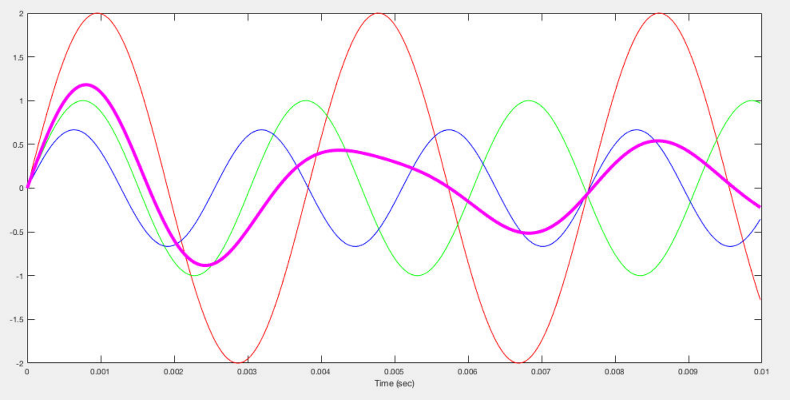
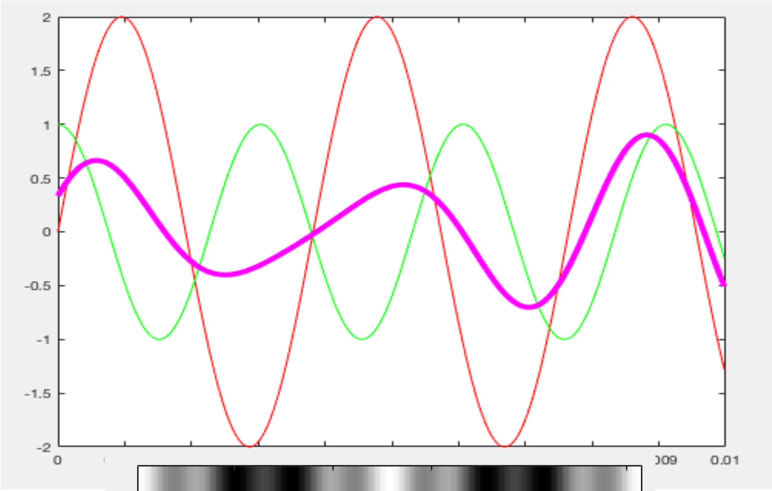
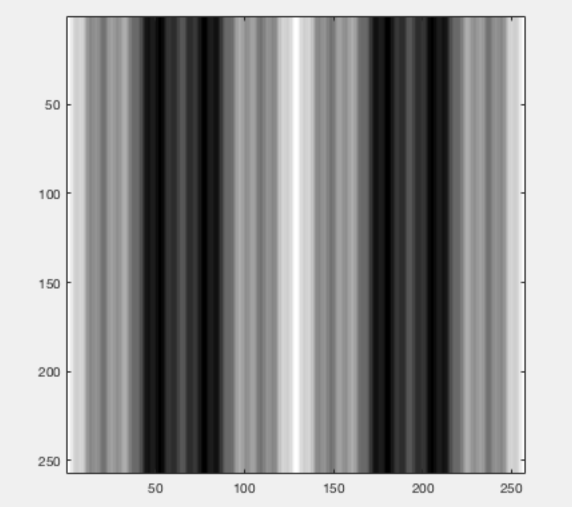
Video 4.1
Kostas Daniilidis

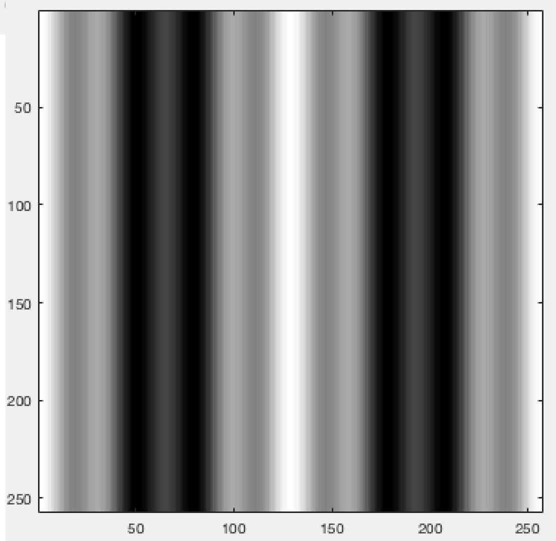
Filter selectivity

Frequency-selective filters



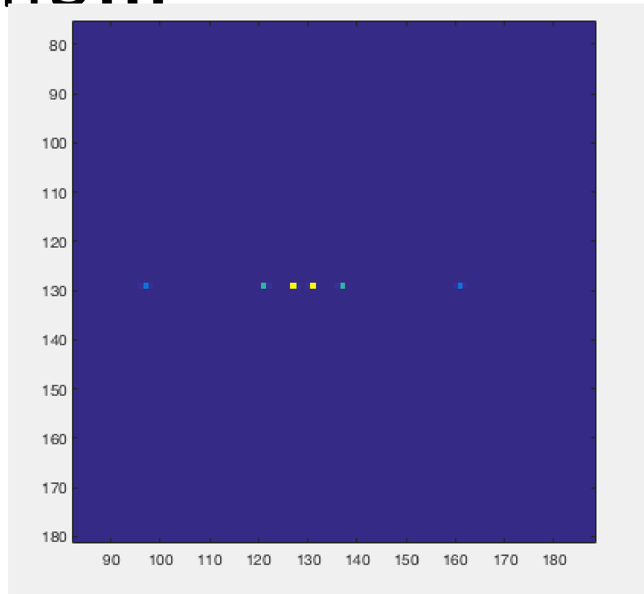




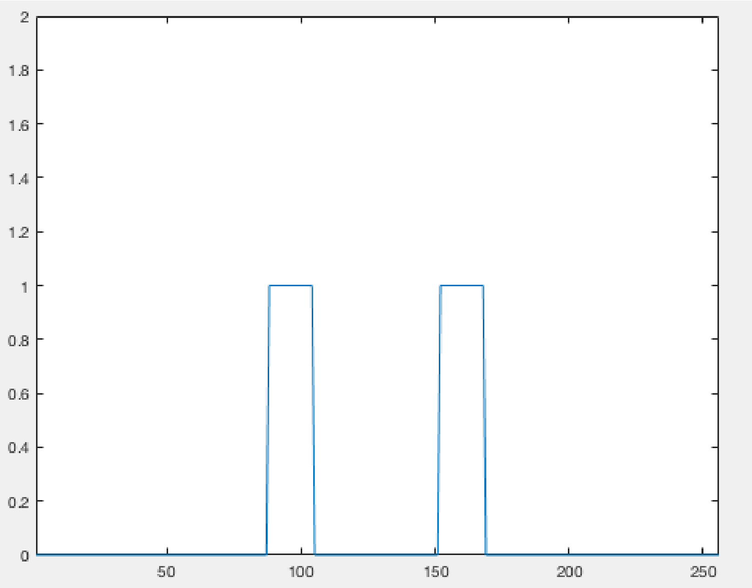


Property of Penn Engineering, Kostas Daniilidis

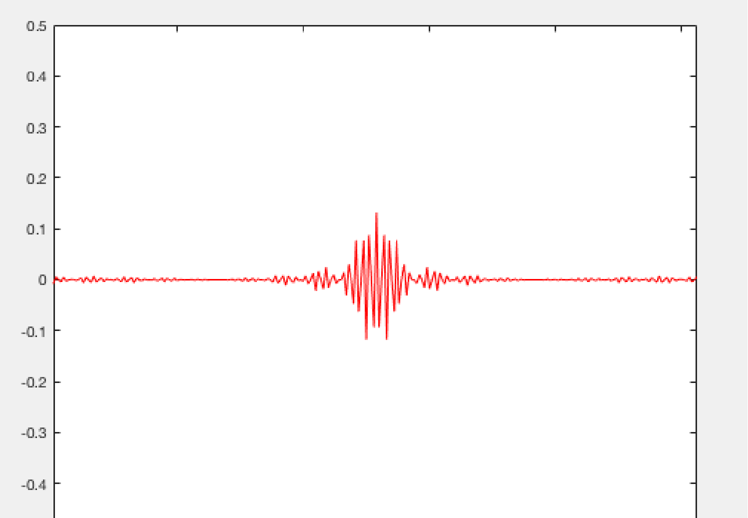
One might thing of a bandpass filter in frequency like this…



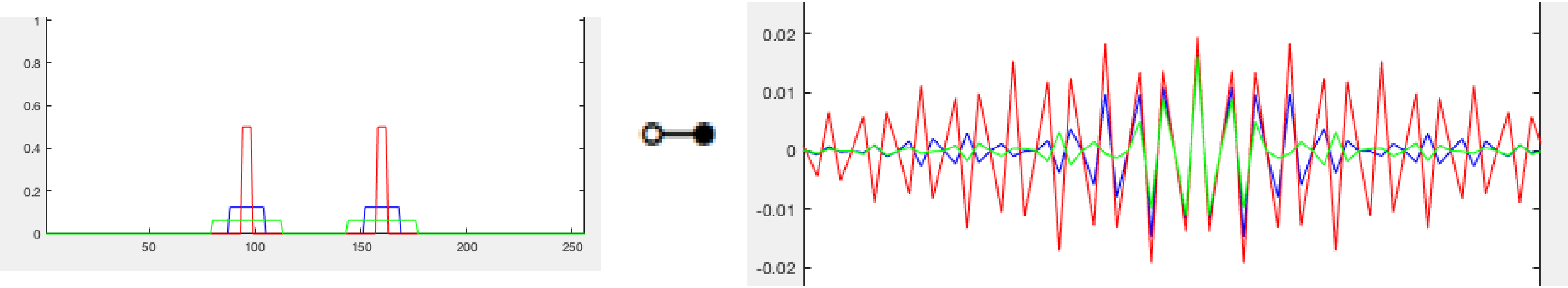
times



Would mean convolving with a huge mask:



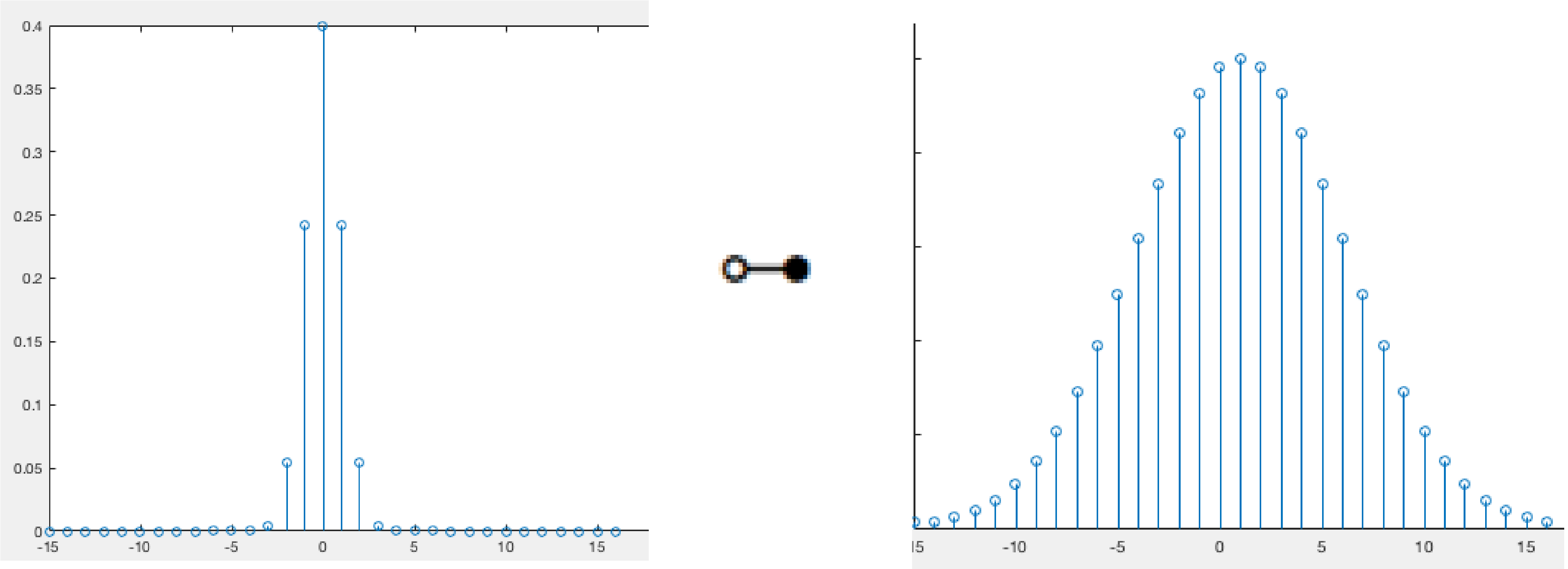
Frequency selectivity inversely proportional to spatial support (and hence location selectivity)



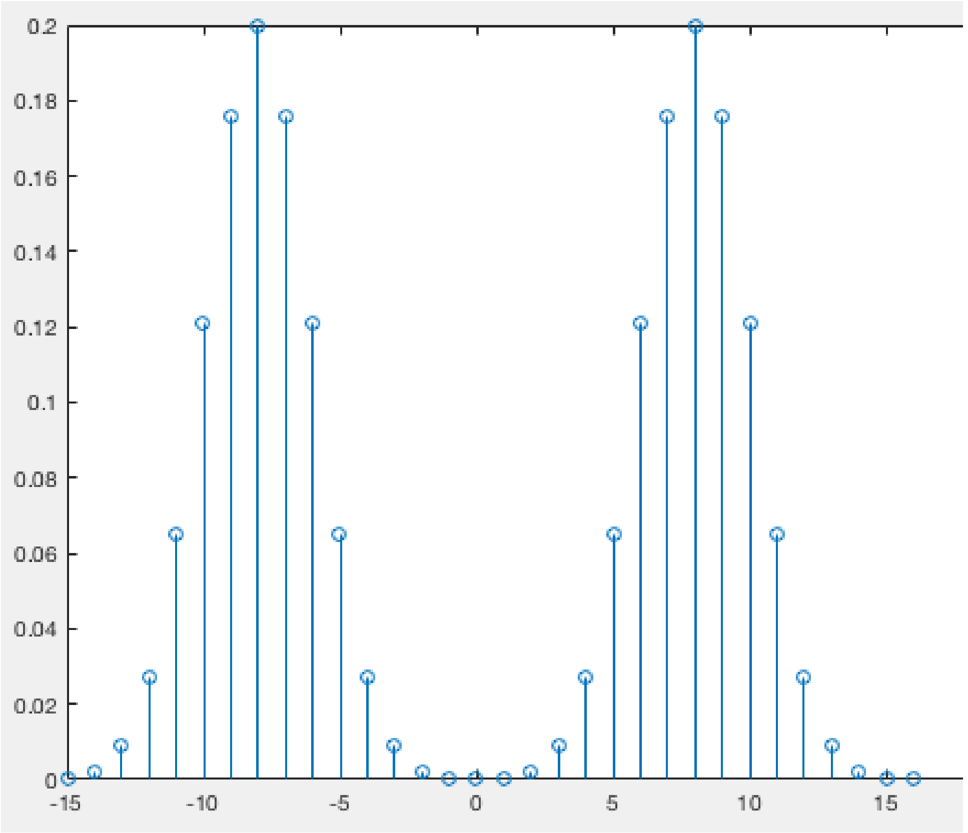
The uncertainty principle of signal processing

Best way to alleviate uncertainty principle….

