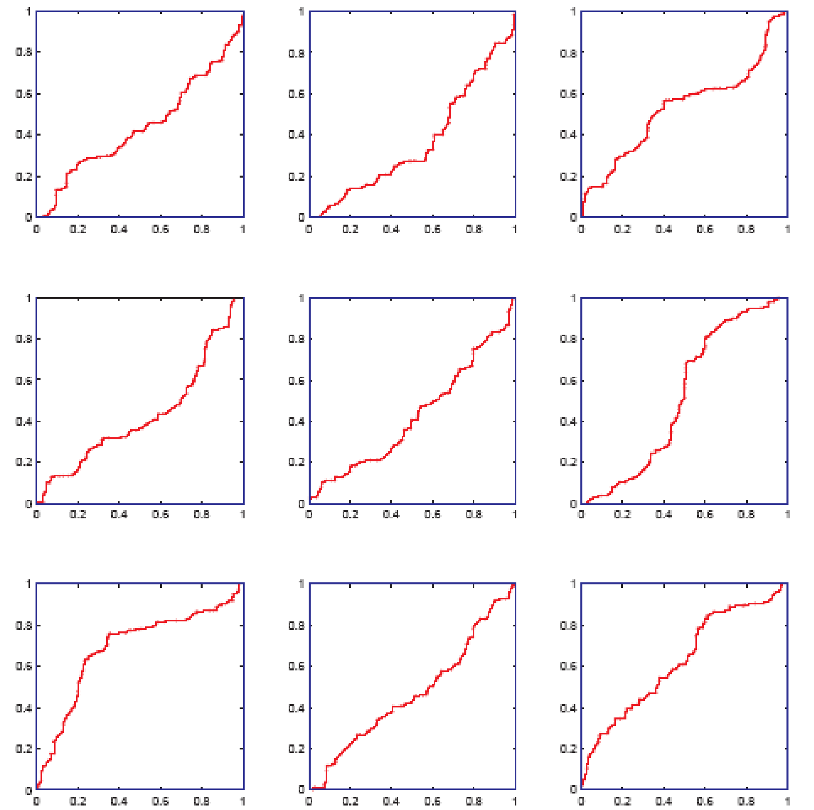


Case Description	increasing	unimodal	Lipschitz in $\mathcal{R}^2$
Slope	0.6378	0.4733	0.6582

III-4. Typical Cases

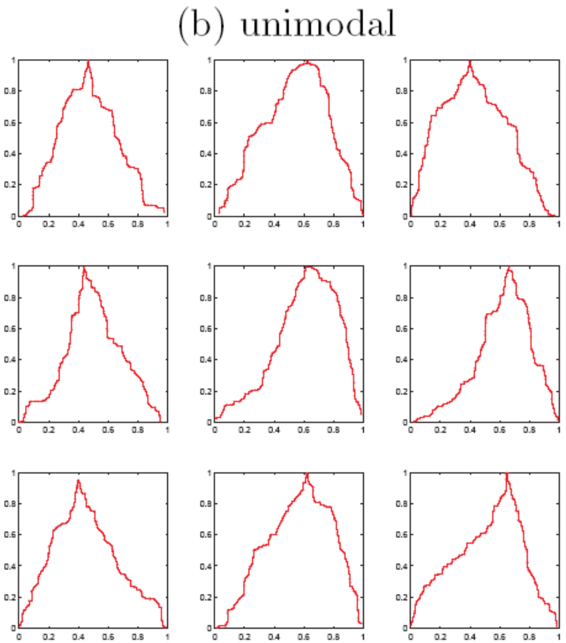
- Increasing, consistent with current literature.
- Deuschel & Zeitouni. (1995,1999)

