Detecting Faint Edges in Noisy Images: statistical limits, computationally efficient algorithms and their interplay

Boaz Nadler

Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science

Joint works with
Inbal Horev, Sharon Alpert, Nati Ofir,
Meirav Galun, Ronen Basri (WIS)
and
Ery Arias-Castro (UCSD)

Jan. 2016

Edge Detection

A fundamental task in low level image processing. Key ingredient in various applications.

Edge Detection

A fundamental task in low level image processing. Key ingredient in various applications.

Let I be an $n \times n$ (discrete) image. An edge is a curve Γ s.t. at all pixels $(i,j) \in \Gamma$

$$\nabla I \cdot \mathbf{n}\Big|_{(i,j)\in\Gamma} \text{ is ' large'}$$

$>$ 30 years of research, many edge detection algorithms

Edge Detection

A fundamental task in low level image processing. Key ingredient in various applications.

Let I be an $n \times n$ (discrete) image. An edge is a curve Γ s.t. at all pixels $(i,j) \in \Gamma$

$$\nabla I \cdot \mathbf{n}\Big|_{(i,j)\in\Gamma} \text{ is ` large'}$$

$>$ 30 years of research, many edge detection algorithms

Edge Detection

A fundamental task in low level image processing. Key ingredient in various applications.

Let I be an $n \times n$ (discrete) image. An edge is a curve Γ s.t. at all pixels $(i,j) \in \Gamma$

$$\nabla I \cdot \mathbf{n}\Big|_{(i,j)\in\Gamma} \text{ is ' large'}$$

$>$ 30 years of research, many edge detection algorithms

Popular Methods: Detect edges from *local* image gradients or more recently learned edge filters.

Previous Works

Hundreds of papers on edge detection...

Classical Works: zero-crossings of image Laplacian [Marr & Hildreth 80'], Gaussian smoothing+gradients [Canny 1986], variational interpretations [Kimmel & Bruckstein 03']

Anisotropic Diffusion: Perona and Malik 90', Weickert 97', etc.

Wavelet / Curvelet / Contourlet Methods: focus is on sparse image representation, but can be used for edge detection.

Learning-Based Approaches for Natural Images PB [Malik et. al.] Boosted Edge Learning (BEL) [Dollar, Tu, Belongie, 2007], Structured forests [Dollar and Zitnick, 2013].

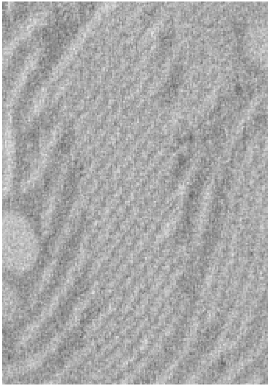
Edge Detection at low SNR

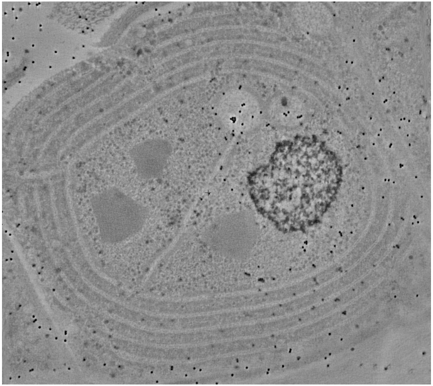
Our Focus:

Faint edge detection in **very noisy** 2D images and 3D video

Motivations:

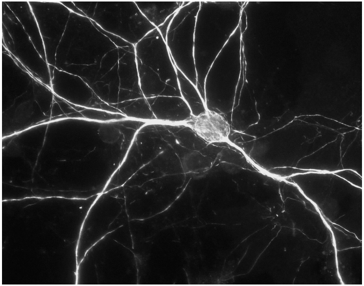
1. Bio-medical imaging.
2. Natural images at non-ideal conditions: poor lighting, fog, rain, night.
3. SAR images, various surveillance applications.
4. Object tracking in (noisy) 3D video.