Replace the rectangle with a Gaussian!!



How to make a bandpass out of a smoother (lowpass) ?



How to make a bandpass out of a smoothing filter (lowpass) ?

By creating two copies!

How?

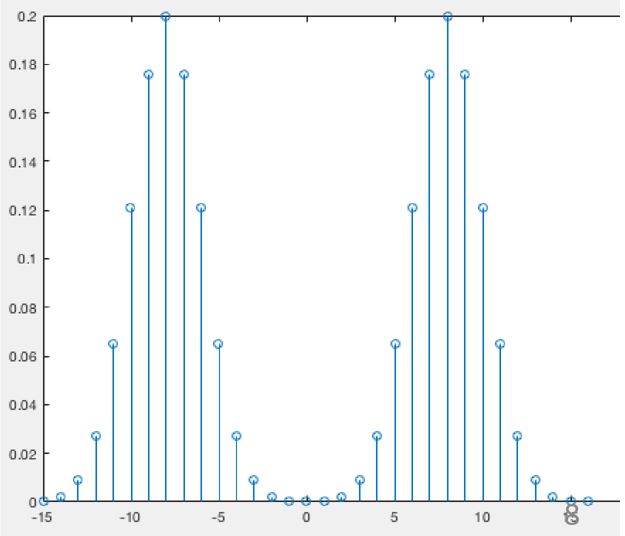
Modulating with a cosine!

Modulation with a cosine means

$$g(t) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-t^2/2\sigma^2} \circ\!\!-\!\!\bullet\ G(\omega) = e^{-\sigma^2\omega^2/2}$$

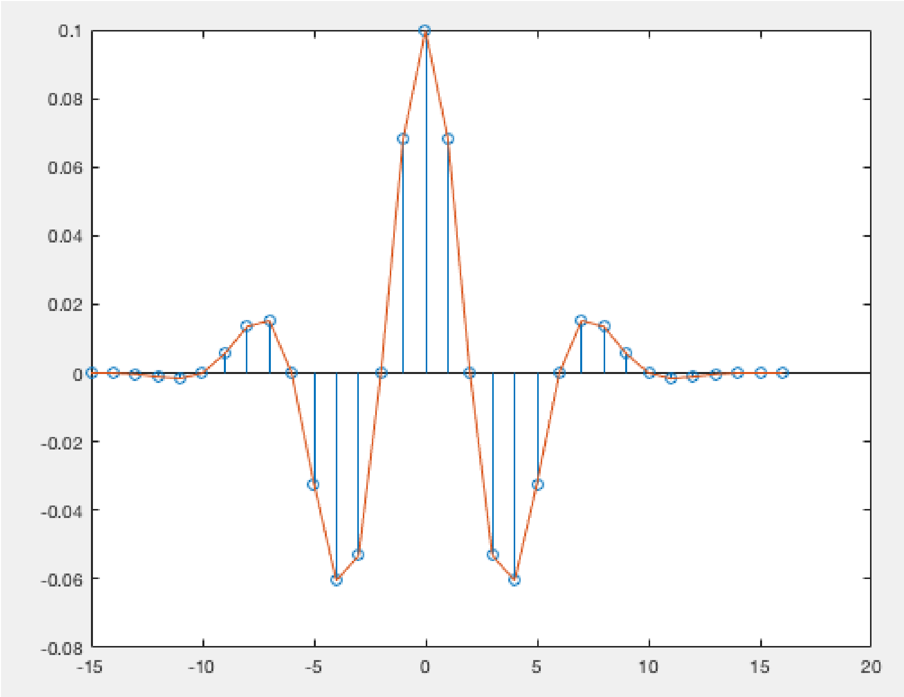
$$f(t) = \cos(\omega_0 t) \circ\!\!-\!\!\bullet\ \frac{1}{2}(\delta(\omega-\omega_0)+\delta(\omega+\omega_0))$$

$$F(\omega)G(\omega) = e^{-\sigma^2\omega_0^2/2}(\frac{1}{2}(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)))$$



Modulated Gaussian

$$\frac{1}{\sigma\sqrt{(2\pi)}} e^{-t^2/2\sigma^2} \cos \omega_0 t$$



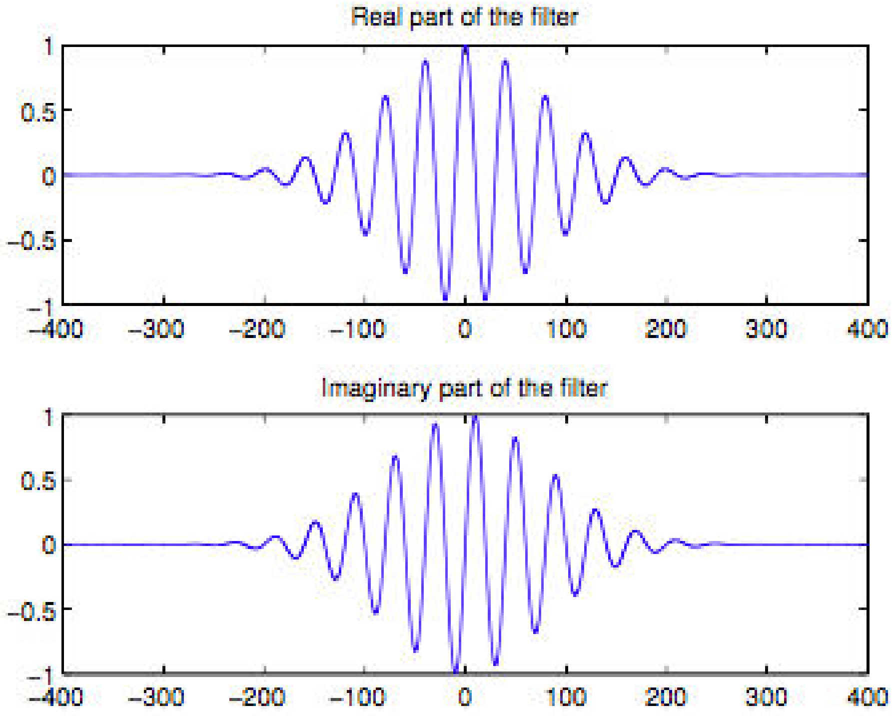
What happened with the phase?

$$\sin(\omega_0 t) \star \quad \frac{1}{\sigma\sqrt{(2\pi)}} e^{-t^2/2\sigma^2} \cos \omega_0 t \qquad =? \qquad \sin(\omega_0 t)$$

We need a phase-independent result!

Quadrature: complex filter with Re^2+Im^2=1

$$e^{-t^2/(2\sigma^2)}(\cos(\omega_0 t) + j\sin(\omega_0 t))$$



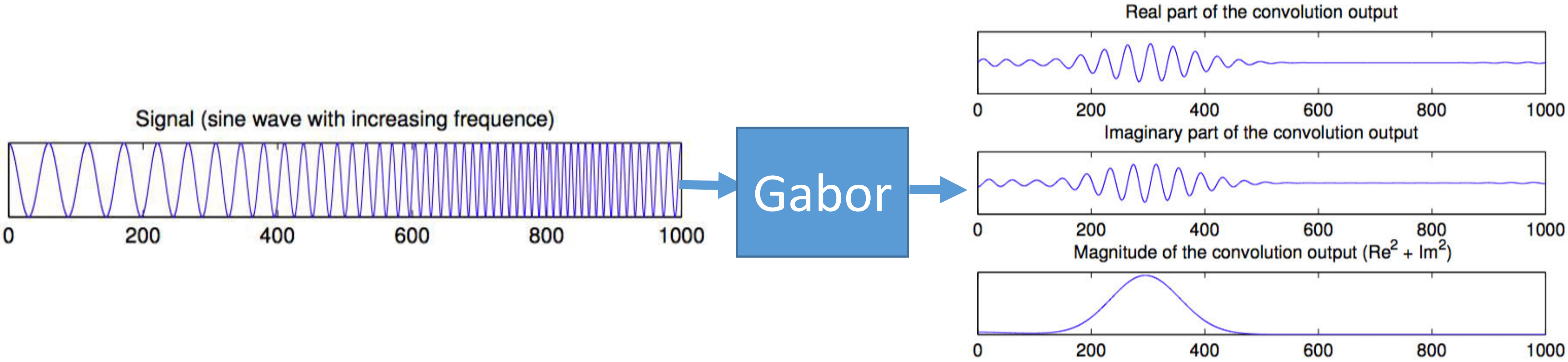
The Gabor Function

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/(2\sigma^2)} e^{j\omega_0 t}$$

$$\circ\!\!-\!\!\bullet$$

$$e^{-\sigma^2(\omega-\omega_o)^2/2}$$

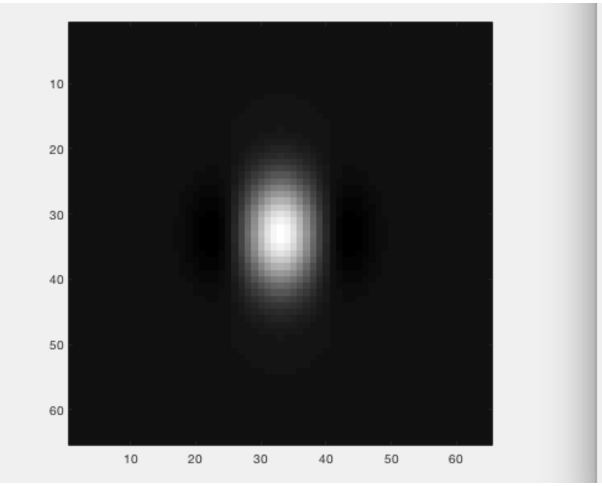
Frequency selectivity

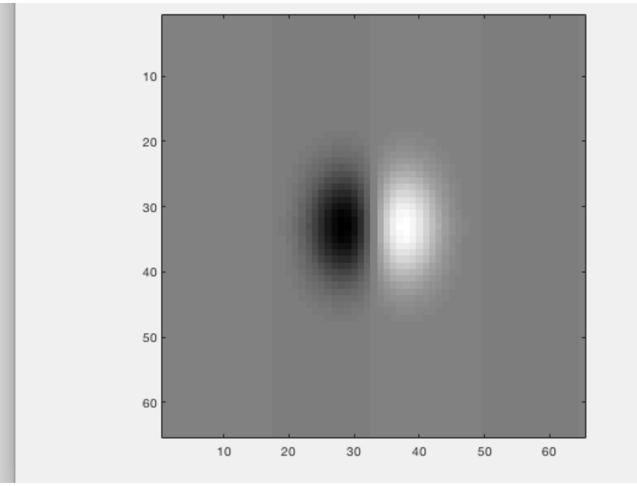


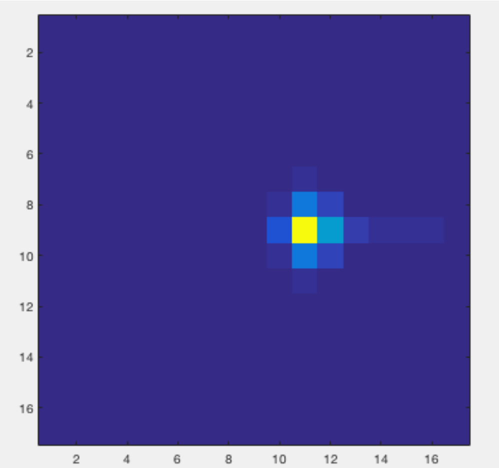
2D Gabor Function: selectivity in frequency and orientation

$$h(x, y, \sigma_1, \sigma_2, \omega_1) = \frac{1}{\sigma_1 \sigma_2 2\pi} e^{-(x^2/2\sigma_1^2 + y^2/2\sigma_2^2)} e^{j\omega_1 x}$$

$$e^{-(x^2/2\sigma_1^2 + y^2/2\sigma_2^2)} e^{j\omega_1 x} \circ\!\!-\!\!\bullet\; e^{-(\sigma_1^2(\omega_1 - \omega_x)^2 + \sigma_2^2 \omega_y^2)/2}$$