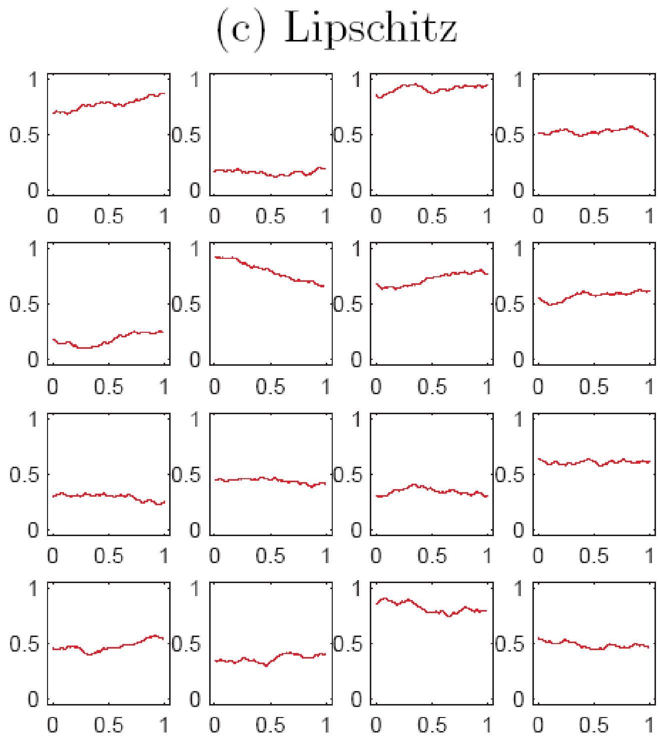
(a) increasing

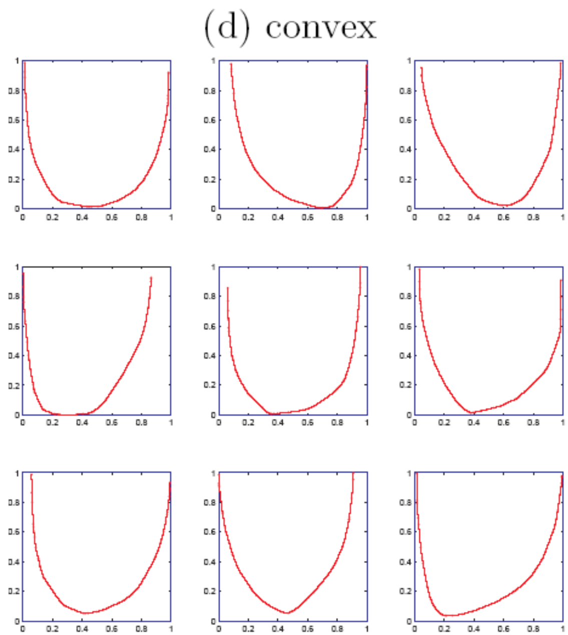


Typical Cases (2)

- Unimodal
- Lipschitz
- Convex







IV. Software

- CTDLab.
- Implementation of all algorithms
- Demos for the figures
- Documentation

V. Conclusion

- Introducing CTD problems.
- Dynamic programming approach for a set of CTD problems - New algorithms, with low computational complexity.
- Theoretical insights.
- Applications: hypothesis testing, data analysis, fundamental detection theorems, …

Multi-scale Significance Run Algorithm

Xiaoming Huo
School of Industrial and Systems Engineering
Georgia Institute of Technology

