LSD: a Line Segment Detector

rafael grompone von gioi

Cachan, September 30, 2011

Outline

- What is LSD, demo
- Overview
- Theoretical background
- Details of the algorithm
- Projects proposed

LSD

- LSD is a Line Segment Detector
- It is based in Burns, Hanson, and Riseman method
- It uses a false detection control based on Desolneux, Moisan, & Morel's theory.
- LSD is fast, produces precise results, and controls false detections.

Resources

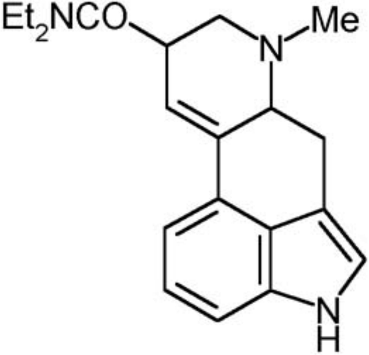
Google: lsd + grompone

Google: lsd + morel

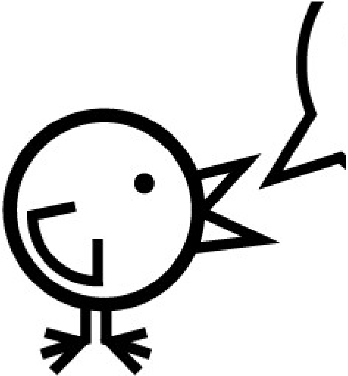
www.ipol.im → LSD: A LINE SEGMENT DETECTOR