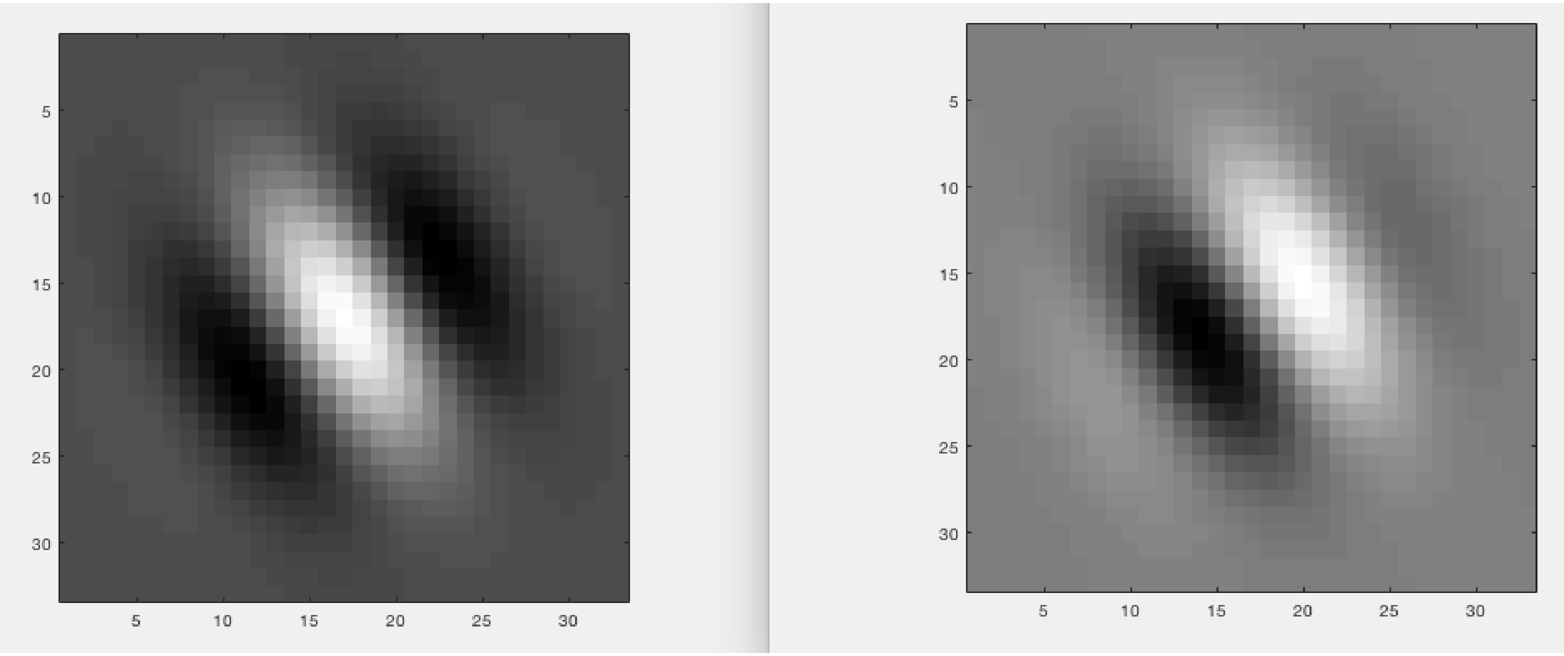


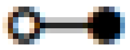


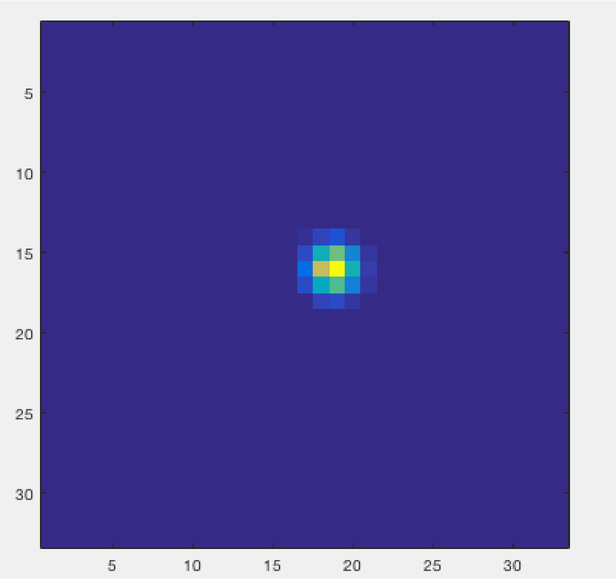


$$e^{-(x^2/2\sigma_1^2+y^2/2\sigma_2^2)}\, e^{j\omega_1 x} \;\circ\!\!-\!\!\bullet\; e^{-(\sigma_1^2(\omega_1-\omega_x)^2+\sigma_2^2\omega_y^2)/2}$$

Rotated 2D Gabor Filter







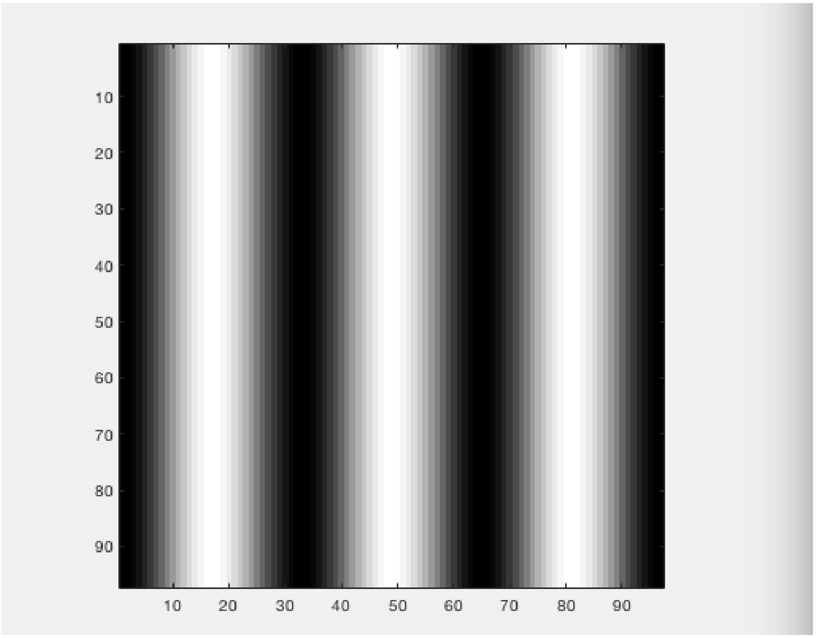
Video 4.2
Kostas Daniilidis

Property of Penn Engineering, Kostas Daniilidis

Scale selectivity

Build a system that detects edges and features independent from scale (size)

Let us look at an "unconventional edge":



$$\cos \omega_0 x \longrightarrow \boxed{g'_\sigma(x)} \longrightarrow$$

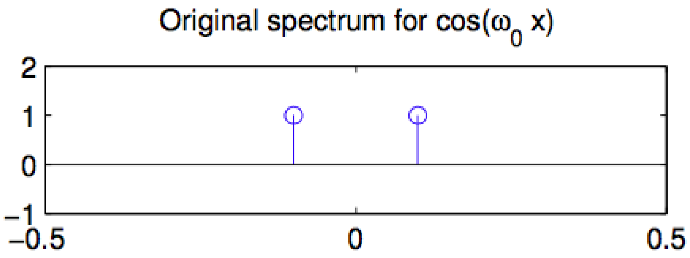
Build a system that detects edges and features independent from scale (size)

$$\cos \omega_0 x \longrightarrow \boxed{g'_\sigma(x)} \longrightarrow \text{“smoothed derivative”}$$

$$\tfrac{1}{2}(\delta(\omega-\omega_0)+\delta(\omega+\omega_0))\,j\omega e^{-\omega^2\sigma^2/2}$$

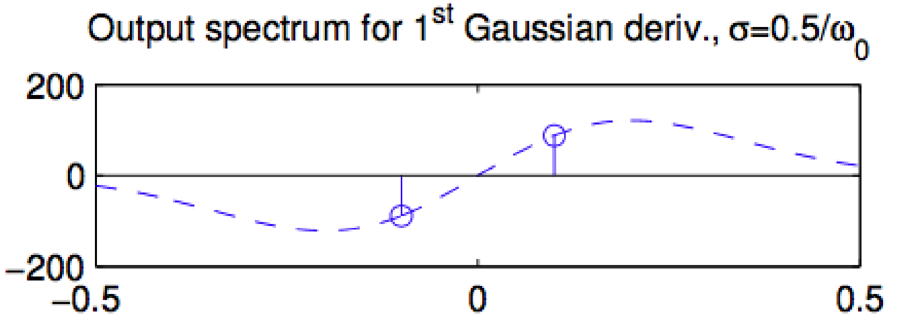
$$= \tfrac{1}{2}(\delta(\omega-\omega_0)-\delta(\omega+\omega_0))\,j\omega_0 e^{-\omega_0^2\sigma^2/2}$$

$$g'_\sigma = \frac{dg_\sigma}{dx} \text{ where } g_\sigma = \frac{1}{\sqrt{2\pi}\sigma}e^{-x^2/2\sigma^2}$$



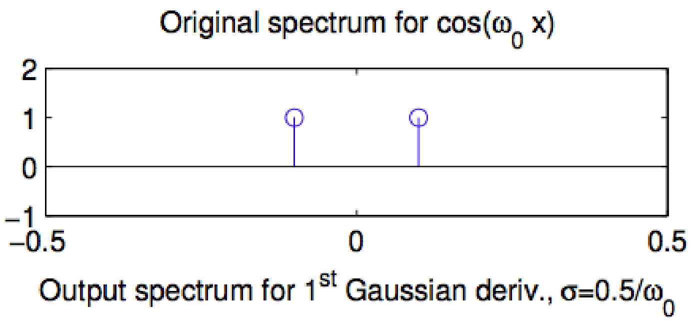
Output spectrum for 1st Gaussian deriv., σ=0.5/ω0

$$\longrightarrow \boxed{g'_\sigma(x)} \longrightarrow$$

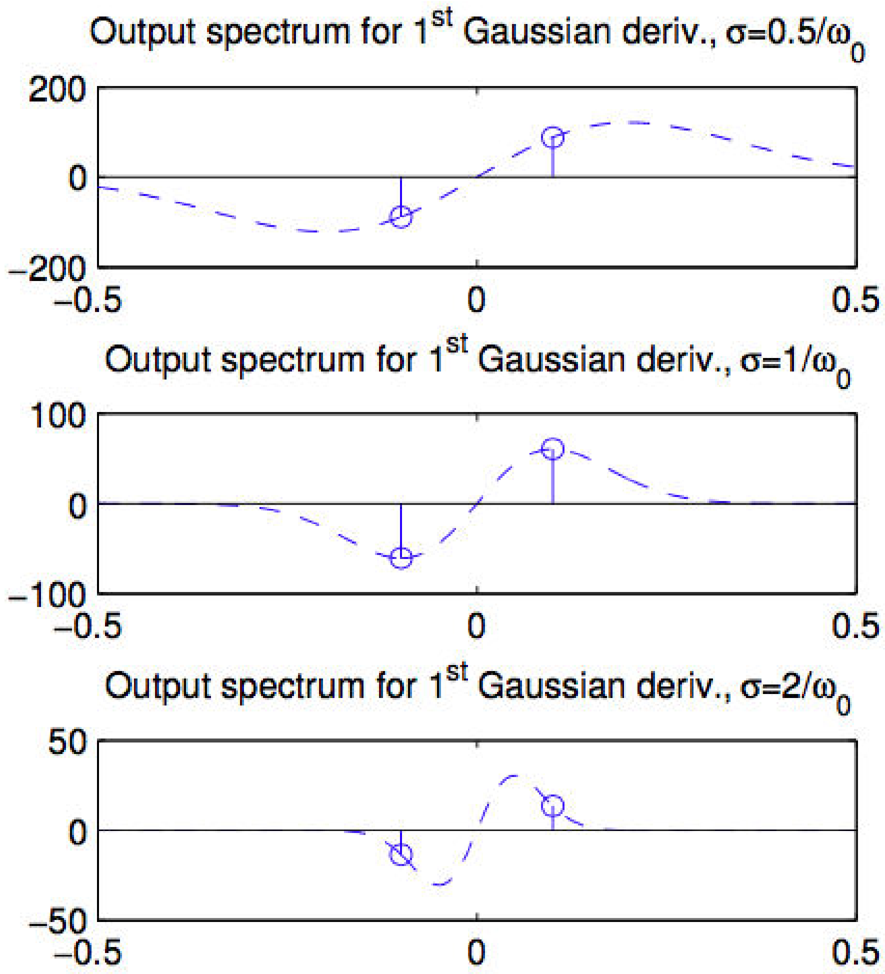


Property of Penn Engineering, Kostas Daniilidis

Response is maximum when filter scale σ matches incoming scale (ω_o).



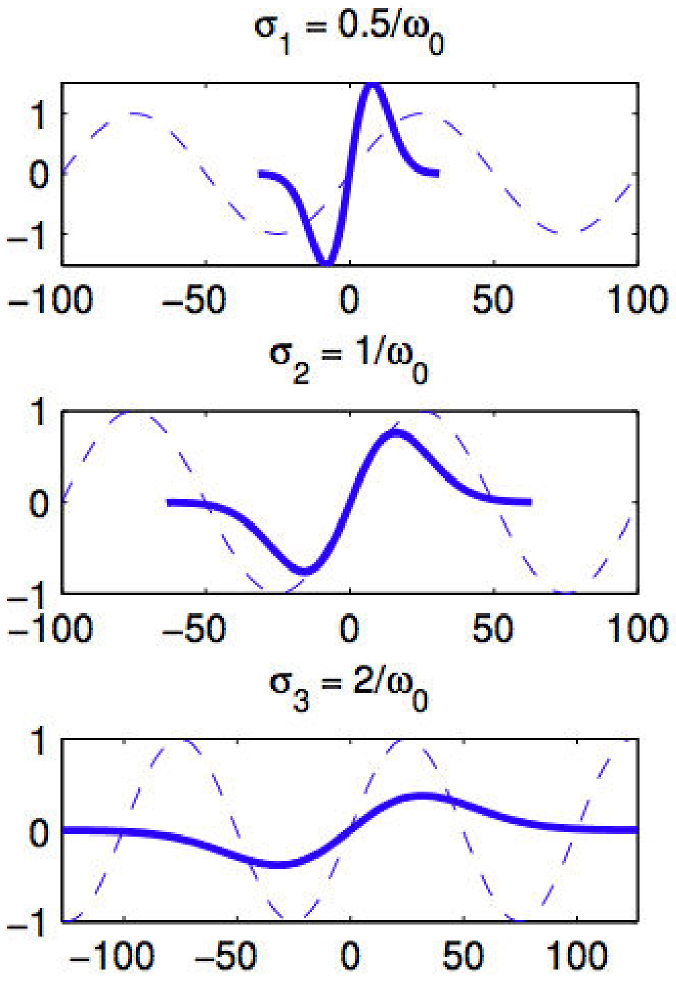
$$g'_\sigma(x)$$



$$j\omega_0 e^{-\omega_0^2\sigma^2/2}$$

Look at edge detection as template matching

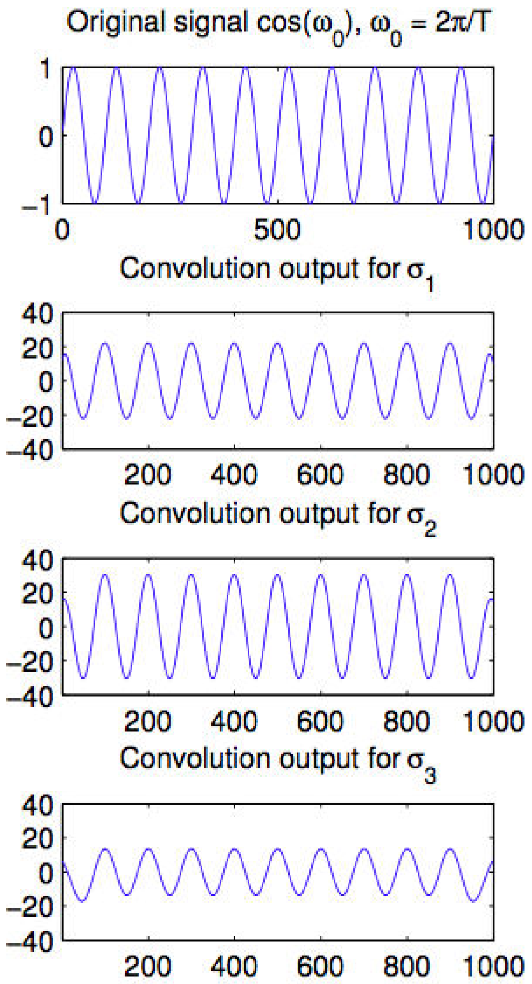
Which size of “edge template” matches the original edge?