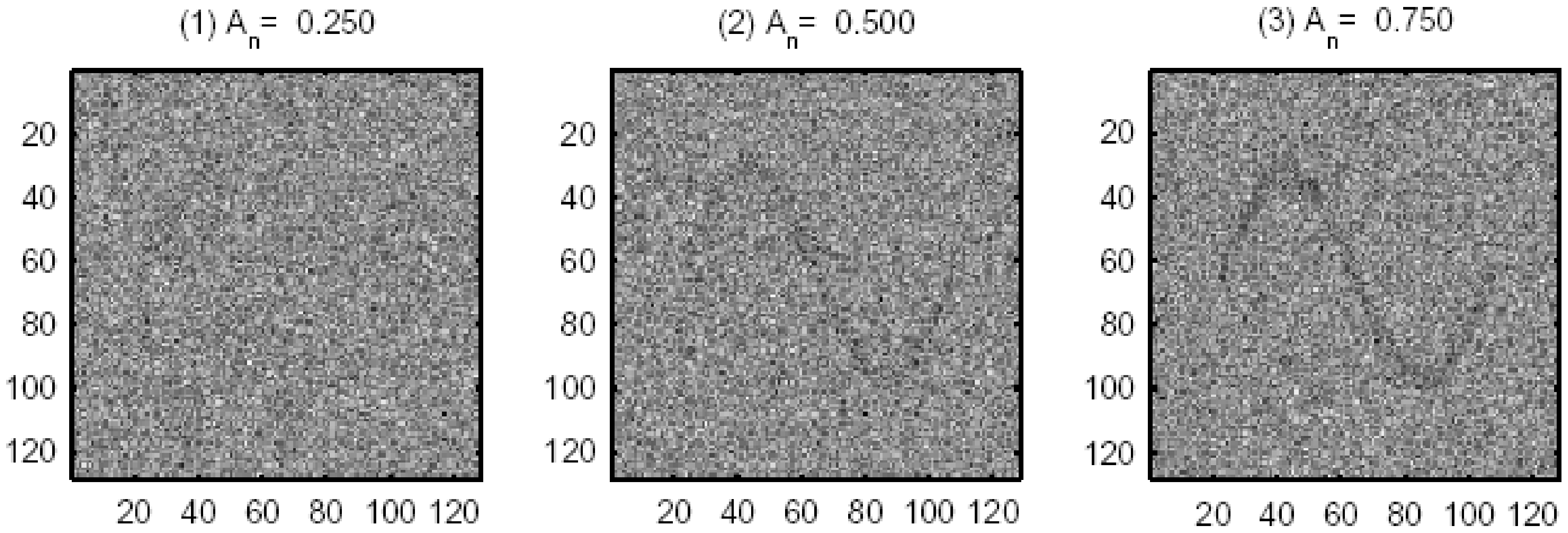
Joint work with David L. Donoho, Ery Arias-Castro

9 / 23 / 2004

I. Introduction — The Problem

- What's the problem?

  ---- Detecting the presence of a filament (or a filament-like feature) in a noisy picture.



II. Multiscale Analysis and Significance Run

- Multiscale components:
  - Axoids
  - Beamlets
- Significance run and detection
- Fundamental theory in detecting filamentary structures
- Most powerful test!
- Scan statistics

Multiscale Analysis

- Axoids

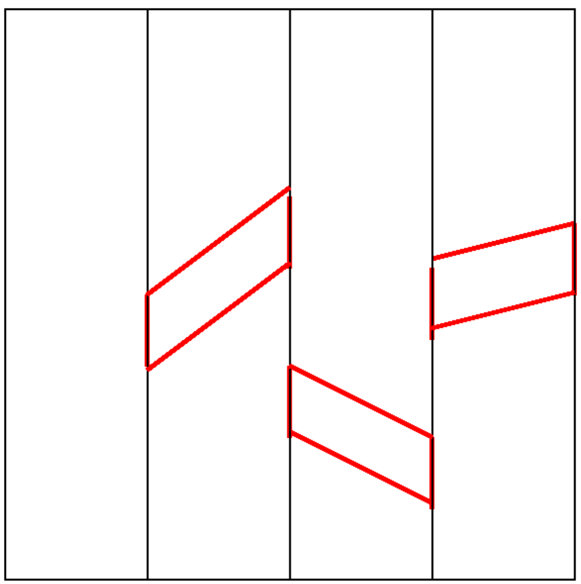
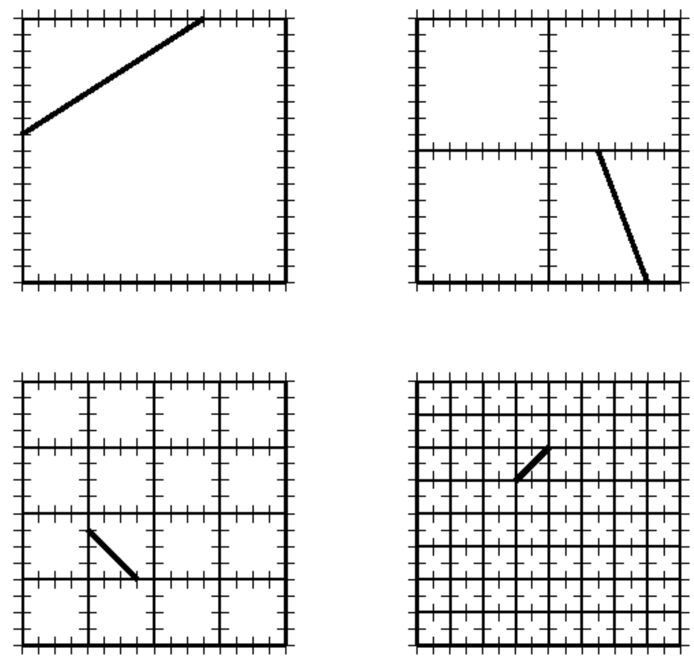