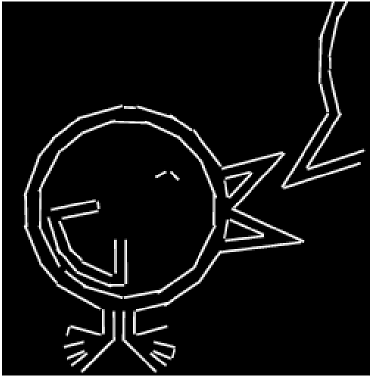
Examples

Examples

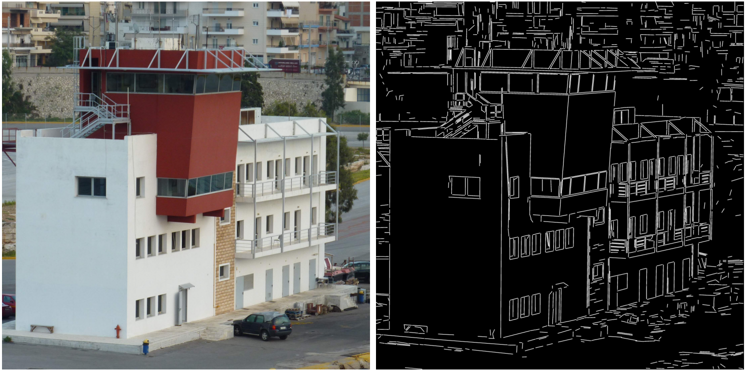


Examples



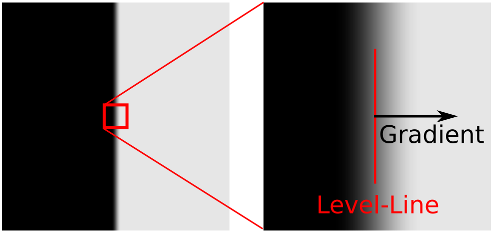


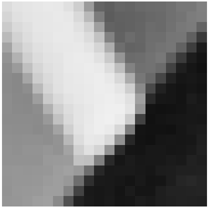
Examples



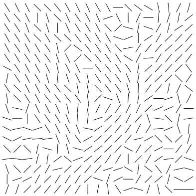
Overview

Gradient and Level-Line Field





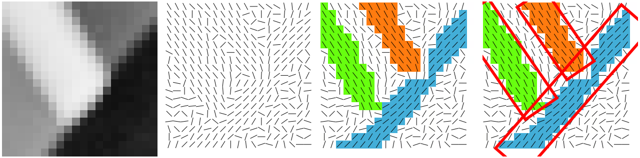
image



level-line field

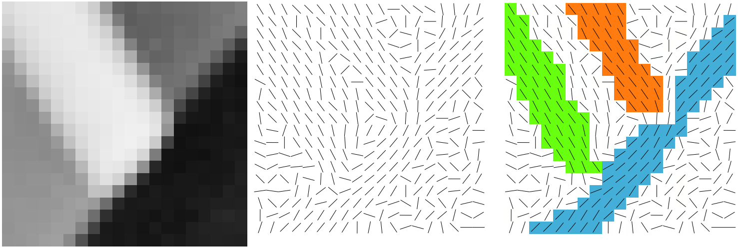
LSD in 4 steps

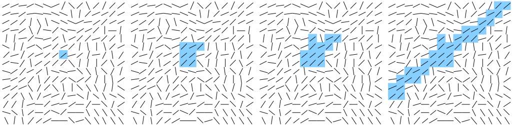
1. Compute the level-line field
2. Partition the image into groups of connected pixels that share the same level-line angle up to a certain tolerance
3. Find rectangular approximations
4. Validation



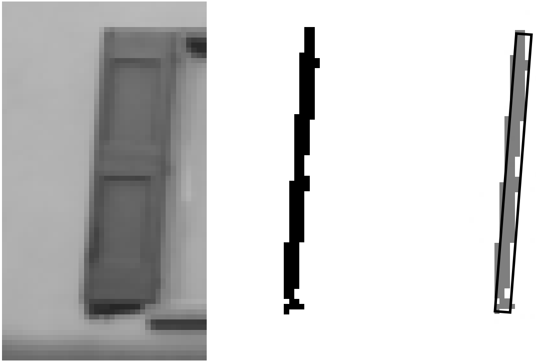
Line-Support Regions

A group of connected pixels that share the same level-line angle up to a certain tolerance τ.



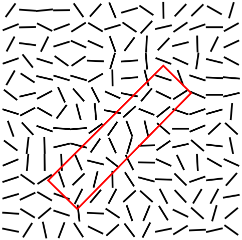


Rectangular Approximation of Regions

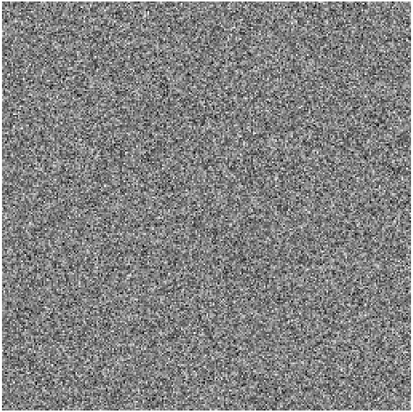


- Pixel's mass is proportional to the gradient modulus
- Region's center of mass $\longrightarrow$ rectangle's center
- First inertia axis of the region $\longrightarrow$ rectangle's angle
- Length and width to envelope most of region's mass

Validation



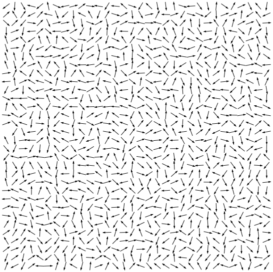
Helmholtz Principle



There is no perception on noise.

A Contrario Detection [Desolneux, Moisan, Morel]

Structure is detected as outliers of a noise model H_0:



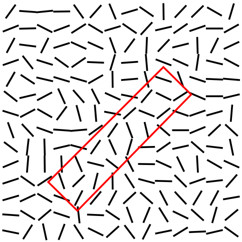
Non-Structured Level-Line Orientations:

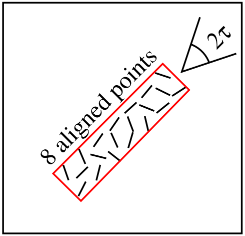
- angles are independent random variables
- uniformly distributed in $[0, 2\pi]$

More precisely: an observed geometric structure becomes meaningful when the expectation of its number of occurrences is very small in the non-structured data model.