Edge detection output for different σ's:

As σ increases the peaks at the edges weaken!!

$$\omega_0 e^{-\omega_0^2 \sigma^2 / 2}$$



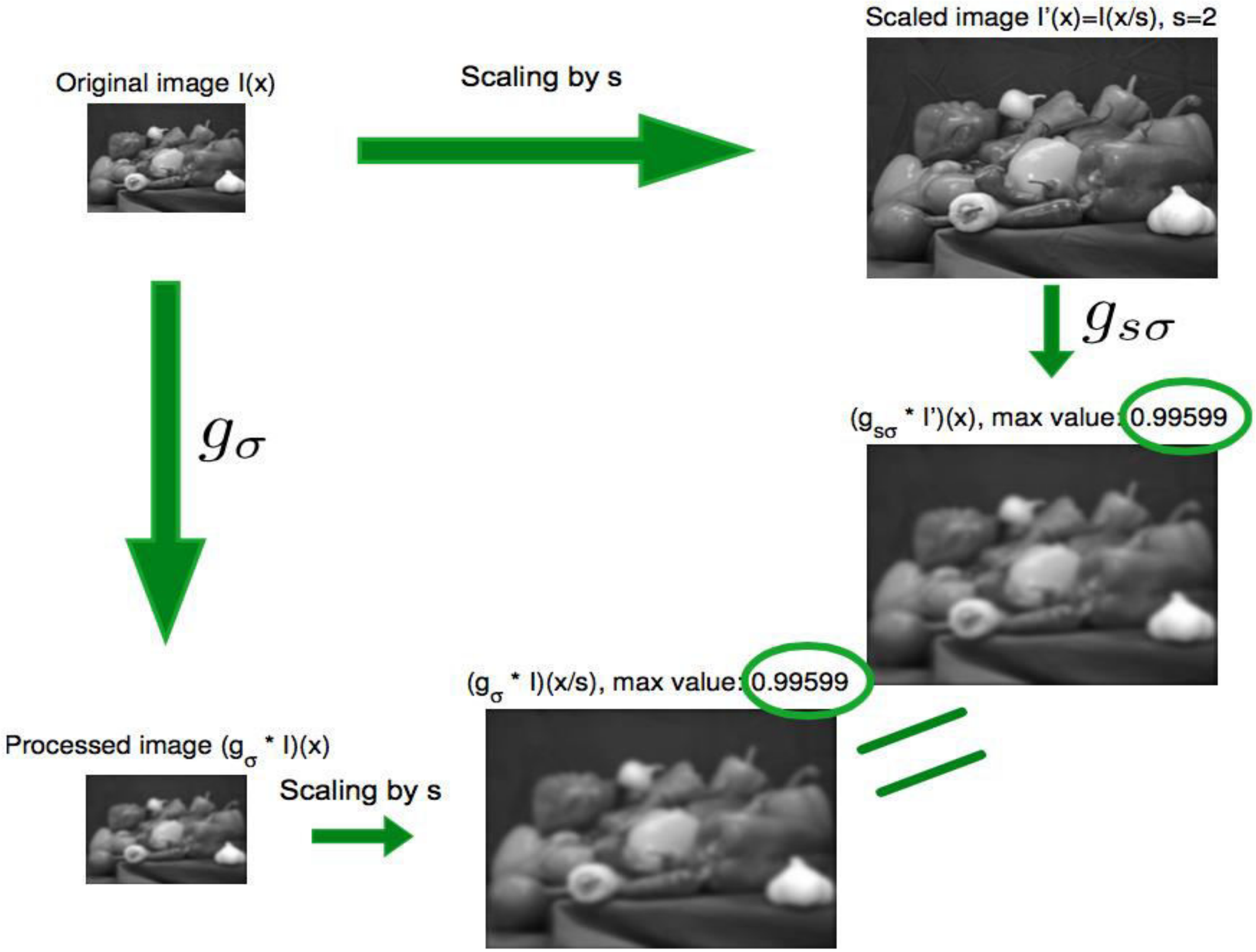
Why we need a scale normalization of the filter!

Assume a scaled version of the original signal (twice as wide for s=2), then calculate the convolution with Gaussian:

$$I'(x) = I\left(\frac{x}{s}\right)$$

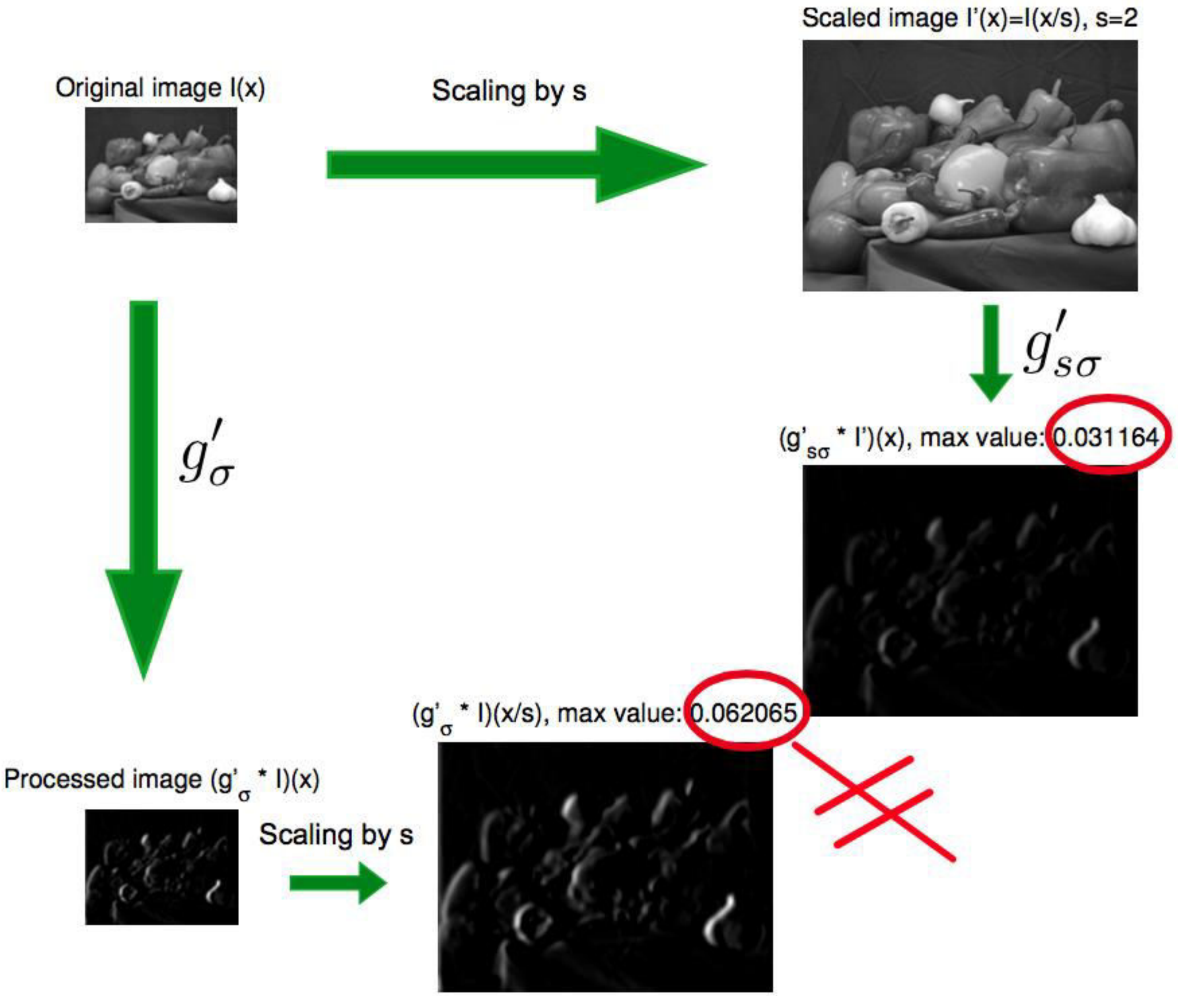
$$\boxed{(I' \star g_{s\sigma})(x) = (I \star g_{\sigma})\left(\frac{x}{s}\right)}$$

Scaling by s and convolving with scaled Gaussian is the same as applying a Gaussian and then scaling by s!!!

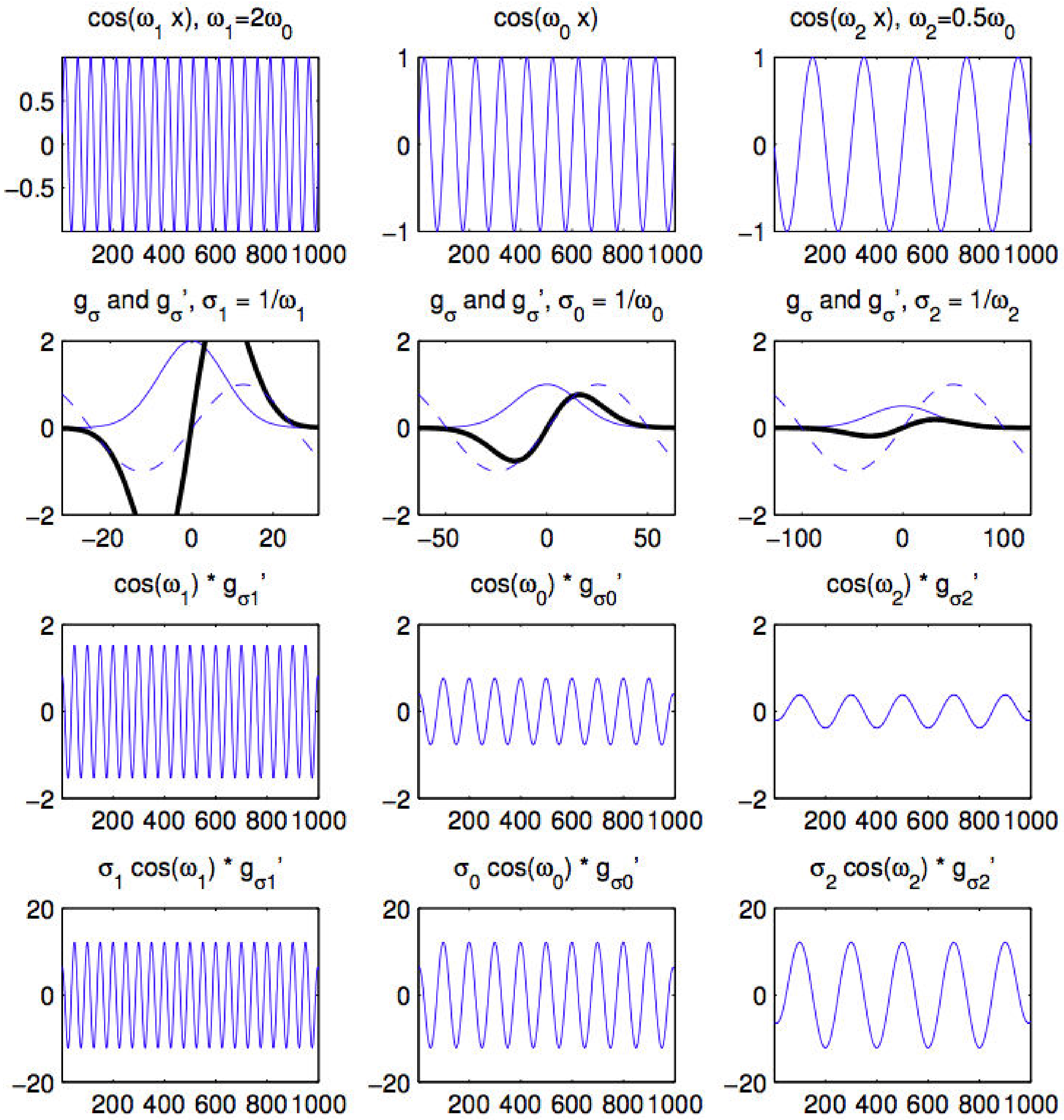

Original image I(x)
Scaling by s
Scaled image I'(x)=I(x/s), s=2
$g_{s\sigma}$
g_σ
$(g_{s\sigma}$ * I')(x), max value: 0.99599
$(g_\sigma$ * I)(x/s), max value: 0.99599
Processed image $(g_\sigma$ * I)(x)
Scaling by s
=

But this is not true for the 1st derivative!

$$(I' \star g'_{s\sigma})(x) = \frac{1}{s}(I \star g'_{\sigma})\left(\frac{x}{s}\right)$$



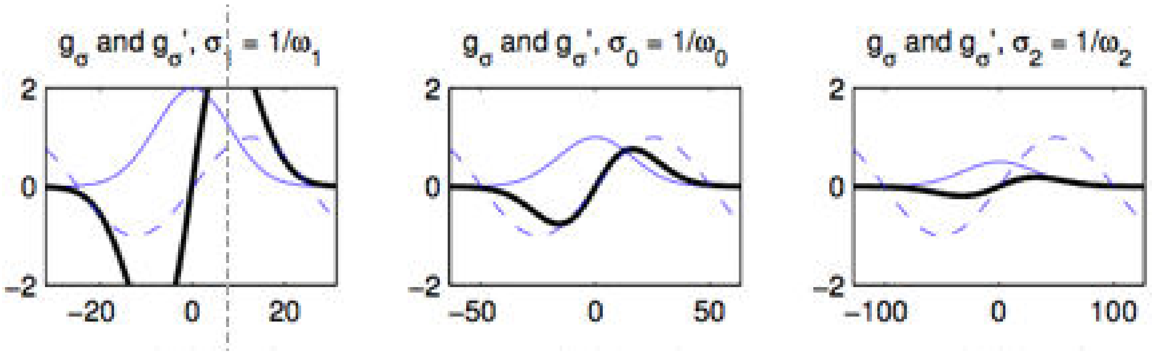
No matter what the scale of the feature, edge detector (1^{st} Gaussian derivative) response should be the same when the filter matches the feature scale.

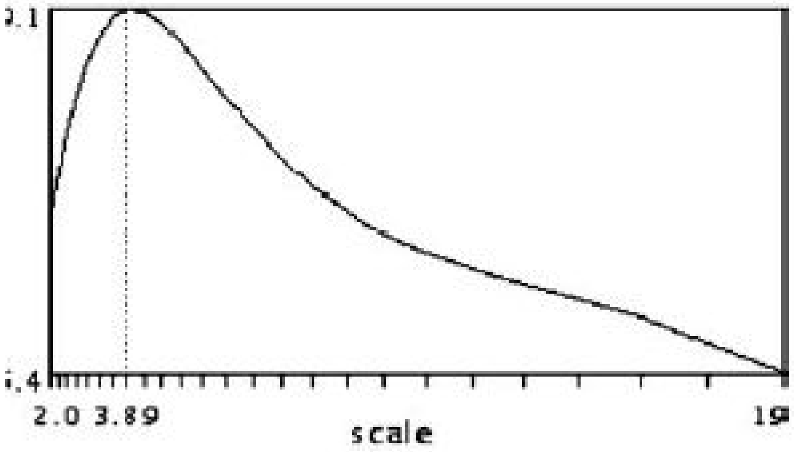


Pr

Can we find (select) the intrinsic image scale?