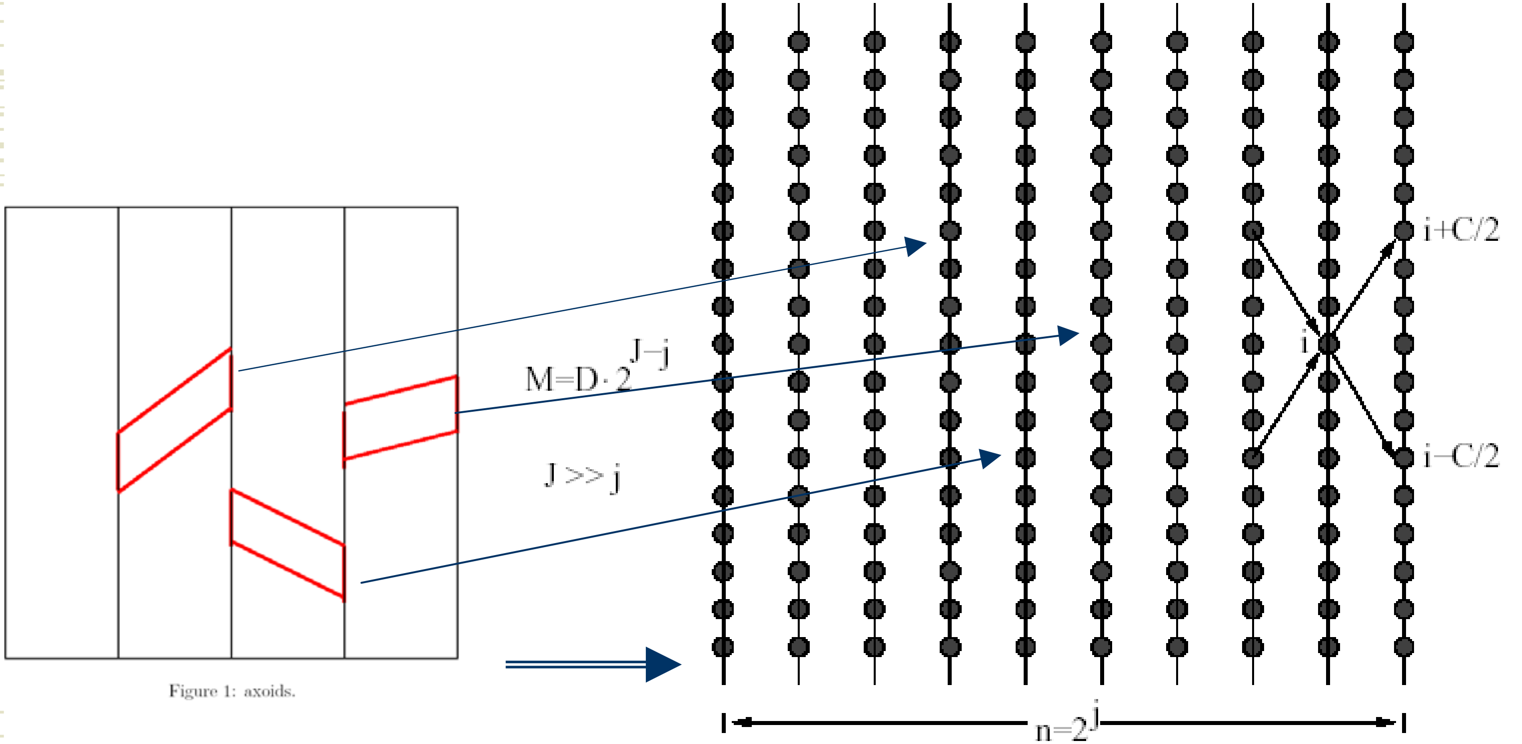
Figure 1: axoids.