Aligned Point

A point whose level-line angle is equal to the rectangle angle up to a certain tolerance τ.





$k(r, i)$ is the number of aligned points of rectangle r in image i.
$n(r)$ is the total number of pixels in the rectangle r.
In the example, $k(r, i) = 8$ and $n(r) = 27$.

Meaningful Rectangle

Given a rectangle r with $k(r,i)$ observed aligned points, we define

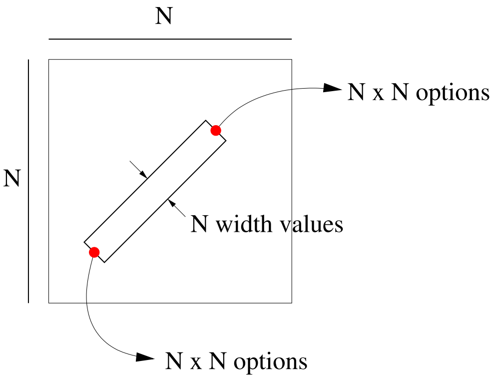
$$\mathrm{NFA}(r,i) = N_{test} \cdot P_{H_0}\left[k(r,I) \geq k(r,i)\right]$$

where:
I is a random image on H_0,
N_{test} is the number of tests.

$\mathrm{NFA}(r,i)$ is the expected number of event as good as (r,i) in H_0.
When $\mathrm{NFA}(r,i)$ is large: a common event in H_0 and not meaningful.
When $\mathrm{NFA}(r,i)$ is small: a rare event in H_0 and probably meaningful.

A rectangle with $\mathrm{NFA}(r,i) \leq \varepsilon$ is called ε-meaningful rectangle.

Number of Tests



$$N_{test} = N^5$$

Probability term

In H_0, the probability that a pixel is an aligned point is

$$p = \frac{\tau}{\pi}.$$

Because of the independence in H_0, $k(r, I)$ follows a binomial distribution. Then,

$$P_{H_0}\left[k(r, I) \geq k(r, i)\right] = B\big(n(r), k(r, i), p\big)$$

where $B(n, k, p)$ is the tail of the binomial distribution:

$$B(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

NFA

The final expression for the Number of False Alarms for a rectangle is:

$$\mathrm{NFA}(r,i) = N^5 \cdot \sum_{j=k(r,i)}^{n(r)} \binom{n(r)}{j} p^j (1-p)^{n(r)-j}$$

Theorem

$$E_{H_0}\left[\sum_{r\in\mathcal{R}} \mathbb{1}_{\mathrm{NFA}(r,I)\leq\varepsilon}\right] \leq \varepsilon$$

where E is the expectation operator, $\mathbb{1}$ is the indicator function, $\mathcal{R}$ is the set of rectangles considered, and I is a random image in H_0.

The theorem states that the average number of ε-meaningful rectangles on the a contrario model H_0 images is less than ε.

In other words, it shows that LSD satisfies the Helmholtz principle.

Proof

We define $\hat{k}(r)$ as

$$\hat{k}(r) = \min \left\{ n \in \mathbb{N},\ P_{H_0}\left[k(r,l) \geq n\right] \leq \frac{\varepsilon}{N^5} \right\}.$$

Then, $\mathrm{NFA}(r,i) \leq \varepsilon$ is equivalent to $k(r,i) \geq \hat{k}(r)$. Now,

$$E_{H_0}\left[\sum_{r\in\mathcal{R}} \mathbb{1}_{\mathrm{NFA}(r,l)\leq\varepsilon}\right] = \sum_{r\in\mathcal{R}} P_{H_0}\left[\mathrm{NFA}(r,l) \leq \varepsilon\right] = \sum_{r\in\mathcal{R}} P_{H_0}\left[k(r,l) \geq \hat{k}(r)\right].$$