Yes, by taking the maximum over scale!





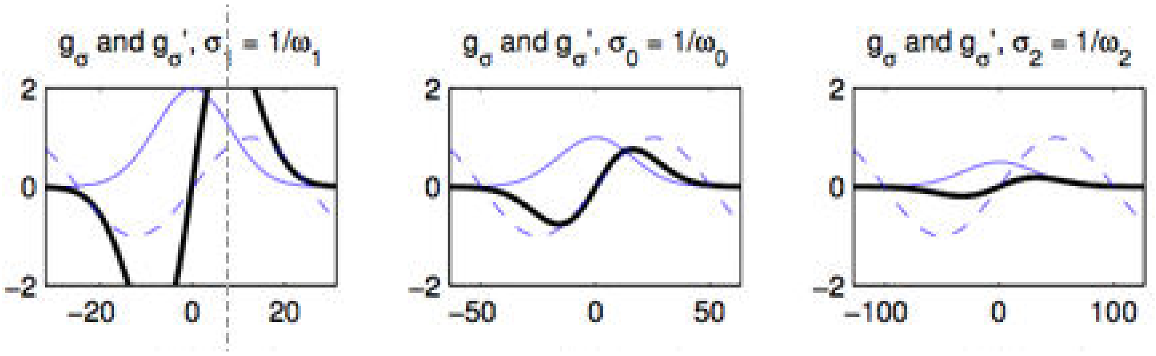
Video 4.3
Kostas Daniilidis

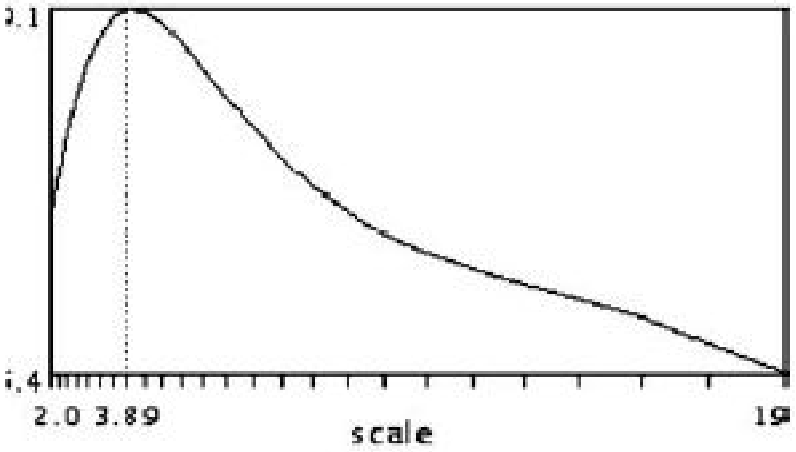
Property of Penn Engineering, Kostas Daniilidis

Scale Invariant Feature Transform (SIFT)

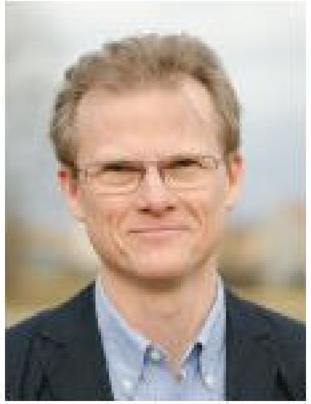
Can we find (select) the intrinsic image scale?

Yes, by taking the maximum over scale!





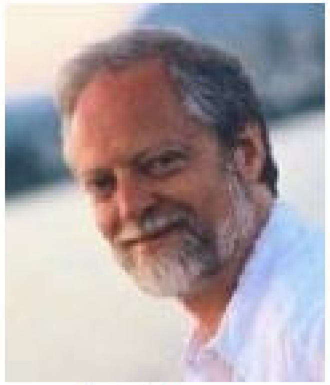
Scale selection and invariance



Tony Lindeberg

Feature detection with automatic scale selection

Authors	Tony Lindeberg
Publication date	1998/11/1
Journal	International Journal of Computer Vision
Volume	30
Issue	2

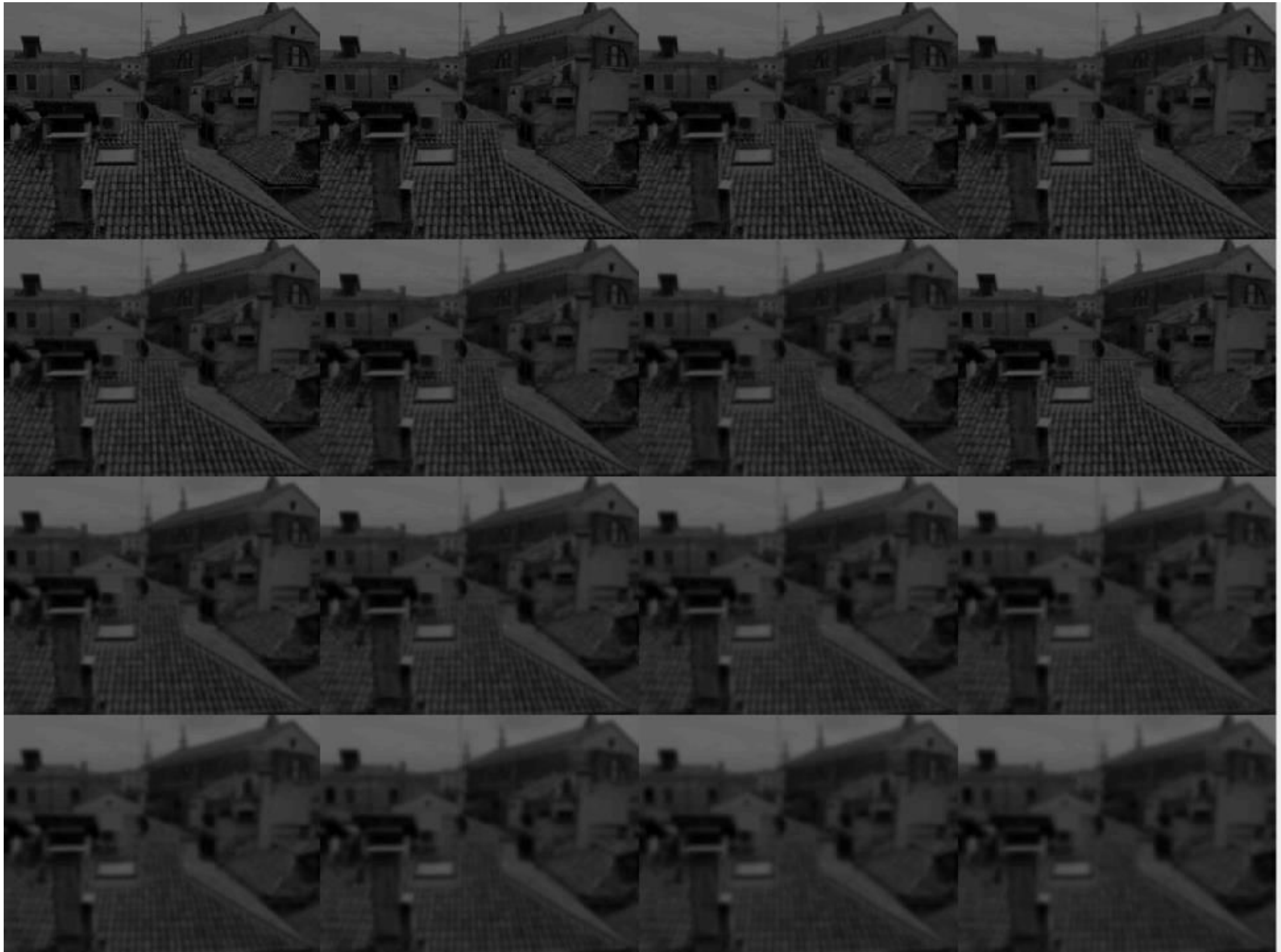


David Lowe

Distinctive image features from scale-invariant keypoints

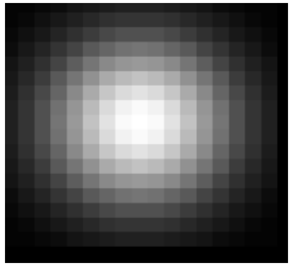
Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2

The notion of scale space



How is scale space built?

*

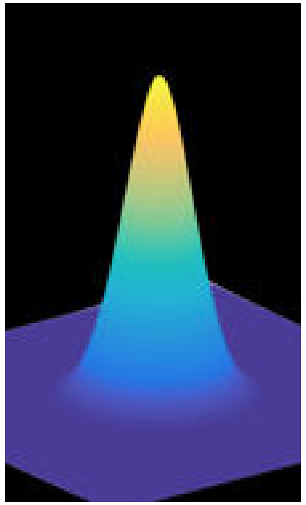


2D Gaussian

=

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

* means convolution



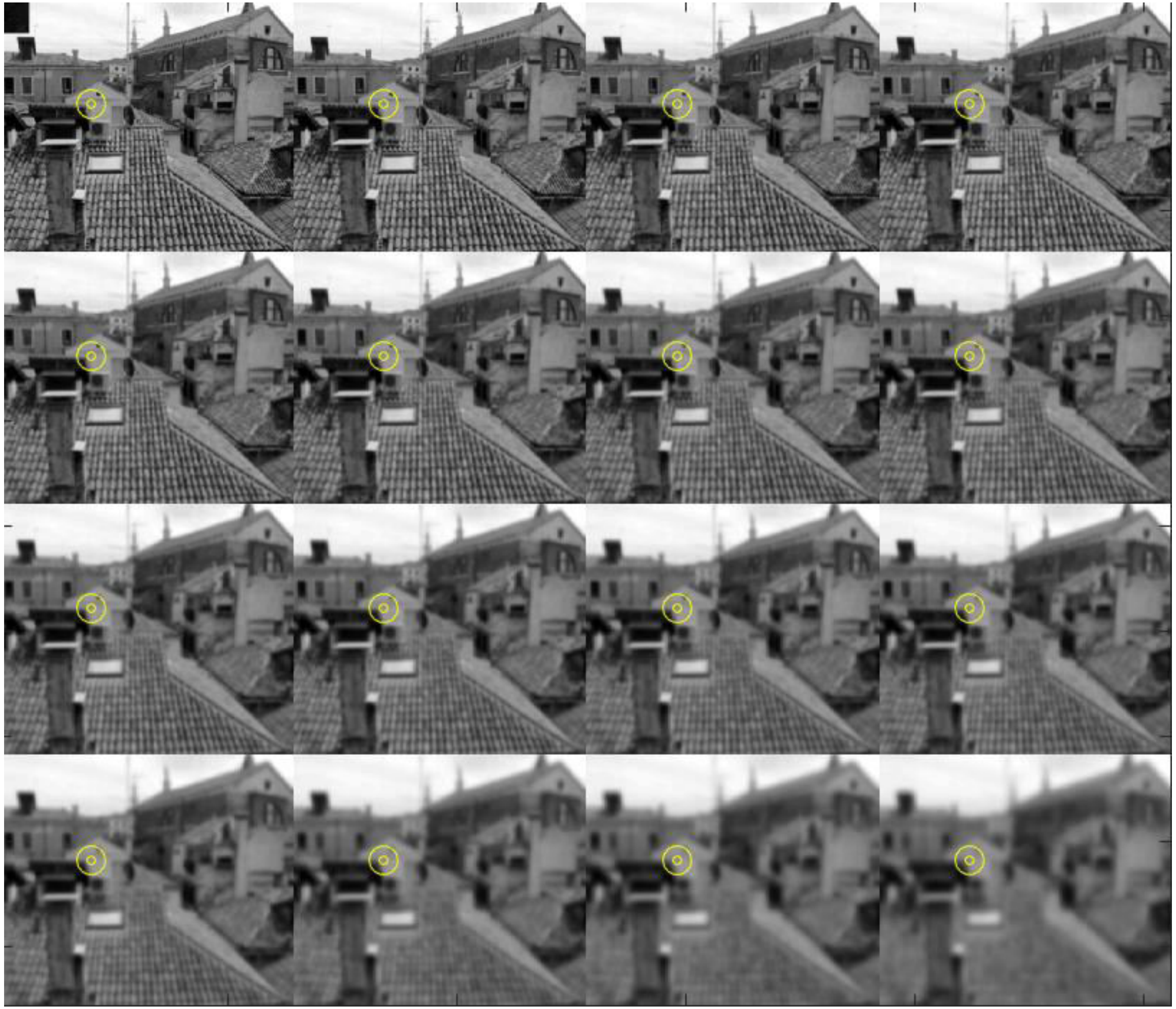
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

F

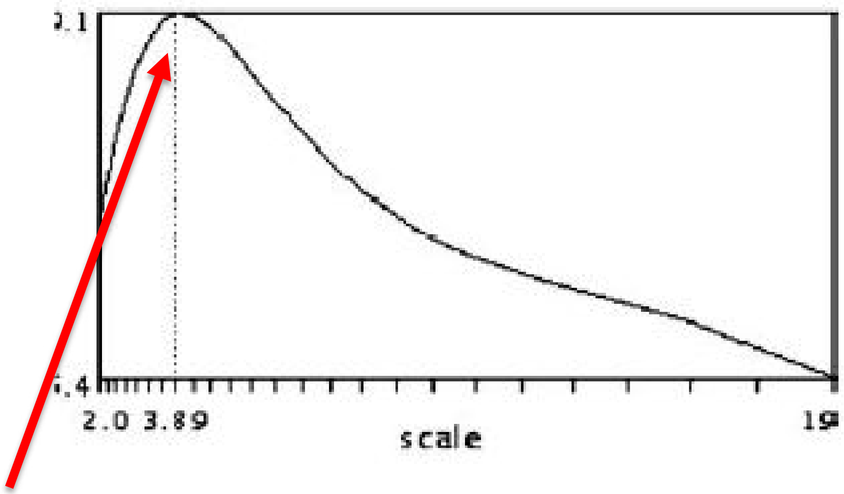
The same scale but subsampled

We subsampled it every time that the sigma of the Gaussian was doubled!