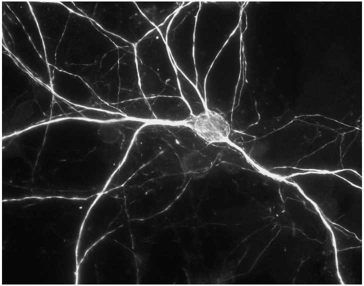


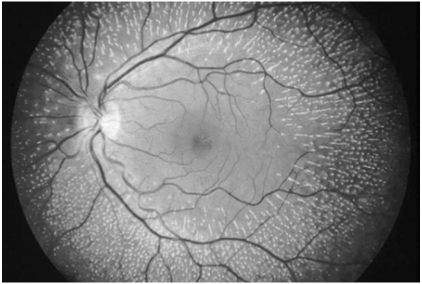
Example: Electron Microscopy
[Photosynthetic membranes in chloroplast]
[Data: Z. Reich, E. Shimoni and O. Rav-Hon, Weizmann]

Biological/Biomedical Applications



Biological/Biomedical Applications





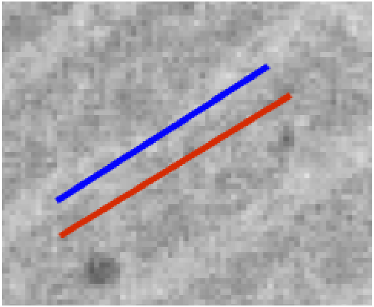
Poor Visibility

Why is faint edge detection difficult ?

Empirically: at high noise levels, local methods typically *fail*

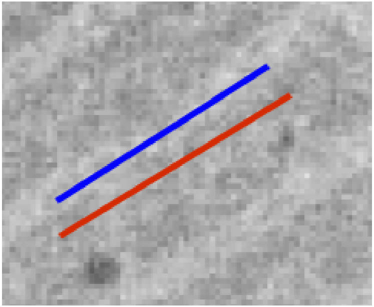
Why is faint edge detection difficult ?

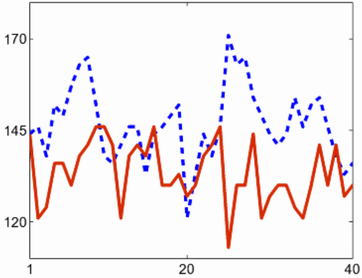
Empirically: at high noise levels, local methods typically *fail*



Why is faint edge detection difficult ?

Empirically: at high noise levels, local methods typically *fail*

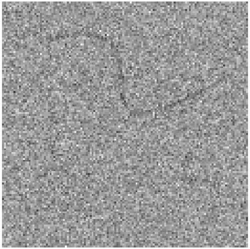




Note the **contrast reversals** (locations where the red curve exceeds the blue one).

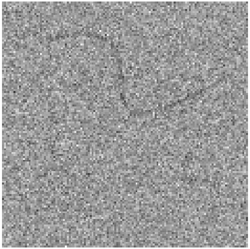
Edge Contrast and Edge Length

At high noise levels: only long edges can be detected



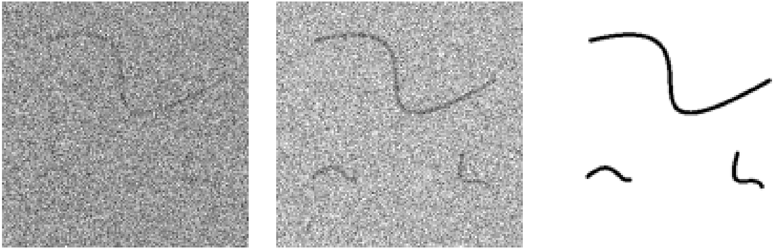
Edge Contrast and Edge Length

At high noise levels: only long edges can be detected

Edge Contrast and Edge Length

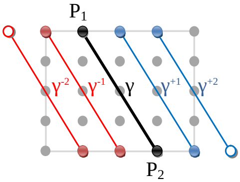
At high noise levels: only long edges can be detected



At lower levels of noise, shorter ones easily detected as well

Optimal Faint Edge Detection

To identify weak noisy edges, apply *matched filter* of width w:

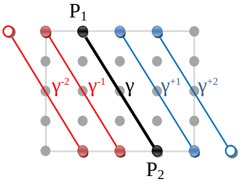


- smooth along the edge

- compute difference across edge (after smoothing)

Optimal Faint Edge Detection

To identify weak noisy edges, apply *matched filter* of width w:



- smooth along the edge

- compute difference across edge (after smoothing)

Problem: Don't know in advance where edge is !