But, by definition of $\hat{k}(r)$ we know that

$$P_{H_0}\left[k(r,l) \geq \hat{k}(r)\right] \leq \frac{\varepsilon}{N^5}$$

and using that $\#\mathcal{R} = N^5$ we get

$$E_{H_0}\left[\sum_{r\in\mathcal{R}} \mathbb{1}_{\mathrm{NFA}(r,l)\leq\varepsilon}\right] \leq \sum_{r\in\mathcal{R}} \frac{\varepsilon}{N^5} = \varepsilon.$$

$\varepsilon = 1$

The result is not very sensitive to the value of ε.

image | $\varepsilon = 1$ | $\varepsilon = 0.1$ | $\varepsilon = 0.01$

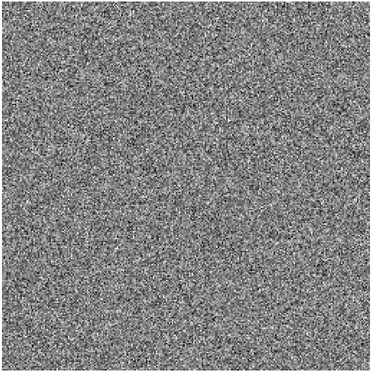
$\varepsilon = 1$ means, on average, one false detection per image.

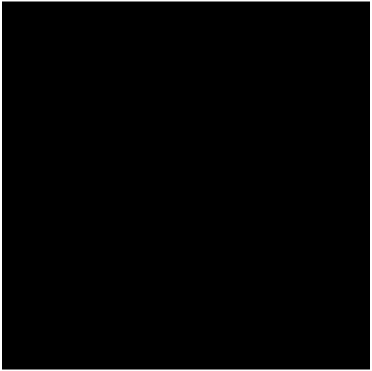
Algorithm Summary

1. Partition the image into Line-Support Regions
2. For each Line-Support Region:
3. Find the Rectangular Approximation
4. Compute NFA value
5. Rectangles with $\mathrm{NFA} \leq 1$ are added to the output.