Now look at the same pixel across scale

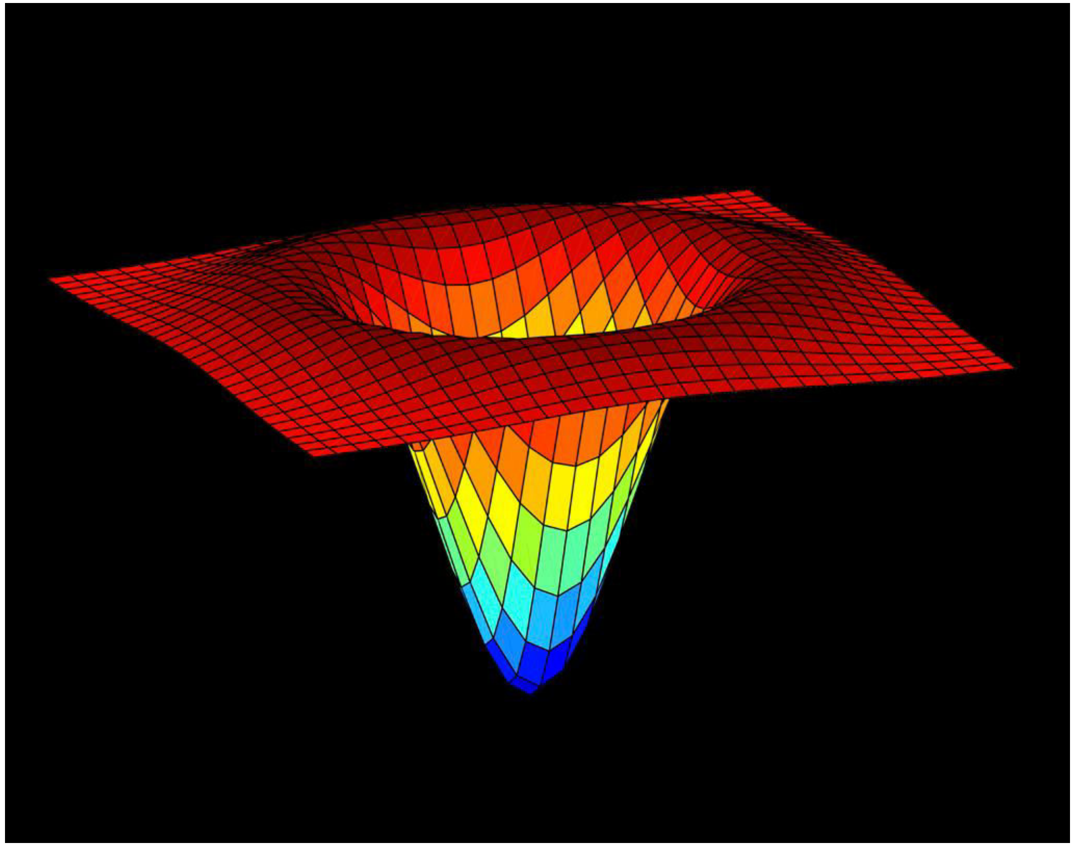


Scale selection



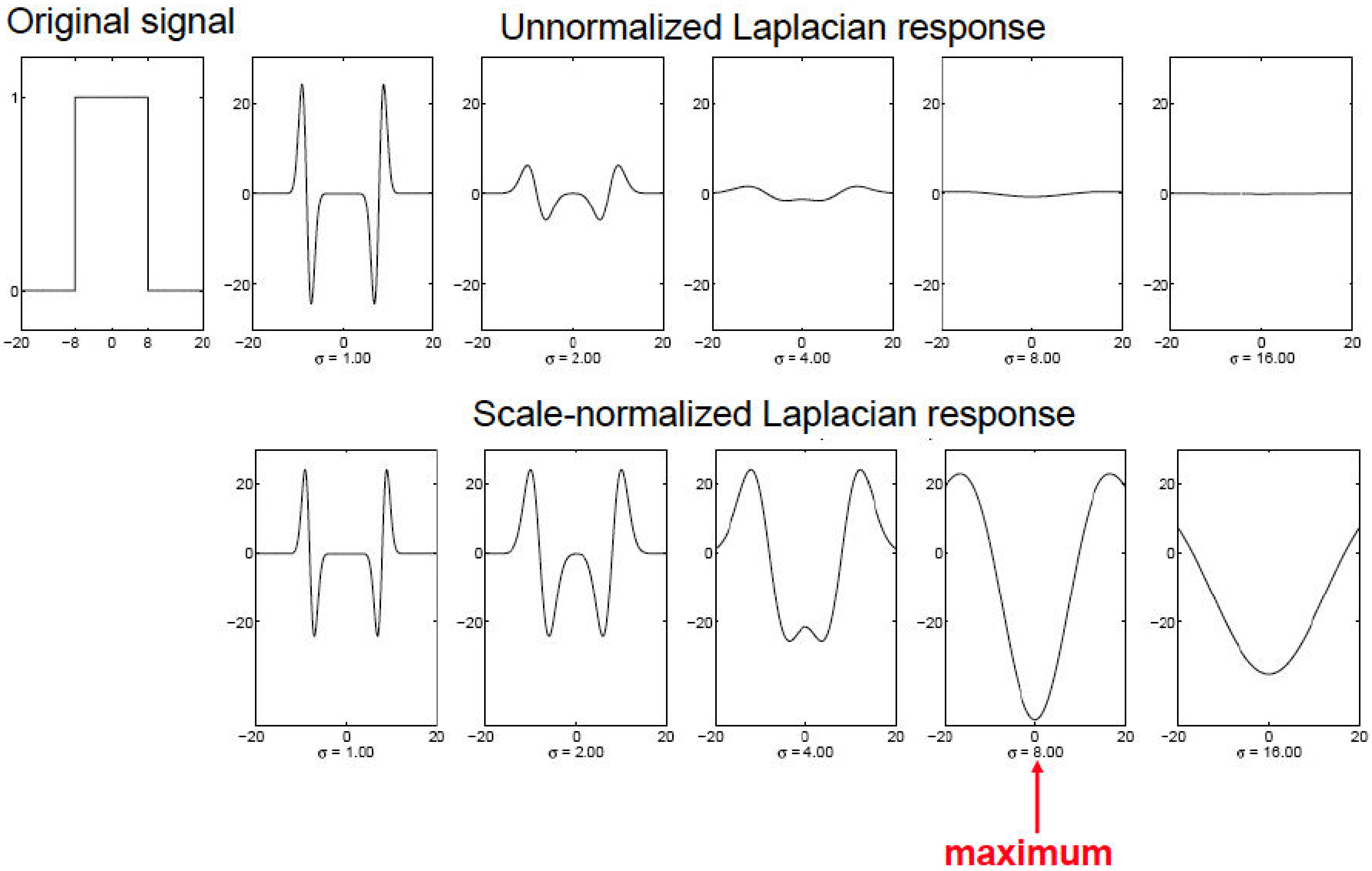
- The maximum across scale is the intrinsic scale of the image structure
- if the smoothed value is scale normalized.
- It turns out that only the derivatives of the Gaussian responses can be normalized.

We choose the 2nd derivative (trace of Hessian) , called Laplacian of Gaussian (LoG)



Which has the nice property that it can be approximated as the difference of two Gaussians and can detect blob like features!

Normalization of the 2nd Gaussian Derivative

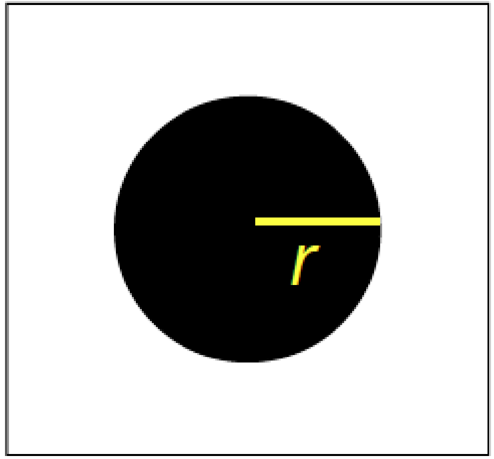


Scale selection

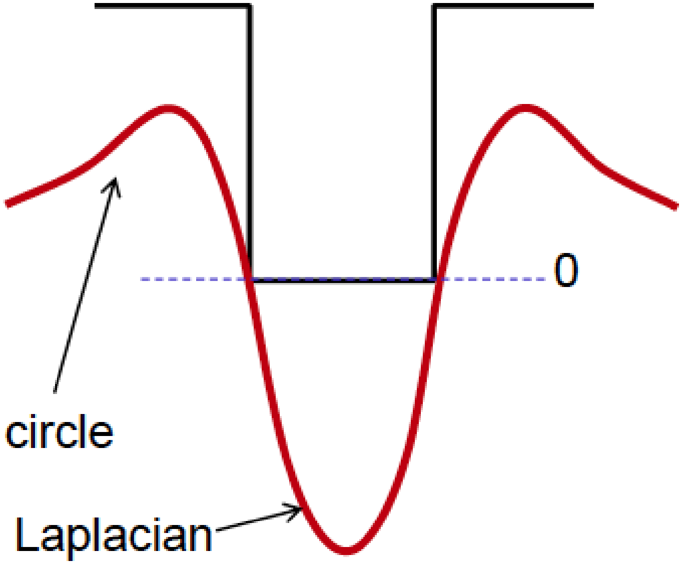
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$

- Therefore, the maximum response occurs at $\sigma = r / \sqrt{2}$.