Edge Detection $\equiv$ Search/Test *all* feasible curves

Edge Detection Algorithmic Framework

Input:
I = Noisy $n \times n$ image
σ = noise level
$\mathcal{S}_L$ = family of feasible curves of length L
$\alpha \in (0,1)$ = desired false alarm

Algorithm:
For $L \in [L_{\min}, L_{\max}]$

- For each $\Gamma \in \mathcal{S}_L$, compute matched filter response $R(\Gamma)$.
- keep Γ only if $|R(\Gamma)| > T = \mathit{threshold}(n, L, \alpha, \mathcal{S}_L)$,

Post-processing: edge localization, refinement, non-maximal suppression.

Output: Set of detected edges.

Choice of Threshold

control number of false detections

Choice of Threshold

control number of false detections

Multiple Hypothesis Testing (Statistics)

A-contrario principle (Morel et. al.)

[von Gioi et. al. 10']

Line Segment Detector with false detection control

Choice of Threshold

control number of false detections

Multiple Hypothesis Testing (Statistics)

A-contrario principle (Morel et. al.)

[von Gioi et. al. 10']

Line Segment Detector with false detection control

I = pure noise image.

$R_1, R_2, \ldots$ = edge responses of all $\Gamma \in \mathcal{S}_L$.

Choose threshold s.t.

$$\Pr[\max |R_i| > \mathit{threshold}(n, L, \alpha)] \leq \alpha$$

Choice of Threshold

control number of false detections

Multiple Hypothesis Testing (Statistics)

A-contrario principle (Morel et. al.)

[von Gioi et. al. 10']

Line Segment Detector with false detection control

I = pure noise image.

$R_1, R_2, \ldots$ = edge responses of all $\Gamma \in \mathcal{S}_L$.

Choose threshold s.t.

$$\Pr[\max |R_i| > \mathit{threshold}(n, L, \alpha)] \leq \alpha$$

Almost no spurious edge detections for pure noise image

Questions / Challenges:

- *Q1 - Minimal Detectable Contrast*: Which edge strengths can be reliably detected ? Dependence on length and complexity of feasible set of edges ?
- *Q2 - Computationally Efficient Methods:* to detect such edges.
- *Q3 - Severe Computational Constraints* (sub-linear time complexity).

Image Model

Observe $n \times n$ noisy image

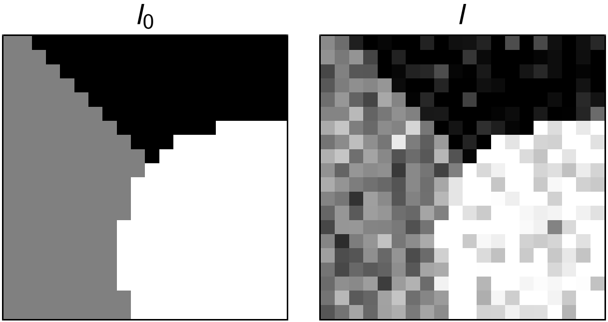
$$I = I_0 + \sigma\xi$$

I_0 = noise free image with few step edges

σ = noise level

ξ = $n \times n$ image of i.i.d. $N(0,1)$ Gaussian noise

Image Model



Step Edges + Noise

Definition: *Edge SNR (=Normalized Edge Contrast)*

$$|\nabla I \cdot \mathbf{n}|/\sigma$$

Q1- Minimal Detectable Contrast

Factors Affecting Edge Detection:

- Edge length L (matched filter reduces noise as $1/\sqrt{L}$)
- Family of feasible curves (size increases with L)

Q1- Minimal Detectable Contrast

Factors Affecting Edge Detection:

- Edge length L (matched filter reduces noise as $1/\sqrt{L}$)

- Family of feasible curves (size increases with L)

Question:

Can *any* arbitrarily faint edge be detected if it is sufficiently long ?

Lower Bound on Full Set of Curves

Lemma: Let I be a pure noise image. There exists a monotone curve $\Gamma = \Gamma(I)$ of length L, such that

$$\mathbb{E}_I[R(\Gamma(I))] = \frac{\sigma}{\sqrt{2\pi}} > 0$$

and s.t. its variance is $O(1/L)$.

Lower Bound on Full Set of Curves

Lemma: Let I be a pure noise image. There exists a monotone curve $\Gamma = \Gamma(I)$ of length L, such that

$$\mathbb{E}_I[R(\Gamma(I))] = \frac{\sigma}{\sqrt{2\pi}} > 0$$

and s.t. its variance is $O(1/L)$.