more examples

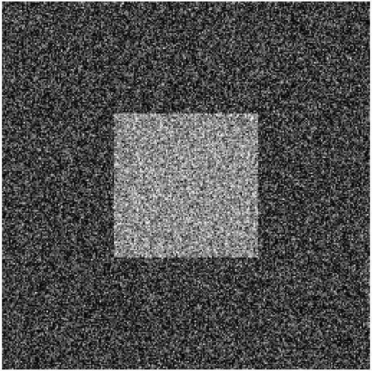
Examples

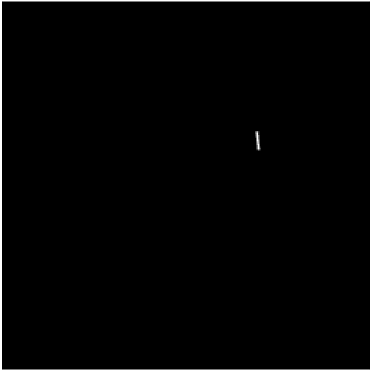




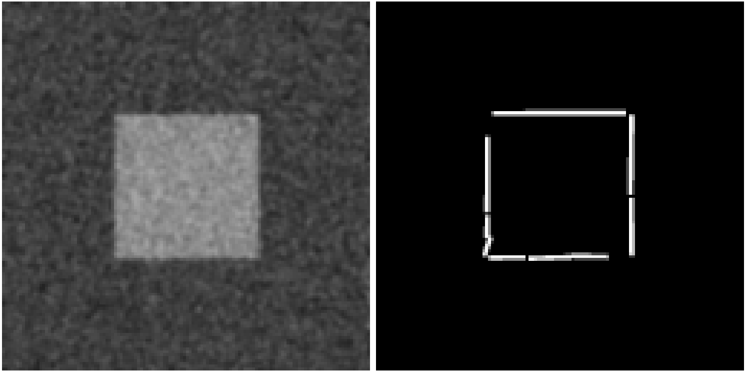
Examples

Examples

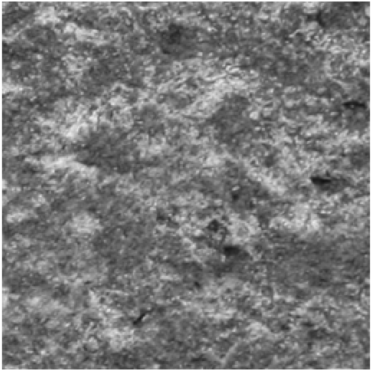
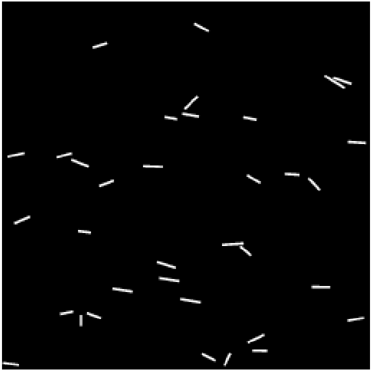




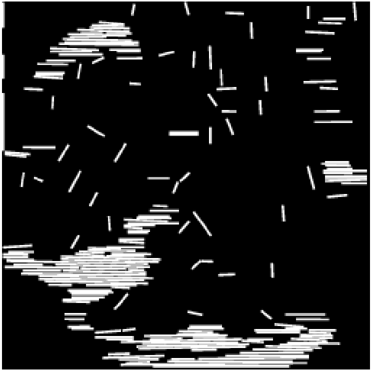
Examples



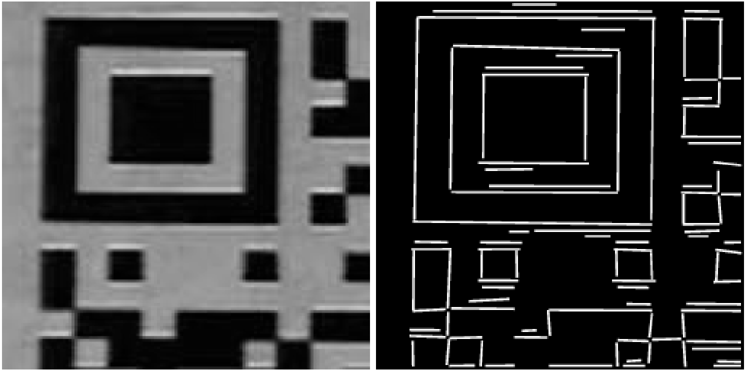
Examples

Examples



Examples



Examples

Details of the algorithm

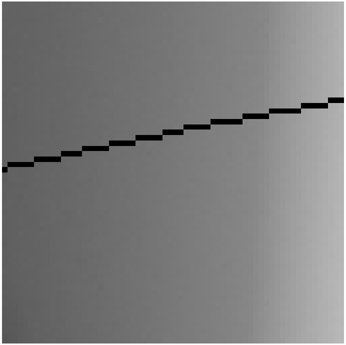
LSD

1. Scale the input image to scale S ($\sigma = \Sigma/S$).
2. Compute level-lines field.
3. List pixels by decreasing gradient magnitude.
4. Set STATUS(every pixel) to NOT USED.
5. Remove pixels where gradient magnitude $\leq \rho$.
6. From the next pixel P in the list with STATUS(P)=NOT USED:
    7. Grow region from P of NOT USED connected pixels that share level-line angle, tolerance τ. Mark pixels in the region as USED.
    8. Compute the rectangular approximation.
    9. Cut region until aligned point density $> D$.
    10. Compute NFA value.
    11. Try to improve rectangle.
    12. If $\mathrm{NFA} \leq \varepsilon$, detection!