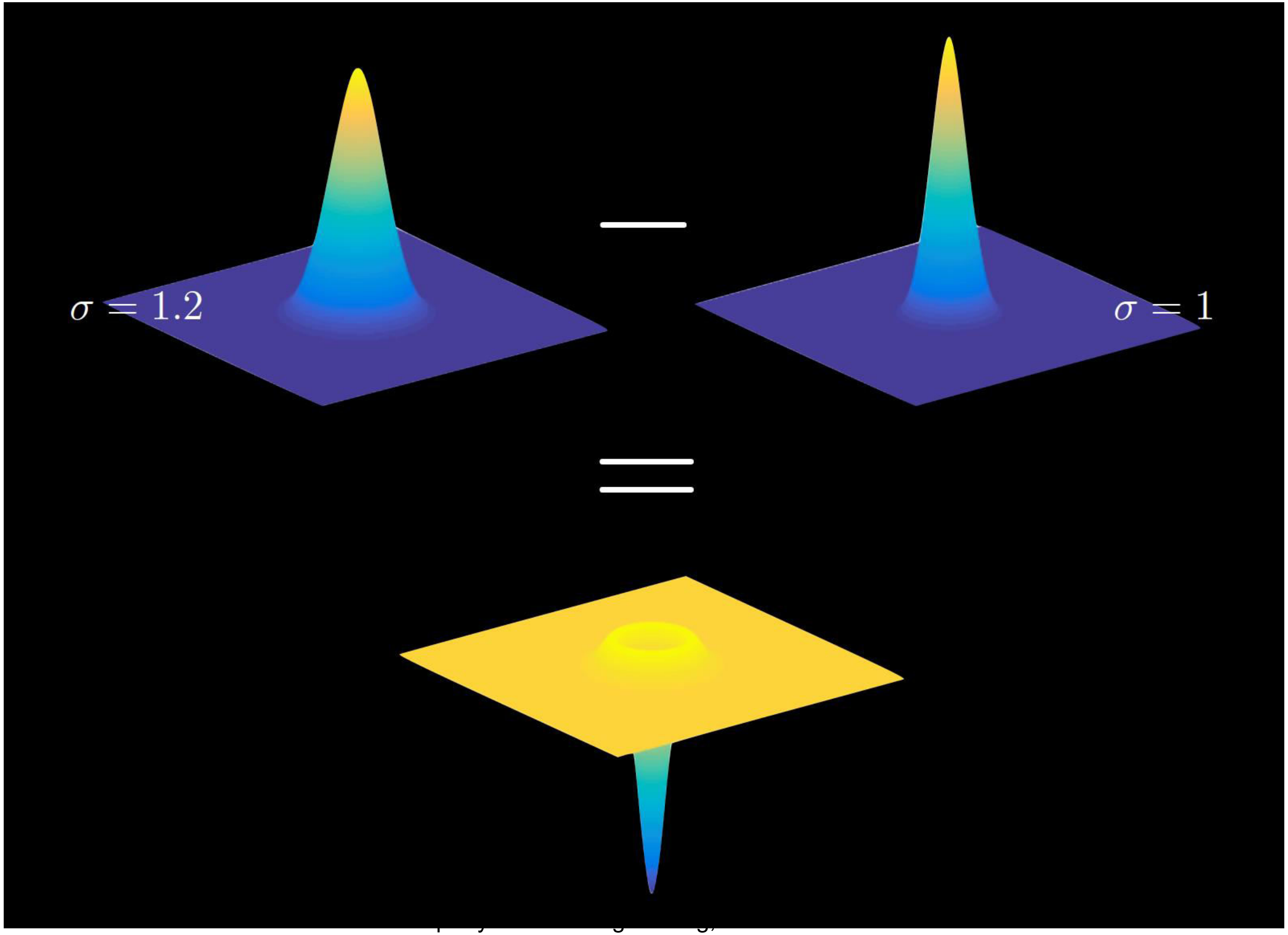


image

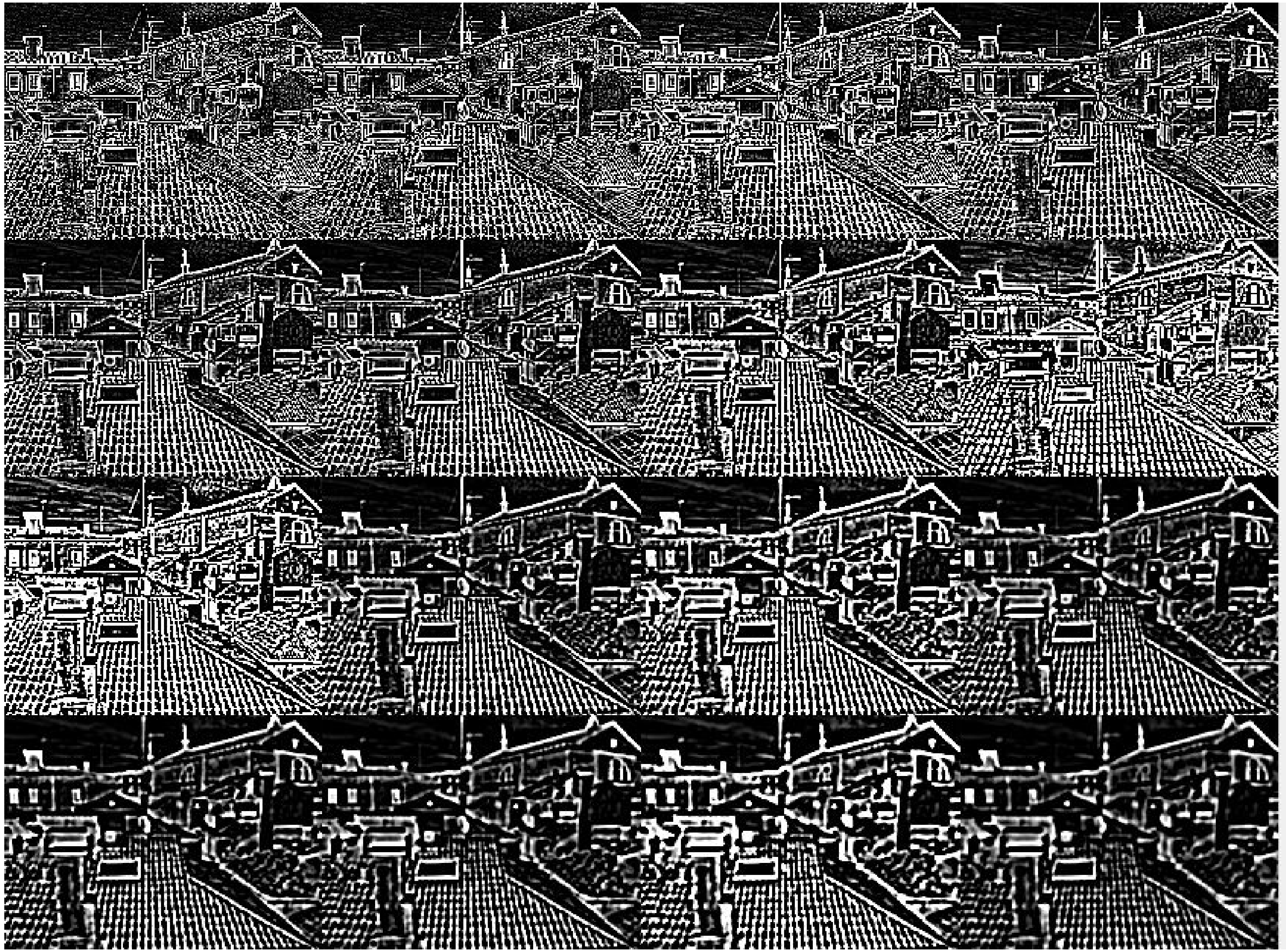


Slide from S. Lazebnik

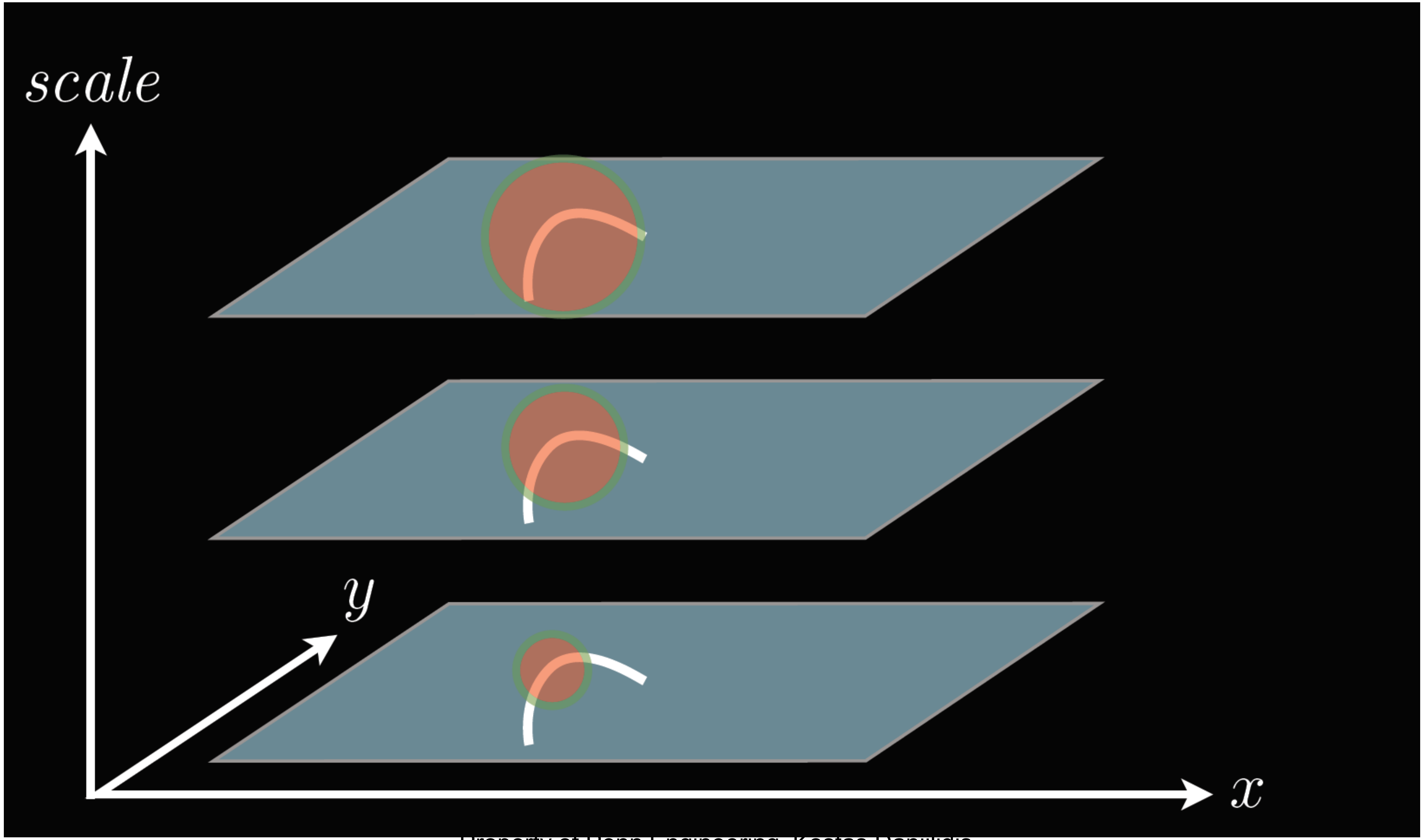
Difference of Gaussians (DoG)



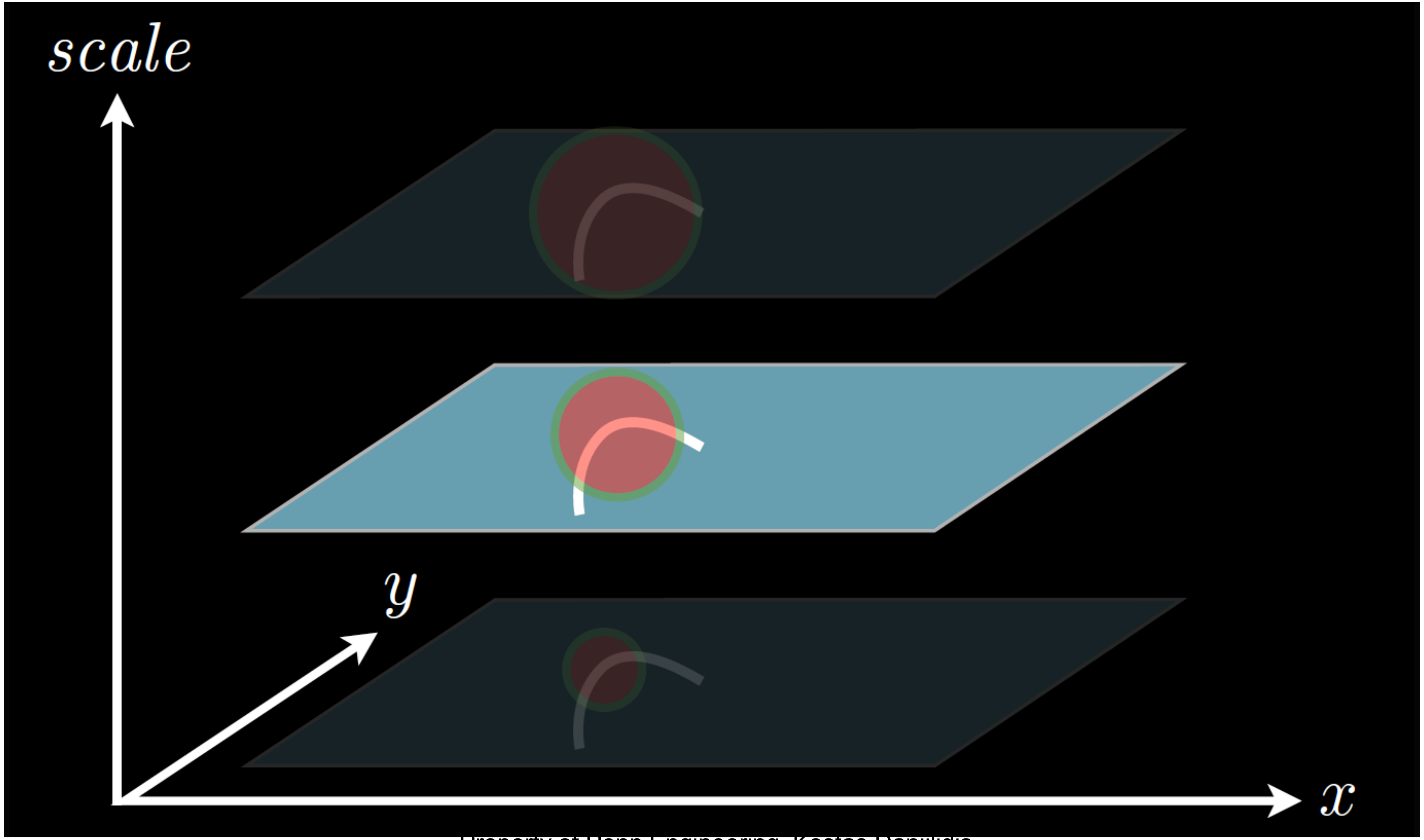
Laplacian Scale Space



We convolve DoG across space at different scales



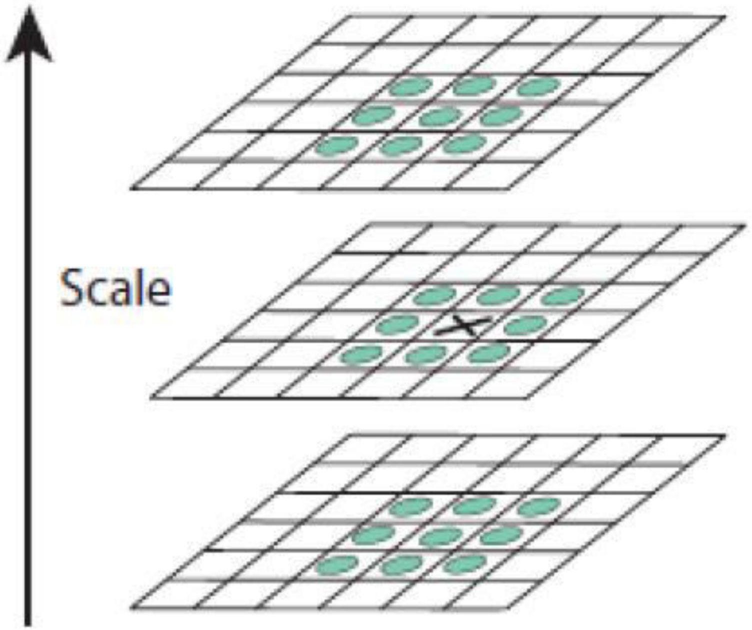
We convolve DoG across space at different scales and detect maximum



Property of Penn Engineering, Kostas Daniilidis

So, where is a SIFT keypoint?

- Definition of a keypoint: Maximum in the 3x3x3 (x,y,σ) region of the point.



Vedaldi’s vlfeat

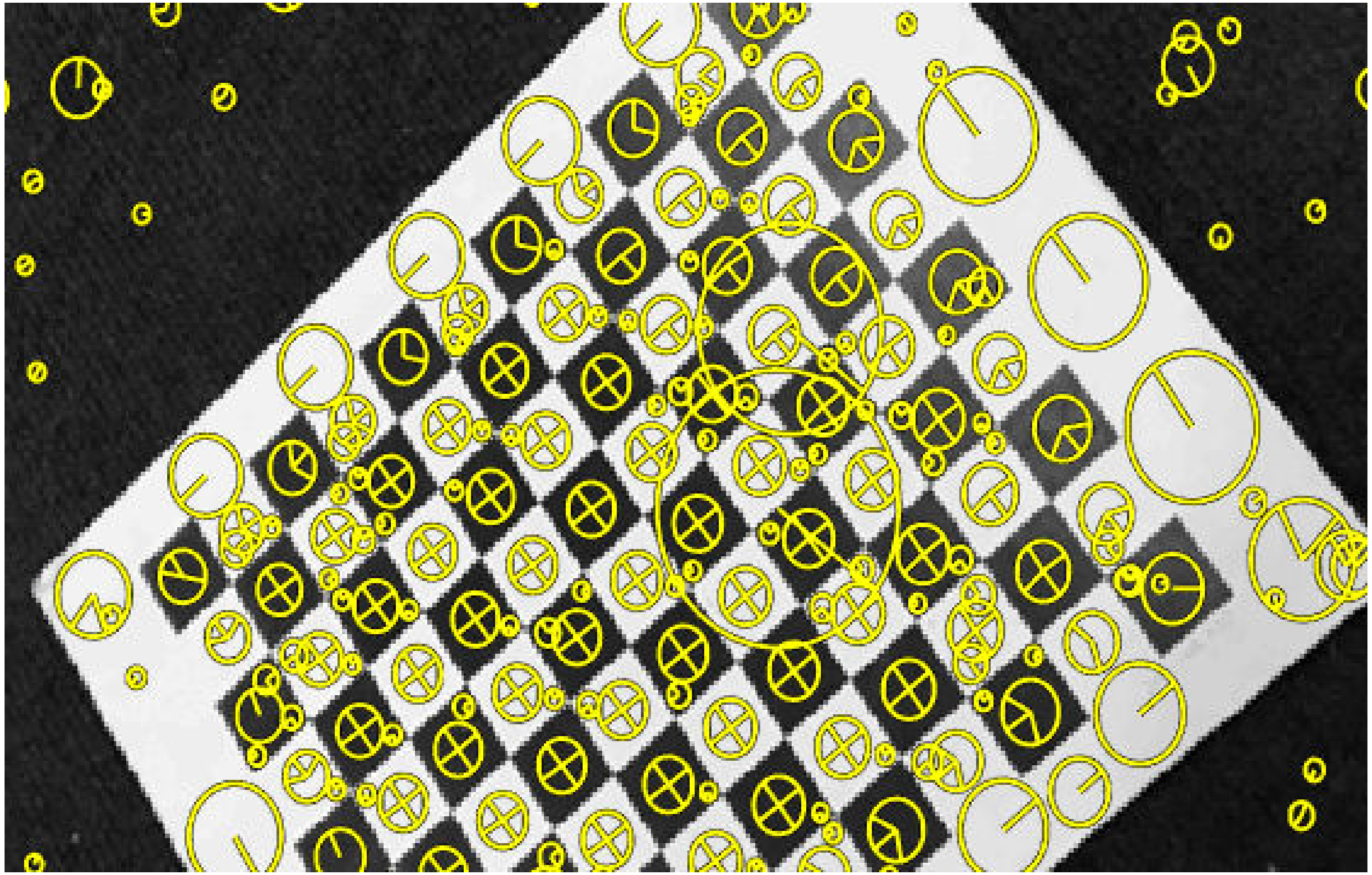
VLFeat.org

Home Download Tutorials Applications Documentation

Andrea Vedaldi, Ph.D. vedaldi@robots.ox.ac.uk
Associate Professor in Engineering Science
Information Engineering Building 30.05, Parks Road, Oxford, OX1 3PJ
Visual Geometry Group (directions)
Tel. +44 1865 273 127