Proof Idea: A greedy approach. At each pixel, choose maximal local contrast between continuing *up* or to the *right*.

Lower Bound on Full Set of Curves

Lemma: Let I be a pure noise image. There exists a monotone curve $\Gamma = \Gamma(I)$ of length L, such that

$$\mathbb{E}_I[R(\Gamma(I))] = \frac{\sigma}{\sqrt{2\pi}} > 0$$

and s.t. its variance is $O(1/L)$.

Proof Idea: A greedy approach. At each pixel, choose maximal local contrast between continuing *up* or to the *right*.

Conclusion: *Cannot* detect any arbitrary edge.

In particular for exponentially large search spaces, lower limit on detectability.

Minimal Detectable Contrast

Lemma: Assume K_L feasible curves at length L. By simple union bound,

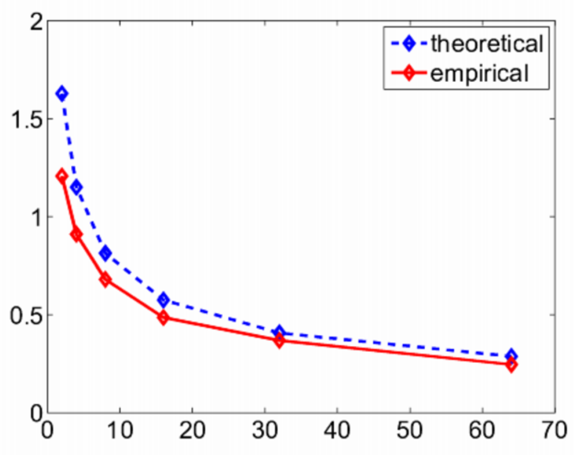
$$\boxed{T \leq \sigma\sqrt{\frac{2\ln(K_L/\alpha)}{wL}}}$$

Remarks:

- If K_L is exponential in L then $T \not\to 0$ as $L \to \infty$.
- If K_L is subexponential in L then $T \to 0$.
- If K_L independent of L, then $T \to 0$ as $1/\sqrt{L}$.

If feasible set $\mathcal{S}_L$ is sub-exponential in L
then asymptotically
any faint edge can be reliably detected if sufficiently long

Example: Straight Edges



L

Q2 - Computationally Efficient Algorithms

How can we efficiently compute all K_L responses ?

Q2 - Computationally Efficient Algorithms

How can we efficiently compute all K_L responses ?

Typically K_L scales (at-least) polynomially with image width n.

Naively going over all possible curves would be extremely slow.

Q2 - Computationally Efficient Algorithms

How can we efficiently compute all K_L responses ?

Typically K_L scales (at-least) polynomially with image width n.

Naively going over all possible curves would be extremely slow.

Key approach: Multi-scale construction.

Straight Lines

[Galun, Basri, Brandt 07']

For $n \times n$ image with $N = n^2$ pixels, there are $O(N^2)$ feasible straight line segments.

using multiscale construction by Brandt and Dym, *Fast calculation of multiple line integrals*, 1999.

- Efficiently compute dense sub-set of $O(N \ln N)$ line integrals.

- Via *hierarchical recursive calculation*, time complexity is $O(N \ln N)$, instead of $N^{3/2} \ln N$ of direct calculation.

works very well for noisy images with straight edges

Straight Lines

[Galun, Basri, Brandt 07']

For $n \times n$ image with $N = n^2$ pixels, there are $O(N^2)$ feasible straight line segments.

using multiscale construction by Brandt and Dym, *Fast calculation of multiple line integrals*, 1999.

- Efficiently compute dense sub-set of $O(N \ln N)$ line integrals.

- Via *hierarchical recursive calculation*, time complexity is $O(N \ln N)$, instead of $N^{3/2} \ln N$ of direct calculation.

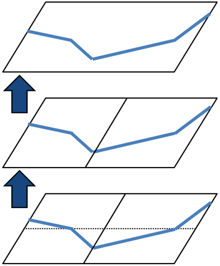
works very well for noisy images with straight edges

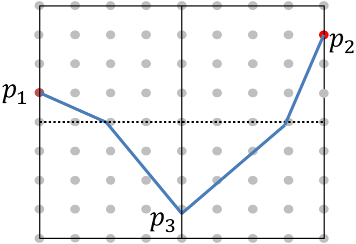
Limitation: In many images, edges are *curved*...