- Beamlets



Significance Run

- Significance run and Bernoulli Table (significance graph)



Significance Run

Longest significance run

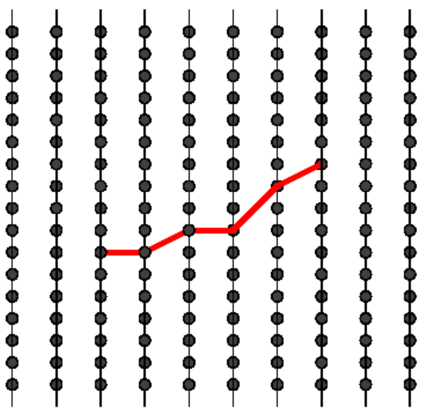
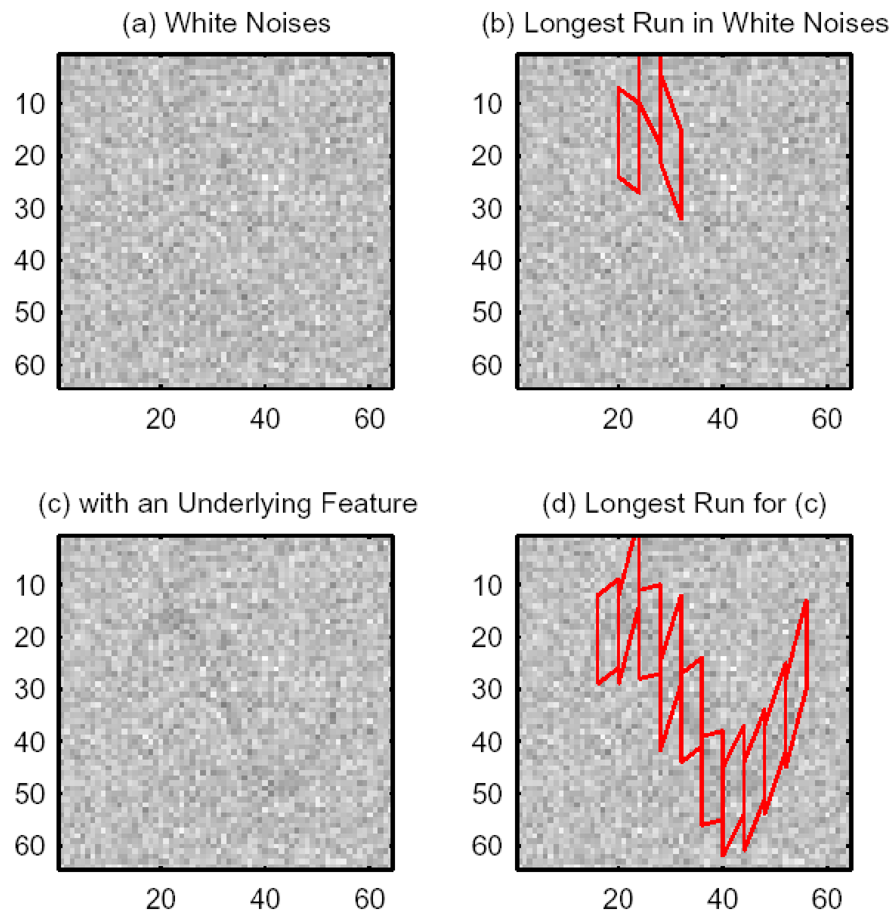


Figure 2: longest path.



Statistical Theorems

- Fundamental theorem in detecting filamentary structures in point clouds

Theorem 3.1 *There is a single choice of thresholds N^* and $(L_n^*)_n$ so that for every $\alpha \in (1,2]$ and $\beta > 0$, there is $T_*(\alpha, \beta, \rho, S)$ so that, for each $\tau > T_*$*

$$P\{ \text{ Test rejects } \mathbf{H}_0 \mid \mathbf{H}_1(\alpha, \beta, S, \tau) \} \to 1 \text{ as } n \to \infty;$$

