# Edge Detection on a Computational Budget: A Sub-linear Time Approach


Boaz Nadler

The Weizmann Institute of Science

Joint work with
Inbal Horev, Ronen Basri, Meirav Galun (WIS)
and
Ery Arias-Castro (UCSD)


May 2014

# Introduction

this work is part of ICRI project

**Understanding and Utilizing**
**Natural Image Statistics**

# Introduction

this work is part of ICRI project

**Understanding and Utilizing**
**Natural Image Statistics**

**Team Members:**

- Yair Weiss, Hebrew University
- Miki Elad, Technion
- Anat Levin, Weizmann Institute
- Boaz Nadler, Weizmann Institute

# Introduction

this work is part of ICRI project

**Understanding and Utilizing**
**Natural Image Statistics**

**Team Members:**

- Yair Weiss, Hebrew University
- Miki Elad, Technion
- Anat Levin, Weizmann Institute
- Boaz Nadler, Weizmann Institute

Main Goals: Better understand natural image statistics and how they can be used in analysis and design of improved algorithms for image processing and computer vision.

# Edge Detection

A fundamental task in low level image processing.

# Edge Detection

A fundamental task in low level image processing.

well studied problem, many algorithms

well understood theory.

# Edge Detection at low SNR

**Our Focus:**
Edge detection in **very noisy** 2D images and 3D video

**Motivation(s):**

1. Images at non-ideal conditions: poor lighting, fog, rain, night.
2. surveillance applications
3. Real time object tracking in 3D video

# Edge Detection at low SNR

**Our Focus:**
Edge detection in **very noisy** 2D images and 3D video

**Motivation(s):**

1. Images at non-ideal conditions: poor lighting, fog, rain, night.
2. surveillance applications
3. Real time object tracking in 3D video

**Image Prior:**
- Interested in long straight edges.

# Example: Powerlines

# Traditional Edge Detection Algorithms

**Typical Approach:** Detect edges from *local* image gradients.

Example: Canny Edge Detector, complexity $O(n^2)$

*linear* in total number of image pixels
fast, suitable for real-time

**Limitation: Does not work well at low SNR**

# Example: Canny, run-time 2.5sec

# Example: Canny, run-time 2.5sec

Cannot detect faint powerlines of second tower

# Modern Sophisticated Methods

[Brandt, Galun, Basri, 07]
[Alpert, Galun, Nadler, Basri, 10]

- Statistical theory for limits of detectability.

- (Theoretically) Efficient multiscale algorithms, robust to noise

# Modern Sophisticated Methods

[Brandt, Galun, Basri, 07]
[Alpert, Galun, Nadler, Basri, 10]

- Statistical theory for limits of detectability.
- (Theoretically) Efficient multiscale algorithms, robust to noise

and yet

*slow*

# Modern Sophisticated Methods

[Brandt, Galun, Basri, 07]
[Alpert, Galun, Nadler, Basri, 10]

- Statistical theory for limits of detectability.
- (Theoretically) Efficient multiscale algorithms, robust to noise

and yet

*slow*

Run time: $O(min)$ for large images, $O(hours)$ for video.

# Example: Straight Segment Detector, run-time 19min

# Challenge: Sublinear Time Edge Detection

**Goal:** Devise extremely fast edge detection algorithm, that is also robust to noise.

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

*sublinear* time

# Challenge: Sublinear Time Edge Detection

**Goal:** Devise extremely fast edge detection algorithm, that is also robust to noise.

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

*sublinear* time

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

# Challenge: Sublinear Time Edge Detection

**Goal:** Devise extremely fast edge detection algorithm, that is also robust to noise.

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!

# Challenge: Sublinear Time Edge Detection

**Goal:** Devise extremely fast edge detection algorithm, that is also robust to noise.

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. $\alpha$ ?

b) Computational: optimal sampling scheme ?