Rectangular Partition Tree

[Ofir, Galun, N. & Basri, 15']

Key idea: Recursive division of square to rectangle to smaller squares

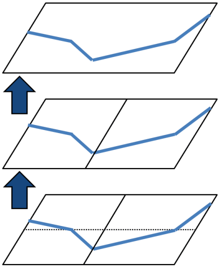


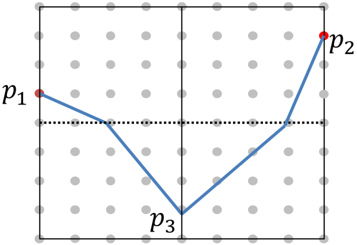


Rectangular Partition Tree

[Ofir, Galun, N. & Basri, 15']

Key idea: Recursive division of square to rectangle to smaller squares





Best edge between p_1 and p_2 : concatenate responses of edges $\Gamma(p_1, p_3)$ and $\Gamma(p_3, p_2)$.

Complexity of Rectangular Partition Tree

$f(A)$ - number of operations on tile of area A.

Divide tile into 2 sub-tiles, each area $A/2$, interface boundary length $O(\sqrt{A})$.

$$f(A) = 2f(A/2) + O(A^{1.5})$$

Complexity of Rectangular Partition Tree

$f(A)$ - number of operations on tile of area A.

Divide tile into 2 sub-tiles, each area $A/2$, interface boundary length $O(\sqrt{A})$.

$$f(A) = 2f(A/2) + O(A^{1.5})$$

Master Theorem: time complexity is $f(n \times n) = O(N^{1.5})$.

Complexity of Rectangular Partition Tree

With more detailed analysis, ($N = n^2 =$ total number of pixels)

$$\boxed{f(n \times n) \approx 18N^{3/2}}$$

Problem: This may still be too slow for large images.

Complexity of Rectangular Partition Tree

With more detailed analysis, ($N = n^2 =$ total number of pixels)

$$\boxed{f(n \times n) \approx 18N^{3/2}}$$

Problem: This may still be too slow for large images.

Insight: if edge is not extremely faint, don't need to go over all $\sqrt{A}$ pixels at interface. Can keep only top k highest responses. With this variant

$$\boxed{f(n \times n) \approx 6k \cdot N \log(N)}$$

Complexity of Rectangular Partition Tree

With more detailed analysis, ($N = n^2$ = total number of pixels)

$$\boxed{f(n \times n) \approx 18 N^{3/2}}$$

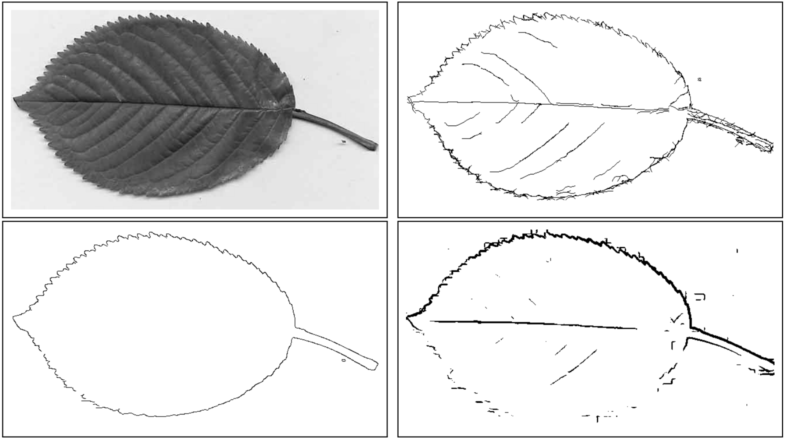
Problem: This may still be too slow for large images.

Insight: if edge is not extremely faint, don't need to go over all $\sqrt{A}$ pixels at interface. Can keep only top k highest responses. With this variant