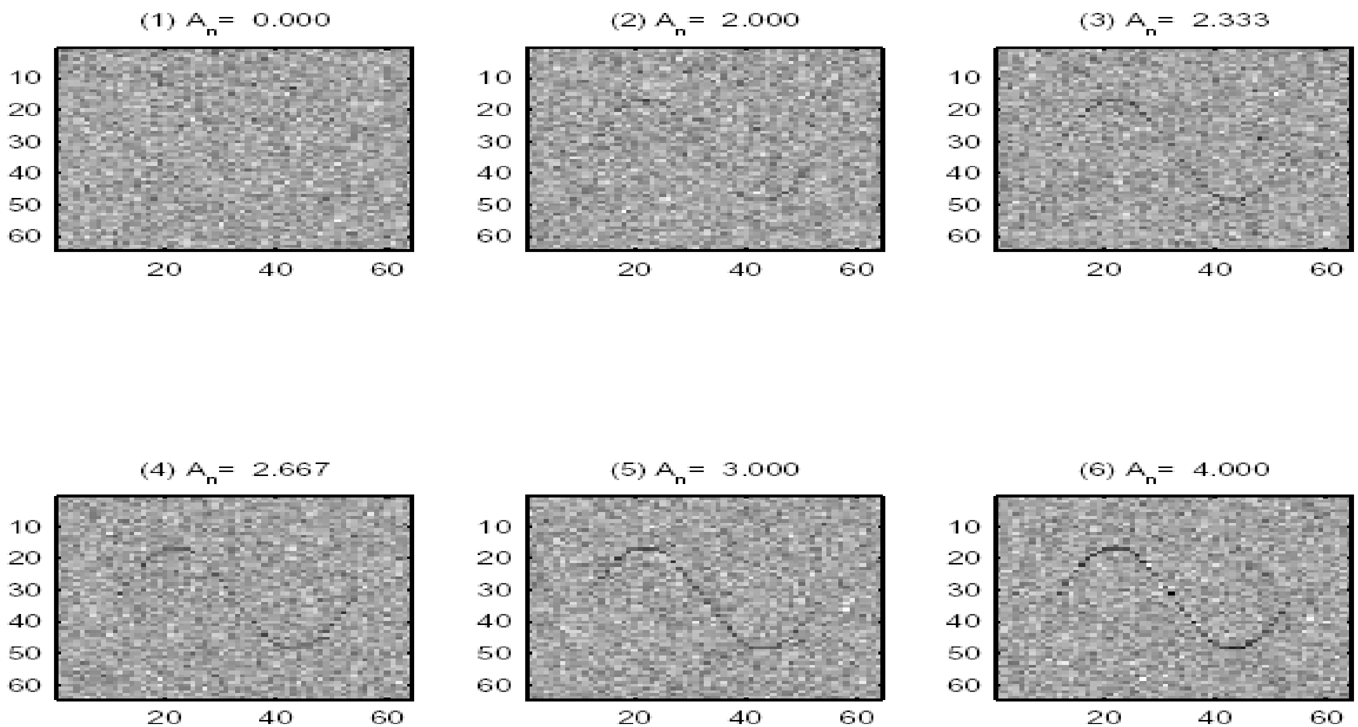
at the same time

$$P\{ \text{ Test rejects } \mathbf{H}_0 \mid \mathbf{H}_0 \} \to 0 \text{ as } n \to \infty.$$

In words, under $\mathbf{H}_0$ the longest significant path is overwhelmingly unlikely to be substantially longer than L_n^* for large n, while under each indicated $\mathbf{H}_1$, the longest significant path is overwhelmingly likely to be substantially longer than L_n^*. The threshold T_* is within a constant factor T_*/T_- of the optimal detection threshold; this shows that, up to constants, we can adaptively test for existence of fragments of C^α graphs.

- Most powerful test in significance runs
  - If there is a test with power ~ 1, then MSRA will have that power.