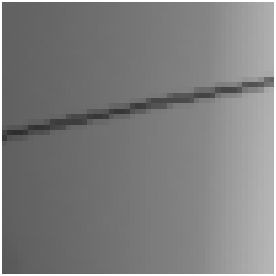
Parameters S, Σ, ρ, τ, D, and ε.

Input image scaling

Staircase problem:



input image



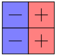
80% scaling

80% scale, then $S = 0.8$. Gaussian sub-sampling with $\sigma = \Sigma / S$

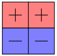
Good balance between blur and aliasing: $\Sigma = 0.6$

Compute level-line field

The gradient is computed at $(x, y) + (1/2, 1/2)$ by



$$g_x(x, y) = \frac{i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1)}{2},$$



$$g_y(x, y) = \frac{i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y)}{2}.$$

The level-line angle is computed as

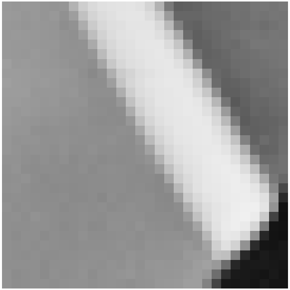
$$\arctan\left(\frac{g_x(x, y)}{-g_y(x, y)}\right)$$

and the gradient norm as

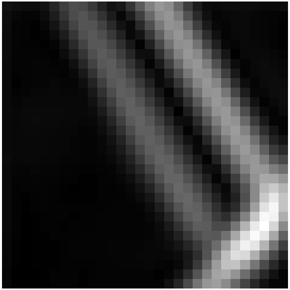
$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)}.$$

Where $i(x, y)$ is the image value at coordinates (x, y).

Pseudo-ordering of pixels



image

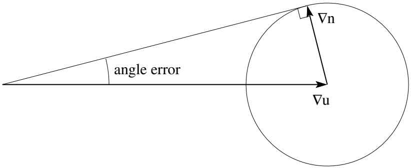


gradient magnitude

Starting from pixels of high gradient magnitude, seed points are near the center of edges.

Sorting cannot be done in linear time. Instead, a pseudo-ordering is performed by classifying pixels in 1024 bin of gradient values.

Gradient threshold



$$\tilde{i} = i + n \qquad \nabla\tilde{i} = \nabla i + \nabla n,$$

where n is the quantization noise.

$$|\text{angle error}| \leq \arcsin\left(\frac{q}{|\nabla i|}\right),$$

where q is a bound to $|\nabla n|$. Imposing $|\text{angle error}| \leq \tau$ we get

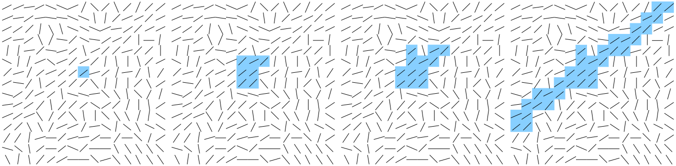
$$\rho = \frac{q}{\sin\tau}.$$

$q = 2$, maximum gradient quantization error in $[0, 255]$ images.

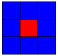
Region Growing

Region's angle:

$$\theta_{region} = \arctan\left(\frac{\sum_j \sin(\text{level-line-angle}_j)}{\sum_j \cos(\text{level-line-angle}_j)}\right) \qquad j \in \text{region}.$$



Recursively, the unused neighbors Q are added if

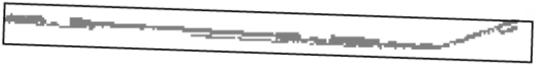


$$\left|\text{level-line-angle}(Q) - \theta_{region}\right|_{\text{mod}2\pi} < \tau.$$

$\tau = 22.5$ degree.

Angle problem

If two line segments for an angle of $180 - \tau$ we get:



We define the density of *aligned points* as

$$d = \frac{k}{\mathrm{length}(r) \cdot \mathrm{width}(r)}$$

where k is the number of aligned points in the region.