$$\boxed{f(n \times n) \approx 6k \cdot N \log(N)}$$

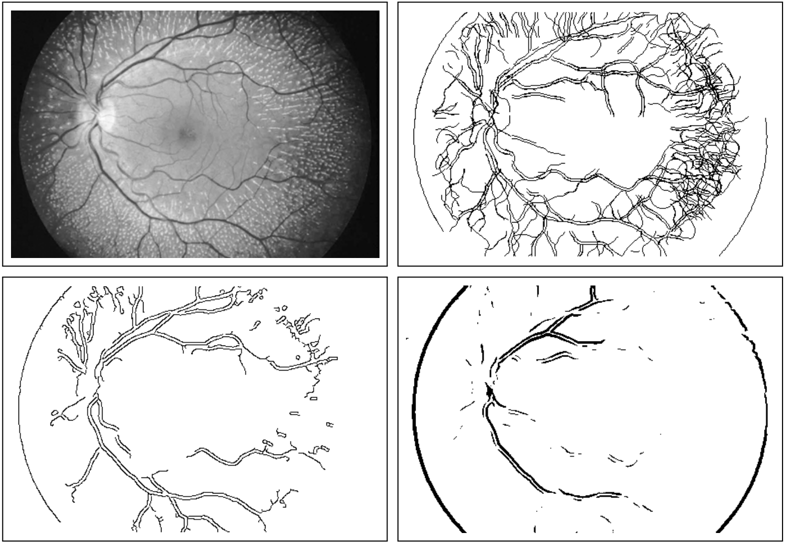
Empirically, 5 seconds on 256×256 image.
Significantly faster than previous methods based on quad-tree beamlets, whose time complexity is $O(N^2)$ or $O(N^{5/2})$.

Some Results



from top left clockwise: original, $O(N^{1.5})$ method, Canny, Crisp [14'].

Some Results



Q3: Severe Computational Constraints

In some applications: *large* and very noisy images (1000 x 1000 pixels or more) or noisy videos.

Q3: Severe Computational Constraints

In some applications: *large* and very noisy images (1000 x 1000 pixels or more) or noisy videos.

Task: Process Images/Video in *real time* **but**

Low power computing devices
or
Severe power constraints

Q3: Severe Computational Constraints

In some applications: *large* and very noisy images (1000 x 1000 pixels or more) or noisy videos.

Task: Process Images/Video in *real time* **but**

Low power computing devices
or
Severe power constraints

Examples:
- Battery of Cell-Phone
- Solar Power of distant surveillance camera
- Mobile Robots

In such cases even $O(N) = O(n^2)$ linear-time algorithm may be too slow.

Problem Setup

$I = I_0 + \xi$ observed $n \times n$ noisy image.
I_0 - noise free original image

Problem Setup

$I = I_0 + \xi$ observed $n \times n$ noisy image.
I_0 - noise free original image

Task: Detect edges in I_0 from noisy I.

Assumptions:

- Image I_0 contains *few* edges (sparsity).
- Edges of interest are **straight** and **sufficiently long.**

Example: Powerlines

Example: Canny, run-time 2.5sec

Example: Canny, run-time 2.5sec

Cannot detect faint powerlines of second tower

Example: Straight Segment Detector, run-time 5 min

Sublinear Time Edge Detection

Goal: Given noisy $n \times n$ image I, detect long straight edges in

sublinear time,

Sublinear Time Edge Detection

Goal: Given noisy $n \times n$ image I, detect long straight edges in

sublinear time,

complexity $O(n^\alpha)$ with $\alpha < 2$

Sublinear Time Edge Detection

Goal: Given noisy $n \times n$ image I, detect long straight edges in

sublinear time,

complexity $O(n^\alpha)$ with $\alpha < 2$

touching only a fraction of the image/video pixels!

Sublinear Time Edge Detection

Goal: Given noisy $n \times n$ image I, detect long straight edges in

sublinear time,

complexity $O(n^{\alpha})$ with $\alpha < 2$

touching only a fraction of the image/video pixels!

Questions:

a) Statistical: which edge strengths can one detect vs. α ?

b) Computational: optimal sampling scheme ?

c) Practical: *sub-linear* time algorithm ?

Previous Sub-linear time methods

[Xu, Oja, and Kultanan 90']
[Kiryati et. al, 91']

Randomized / Probabilistic Hough transforms

Previous Sub-linear time methods

[Xu, Oja, and Kultanan 90']
[Kiryati et. al, 91']

Randomized / Probabilistic Hough transforms

Based on local gradients, cannot in general detect faint edges.

Also, not designed to detect start and end points of edges that do not span whole image.