III. Simulations — Some Figures



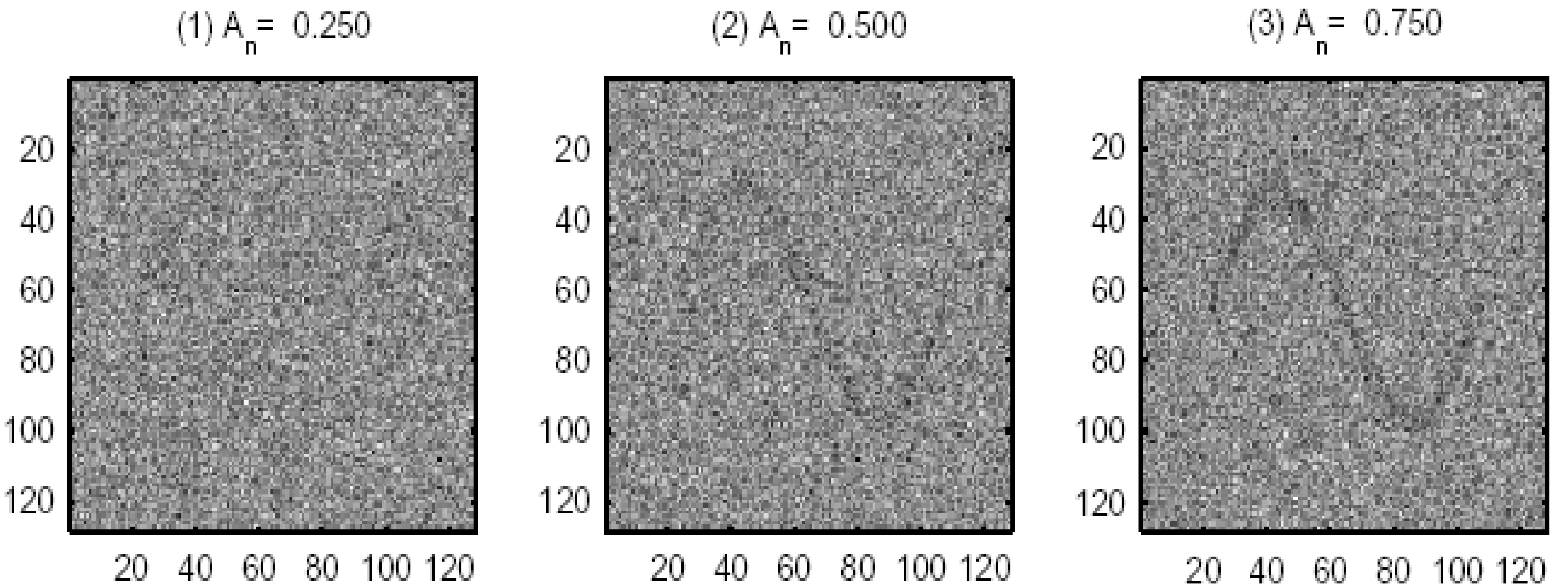
Noisy images with an underlying trigonometric curve having a range of amplitudes (provided in the titles). It shows that the proposed method can detect an underlying feature which is not obviously visible

Table 4: Detections out of 10 simulations for a trigonometric curve while $n = 64$ and $t = 1$.

$A_n =$	0	2.000	2.333	2.666	3.000	4.000
Detected Cases	0	0	4	7	9	10

Simulations

- Increased sample size



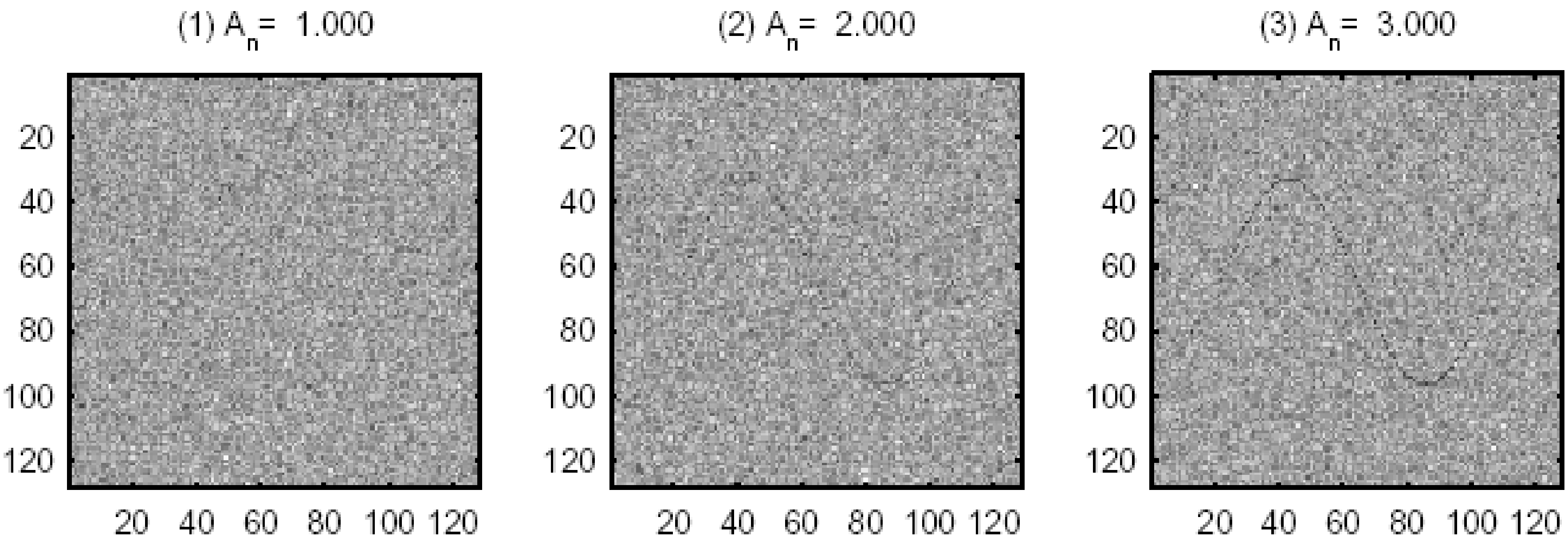
The random images that are used in the experiments that are reported in Table 5. The thickness of the underlying feature is t = 8.