The region is repeatedly cut until $d > D$ or there are no points left.

$D = 0.7$ is an empirical value.

Improve rectangle

Before rejecting a region as not meaningful, some variations to the rectangle are tried:

1. try finer *precisions p*
2. try to reduce width
3. try to reduce one side of the rectangle
4. try to reduce the other side of the rectangle
5. try even finer precisions

Parameters

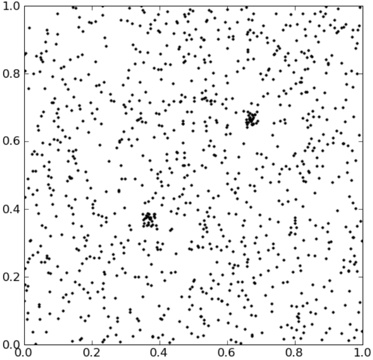
$S = 0.8$, staircase effect
$\Sigma = 0.6$, blur/aliasing balance
$\rho = q/\sin\tau$, $q = 2$, quantization noise
$\tau = 22.5$ degree, empirical but near optimum
$D = 0.7$, empirical
$\varepsilon = 1$, *a contrario* framework

Only *D* has an arbitrary value and determines how curves are approximated.

More details: www.ipol.im

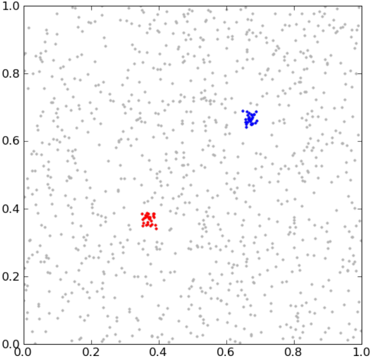
Projects

Project 1: *a contrario* clustering



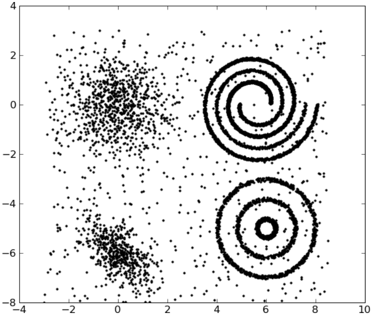
"Meaningful Clustered Forest: an Automatic and Robust Clustering Algorithm" by Mariano Tepper, Pablo Musé, & Andrés Almansa

Project 1: *a contrario* clustering



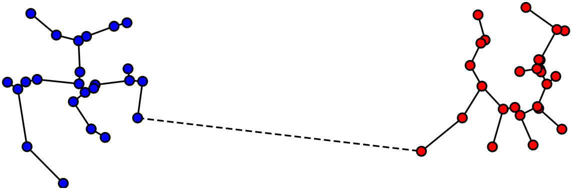
“Meaningful Clustered Forest: an Automatic and Robust Clustering Algorithm” by Mariano Tepper, Pablo Musé, & Andrés Almansa

Project 1: *a contrario* clustering



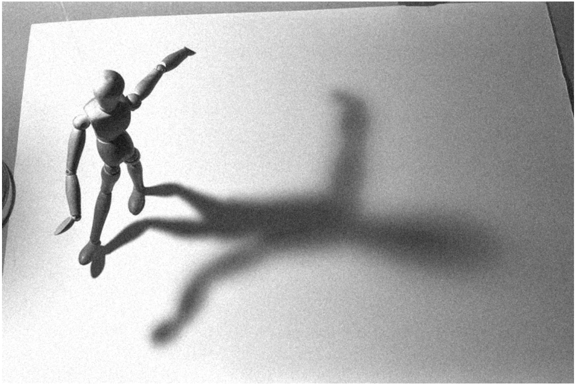
“Meaningful Clustered Forest: an Automatic and Robust Clustering Algorithm” by Mariano Tepper, Pablo Musé, & Andrés Almansa