Optimal Sublinear Edge Detection

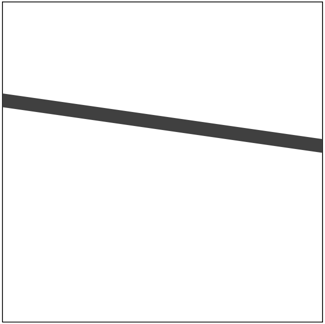
For theoretical analysis, consider following class of images:

$$\mathcal{I} = \{I \text{ contains only noise or one long fiber plus noise}\}$$

Optimal Sublinear Edge Detection

For theoretical analysis, consider following class of images:

$$\mathcal{I} = \{I \text{ contains only noise or one long fiber plus noise}\}$$



Fundamental Limitations / Design Principles

Focus on detection under worst-case scenario.

Fundamental Limitations / Design Principles

Focus on detection under worst-case scenario.

Lemma: If number of observed pixels is n^α with $\alpha < 1$ then there exists $I \in \mathcal{I}$ whose edges cannot be detected.

Fundamental Limitations / Design Principles

Focus on detection under worst-case scenario.

Lemma: If number of observed pixels is n^α with $\alpha < 1$ then there exists $I \in \mathcal{I}$ whose edges cannot be detected.

Theorem: Assume number of observed pixels is s and s/n is integer. Then,
i) any optimal sampling scheme must observe exactly s/n pixels per row.
ii) sampling s/n whole columns is an optimal scheme.

Statistical Accuracy vs. Computational Complexity

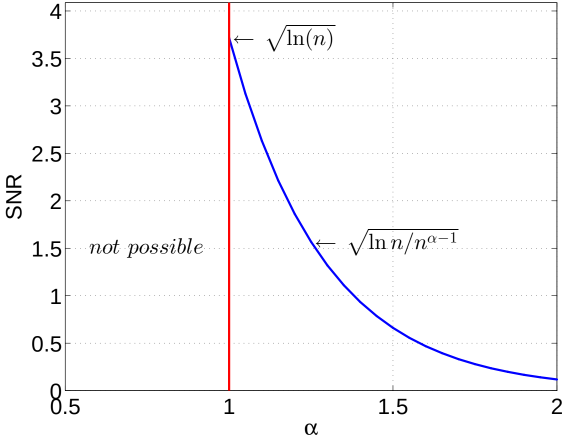
Definition: Edge SNR = edge contrast / noise level.

Statistical Accuracy vs. Computational Complexity

Definition: Edge SNR = edge contrast / noise level.

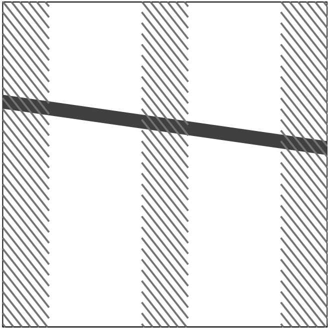
Theorem: At complexity $O(n^\alpha)$, with $\alpha \geq 1$,

$$\text{SNR} \gtrsim \sqrt{\ln n / n^{\alpha-1}}$$



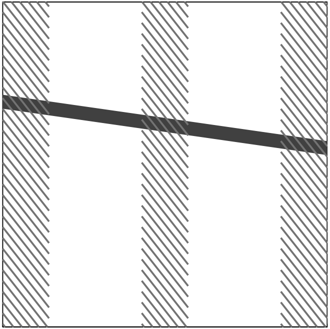
Sublinear Edge Detection Algorithm

Key Idea: Sample few image strips



Sublinear Edge Detection Algorithm

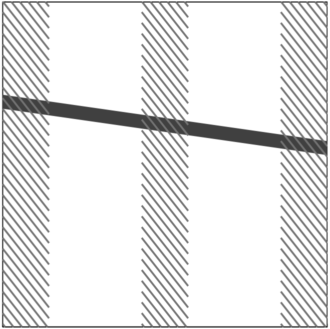
Key Idea: Sample few image strips



first detect edges in strips

Sublinear Edge Detection Algorithm

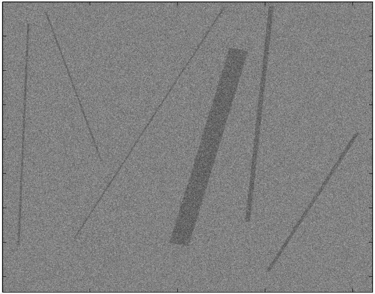
Key Idea: Sample few image strips



first detect edges in strips
next: non-maximal suppression, edge localization

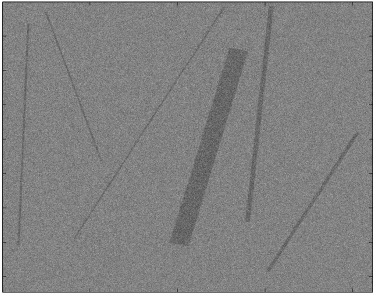
Example:

NOISY IMAGE, SNR=1

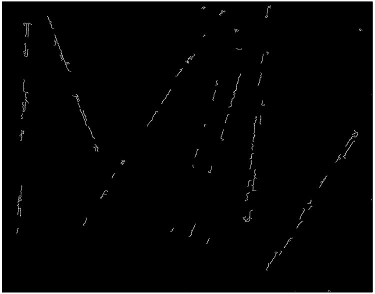


Example:

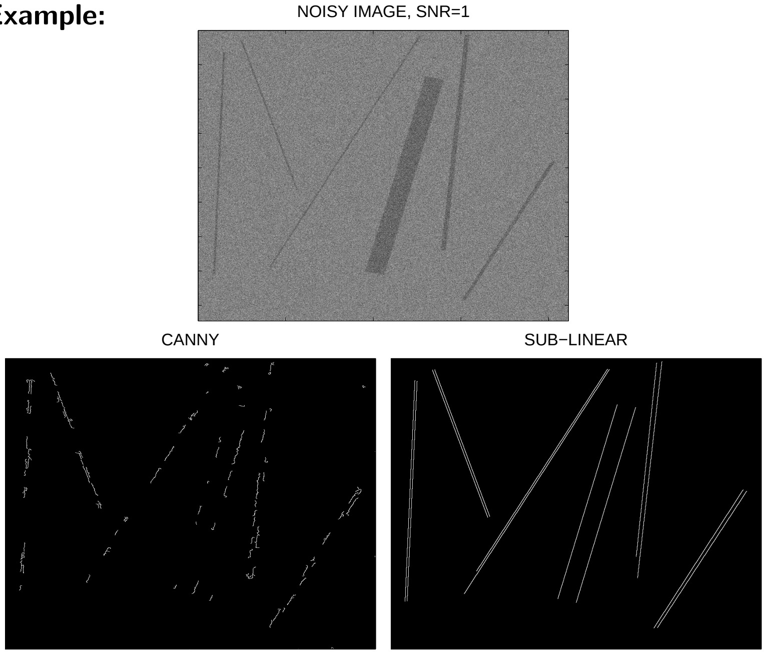
NOISY IMAGE, SNR=1



CANNY



Example:



Sublinear Edge Detection, run-time few seconds

Summary

Presented statistical theory and lower bounds for edge detection.

Fast $O(N \log N)$ algorithm for detection of faint curved edges.

Sublinear algorithm for detection of long straight edges.

Summary

Presented statistical theory and lower bounds for edge detection.

Fast $O(N \log N)$ algorithm for detection of faint curved edges.

Sublinear algorithm for detection of long straight edges.

Current / future work: extension to sublinear detection of curved edges. detection of fibers in 3-D.

Summary

Presented statistical theory and lower bounds for edge detection.

Fast $O(N \log N)$ algorithm for detection of faint curved edges.

Sublinear algorithm for detection of long straight edges.

Current / future work: extension to sublinear detection of curved edges. detection of fibers in 3-D.

General question/challenge: what image processing/machine learning tasks can be performed in sub-linear time, what are the statistical-computational tradeoffs ?

Relevant Papers

1) Galun, Basri, Brandt, Multiscale edge detection and fiber enhancement using differences of oriented means, ICCV (2007).

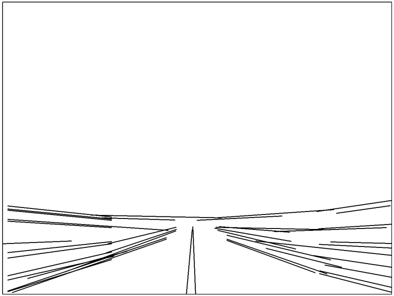
2) Alpert, Galun, Nadler, Basri, Detecting Faint Edges in Noisy Images, ECCV 2010.

3) Ofir, Galun, Nadler Basri, Fast detection of curved edges at low SNR, submitted, 2015.

4) Horev, Arias-Castro, Nadler, Edge Detection in Sub-linear Time, SIAM J. Imaging Sciences, 2015.

The End

Research is a very long path.



Thank you !

www.wisdom.weizmann.ac.il/∼nadler/