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

# Why Sub-linear ?

In various applications:

*Large* and very noisy images (1000 x 1000 pixels or more)
video sequences much larger ($\sim$30 frames/sec)

# Why Sub-linear ?

In various applications:

*Large* and very noisy images (1000 x 1000 pixels or more)
video sequences much larger ($\sim$30 frames/sec)

Task: Process in *real time* **but** with

Low power computing devices
or
Severe power constraints

# Why Sub-linear ?

In various applications:

*Large* and very noisy images (1000 x 1000 pixels or more)
video sequences much larger ($\sim$30 frames/sec)

Task: Process in *real time* **but** with

Low power computing devices
or
Severe power constraints

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

# Problem Setup

Observe $n \times n$ noisy image

$$I = I_0 + \xi$$

$I_0$ - noise free original image
$\xi$ - additive noise, assume $\xi_{ij}$ are i.i.d. zero mean, variance $\sigma^2$.

# Problem Setup

Observe $n \times n$ noisy image

$$I = I_0 + \xi$$

$I_0$ - noise free original image
$\xi$ - additive noise, assume $\xi_{ij}$ are i.i.d. zero mean, variance $\sigma^2$.

**Goal:** Detect edges in $I_0$ from noisy $I$.

## Problem Setup

Observe $n \times n$ noisy image

$$I = I_0 + \xi$$

$I_0$ - noise free original image
$\xi$ - additive noise, assume $\xi_{ij}$ are i.i.d. zero mean, variance $\sigma^2$.

**Goal:** Detect edges in $I_0$ from noisy $I$.

**Definition:** *Edge Signal to Noise Ratio* $=$ edge contrast$/\sigma$.

# Problem Setup

Observe $n \times n$ noisy image

$$I = I_0 + \xi$$

$I_0$ - noise free original image
$\xi$ - additive noise, assume $\xi_{ij}$ are i.i.d. zero mean, variance $\sigma^2$.

**Goal:** Detect edges in $I_0$ from noisy $I$.

**Definition:** *Edge Signal to Noise Ratio* = edge contrast$/\sigma$.

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

# Theoretical Questions

Given sub-linear budget:

1) what are *optimal* sampling schemes ?

2) what are fundamental limitations on sub-linear edge detection ?

3) what is the tradeoff between *statistical accuracy* and *computational complexity* ?

# Optimal Sublinear Edge Detection

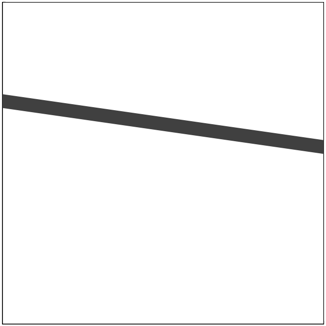
Consider set of possible images:

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

# Optimal Sublinear Edge Detection

Consider set of possible 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.

**Theorem:** If number of observed pixels is $n^\alpha$ 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.

**Theorem:** If number of observed pixels is $n^\alpha$ 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 observes exactly $s/n$ pixels per row.
ii) sampling s/n whole columns is an optimal scheme.

# Statistical Accuracy vs. Computational Complexity

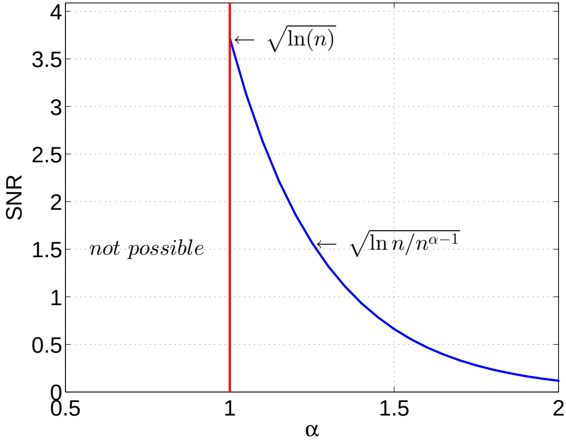
**Definition:** Edge SNR = edge contrast / noise level.