Project 1: *a contrario* clustering

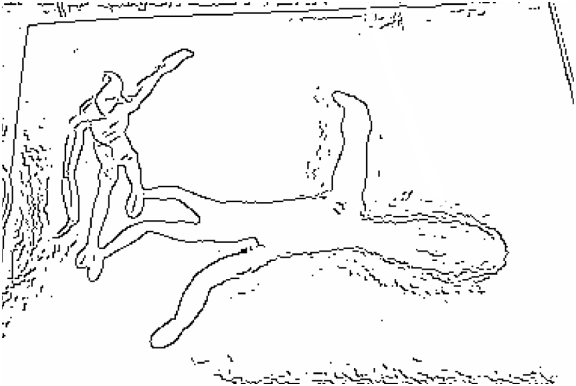


"Meaningful Clustered Forest: an Automatic and Robust Clustering Algorithm" by Mariano Tepper, Pablo Musé, & Andrés Almansa

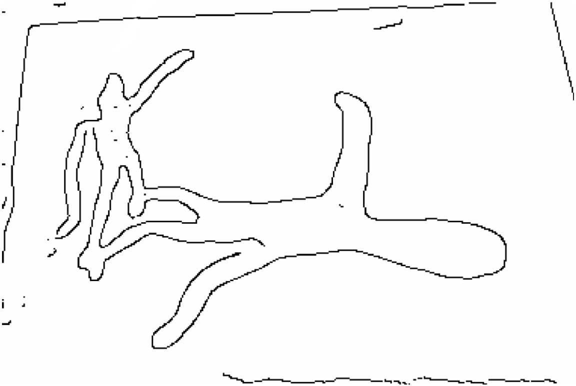
Project 2 & 3: parameterless edge detection



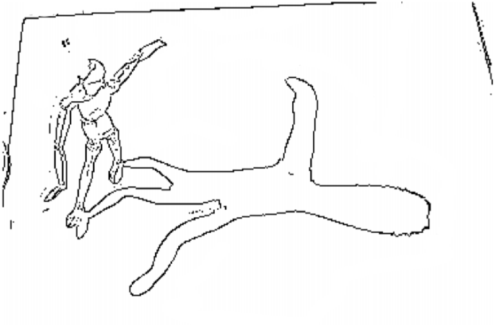
Project 2 & 3: parameterless edge detection



Project 2 & 3: parameterless edge detection

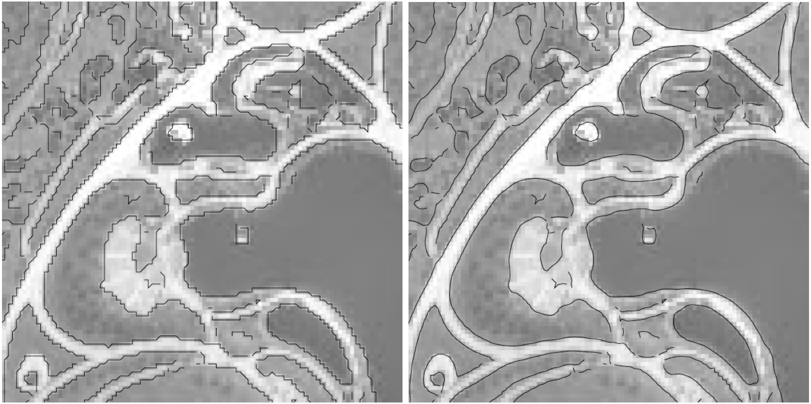


Project 2 & 3: parameterless edge detection



1. “Local Scale Control for Edge Detection and Blur Estimation” by James H. Elder & Steven W. Zucker
2. “Multiscale Edge Detection and Fiber Enhancement Using Differences of Oriented Means” by Meirav Galun, Ronen Basri & Achi Brandt

Project 4: Devernay's sub-pixel edge detector



"A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy" by Frédéric Devernay

IPOL Publication

- Detailed description of the algorithm
- Good quality code: standard and well commented
- A running demonstration

video

merci