Table 5: Detections out of 10 simulations for a trigonometric curve while $n = 128$ and $t = 8$.

$A_n =$	1/4	1/2	3/4	1
Detected Cases	0	10	10	10

Simulations — more

- Thinner objects



The random images that are used in the experiments that are reported in Table 6. The thickness of the underlying feature is $t = 1$.

Table 6: Detections out of 10 simulations for a trigonometric curve while $n = 128$ and $t = 1$.

$A_n =$	1	2	3	4
Detected Cases	0	3	10	10

Simulations

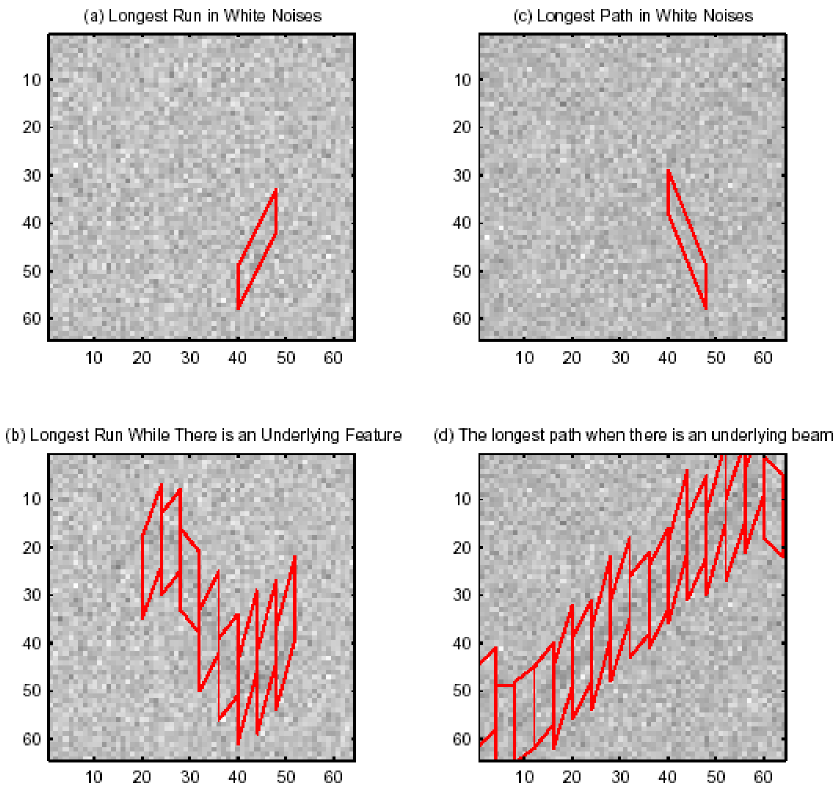


Illustration of an increment of length of the longest run when an underlying feature (a trigonometric function for (a) and (b), and a beam for (c) and (d)) is present. When there is an underlying feature, the length of the longest run is significantly larger.

IV. Conclusion

- Multiscale detection
- Automatic adaptation
- Fast algorithms
- Length of the longest significance run; limit distribution

Multiscale + Geometry