# Statistical Accuracy vs. Computational Complexity

**Definition:** Edge SNR = edge contrast / noise level.

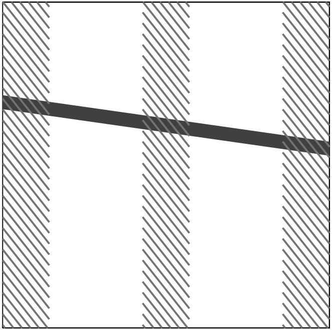
**Theorem:** At complexity $O(n^\alpha)$, with $\alpha \geq 1$,

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



# Sublinear Edge Detection Algorithm

**Key Idea:** Sample fraction of image strips

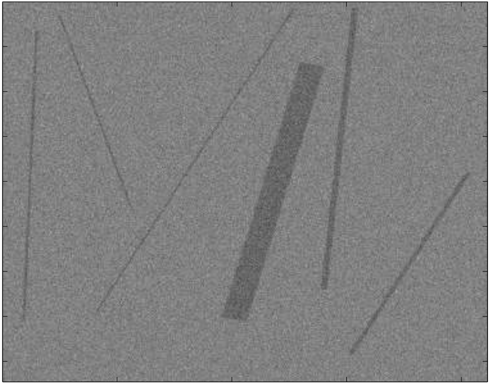


# Sublinear Edge: Postprocessing

- Non-maximal suppression
- Edge Validation
- Edge Localization

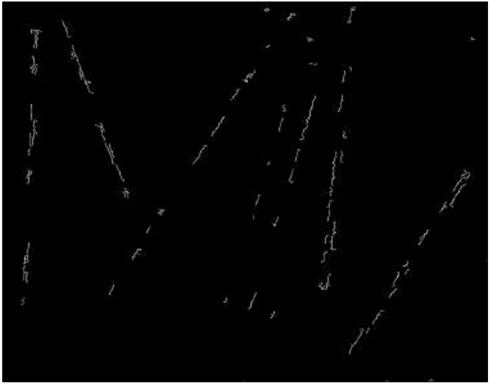
# Example

NOISY IMAGE, SNR=1



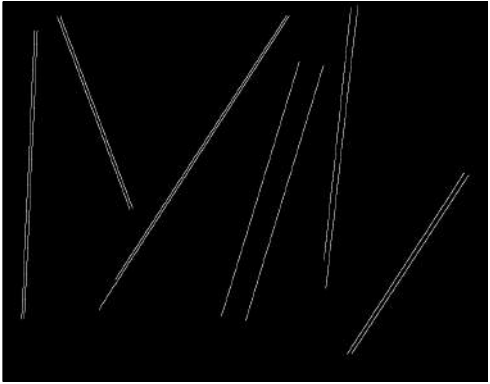
# Example

CANNY



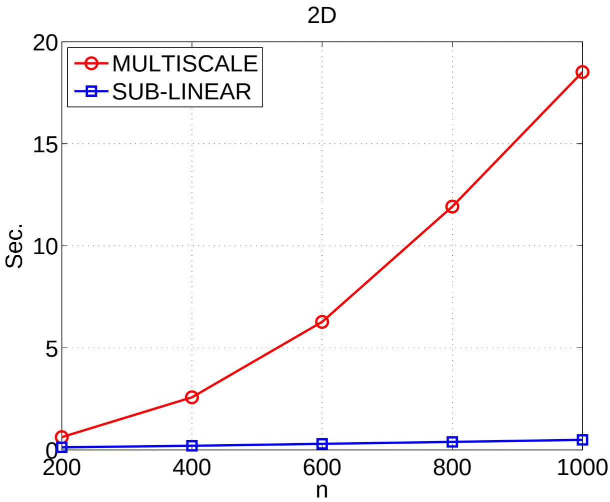
# Example

SUB-LINEAR



# Run Time / 2-D Images

Actual results / Matlab / Preliminary Implementation:



# Sublinear Time Edge Detection, run-time few seconds

# Open Questions / Tasks

- Theory: Better understanding of statistical vs. computational tradeoff for other problems in image analysis, computer vision, and "big-data" in general.

- Develop Robust Real-Time Algorithm in 2-D (beyond proof-of-concept).

- Development of theory and sub-linear algorithm in 3-D.

- Potential applications.

# Open Questions / Tasks

- Theory: Better understanding of statistical vs. computational tradeoff for other problems in image analysis, computer vision, and "big-data" in general.

- Develop Robust Real-Time Algorithm in 2-D (beyond proof-of-concept).

- Development of theory and sub-linear algorithm in 3-D.

- Potential applications.

*Thank You / The End*