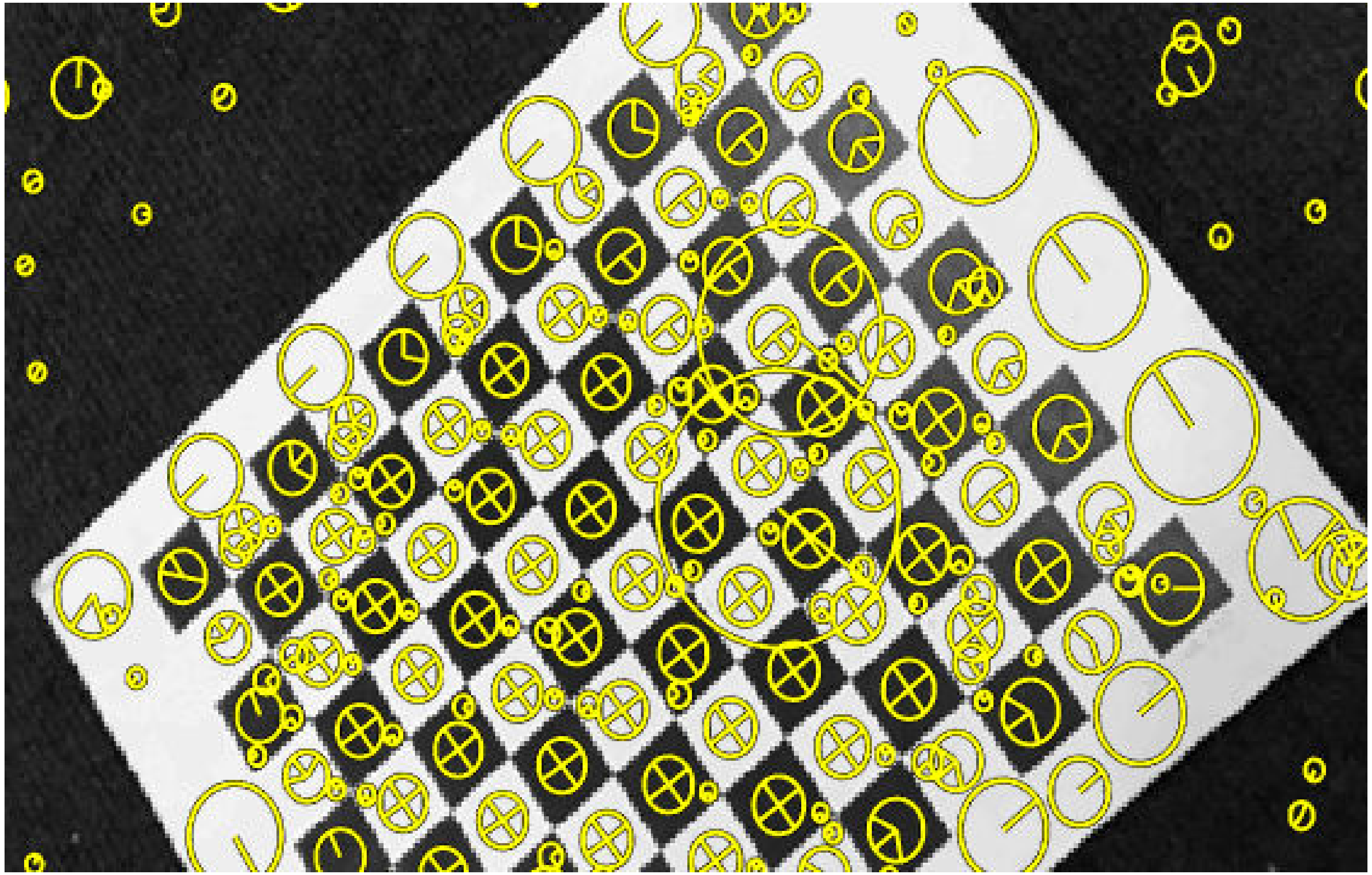
Résumé Google Scholar

Selected σ is visualized with a circle



Denoting the support region of the feature

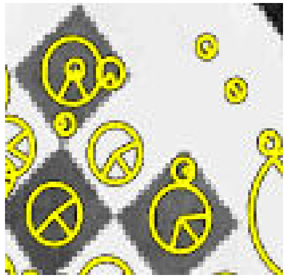
Detector is rotation invariant



Because Laplacian is isotropic and a maximum in (x,y,σ) is invariant to rotations.

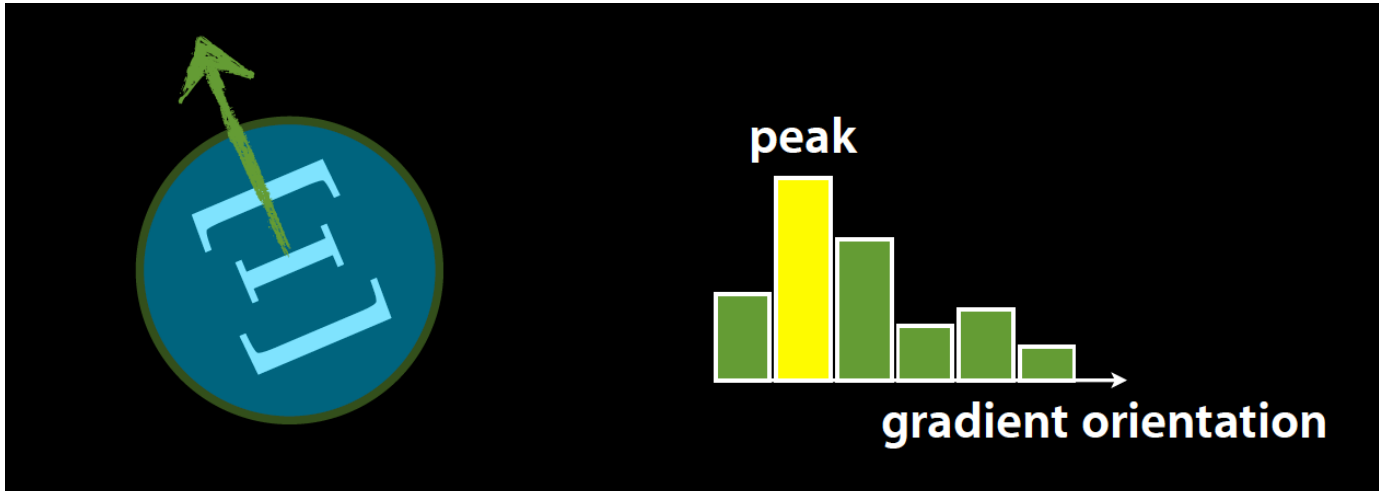
Descriptor invariance

- Since the intrinsic scale is detected (circle size) all circles will be normalized to a 16x16 region.



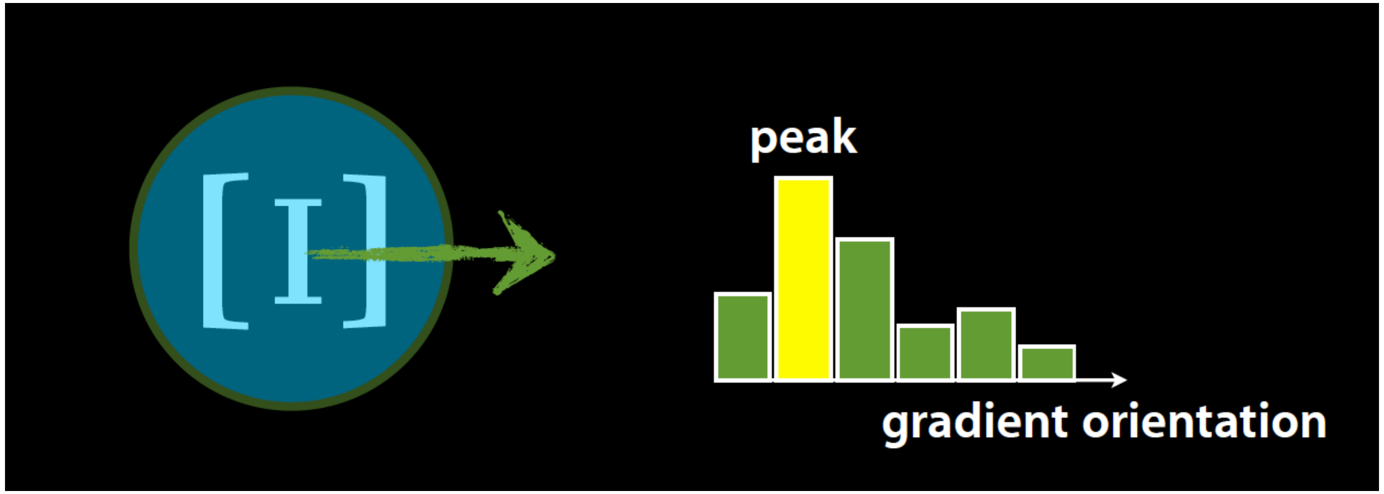
The descriptor should be also rotation invariant

- 1st Step: Find dominant orientation for the patch



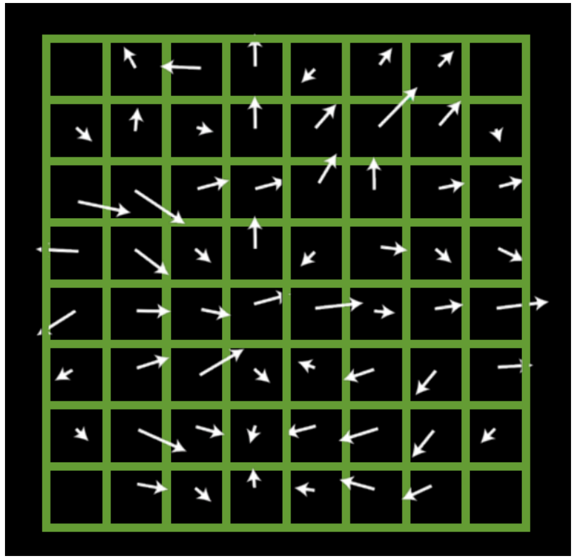
The descriptor should be rotation invariant

- 1st Step: Find dominant orientation for the patch
- 2nd Step: Rotate patch to point along x-axis



To extract a feature descriptor from a cell

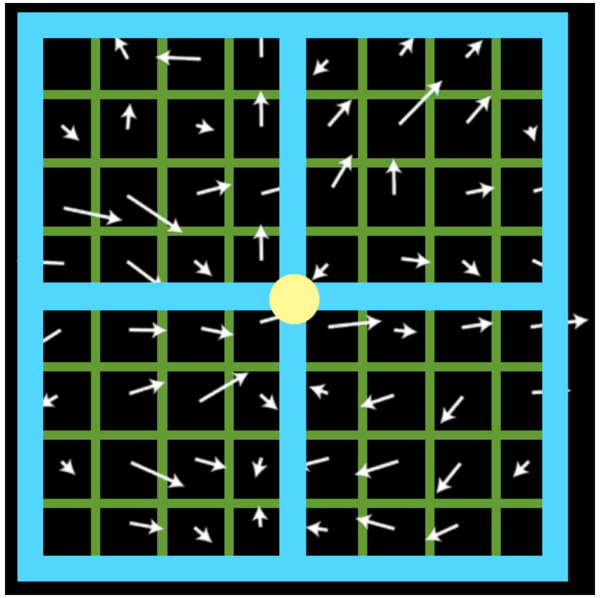
- Compute Image Gradients



To extract a feature descriptor from a cell

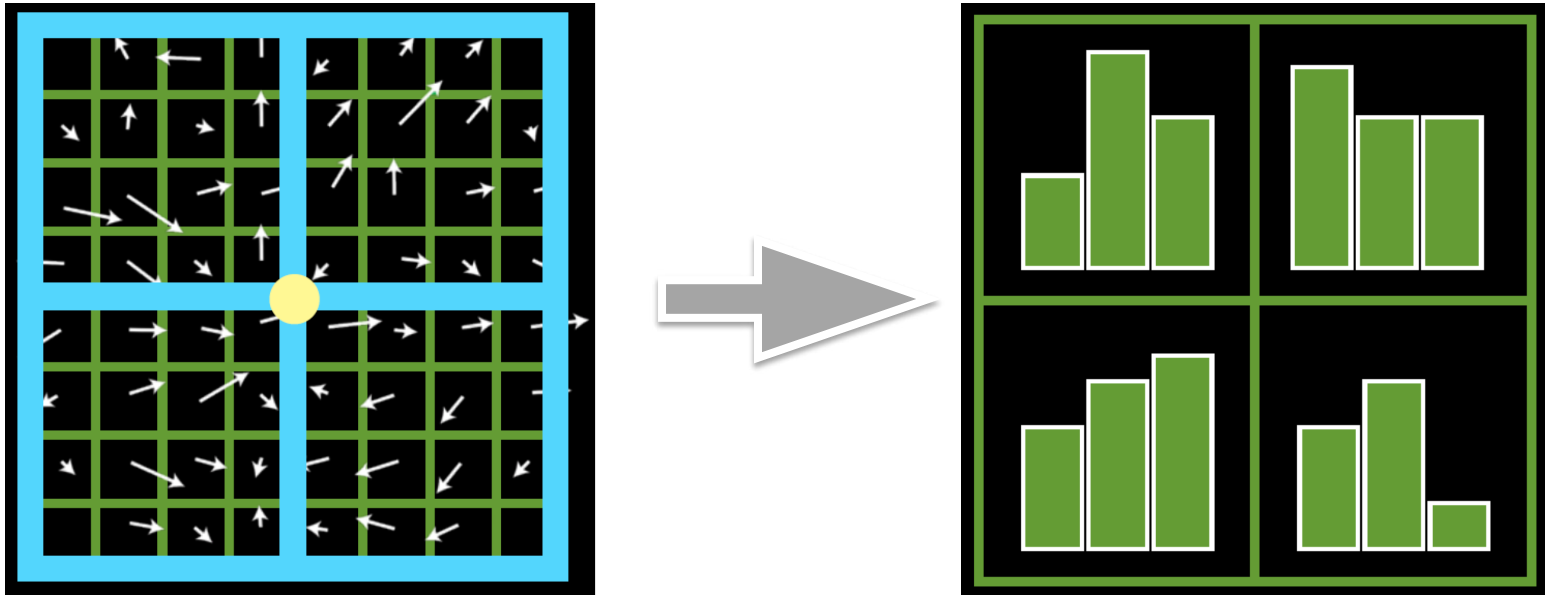
- Compute Image Gradients

- Accumulate gradients along cells

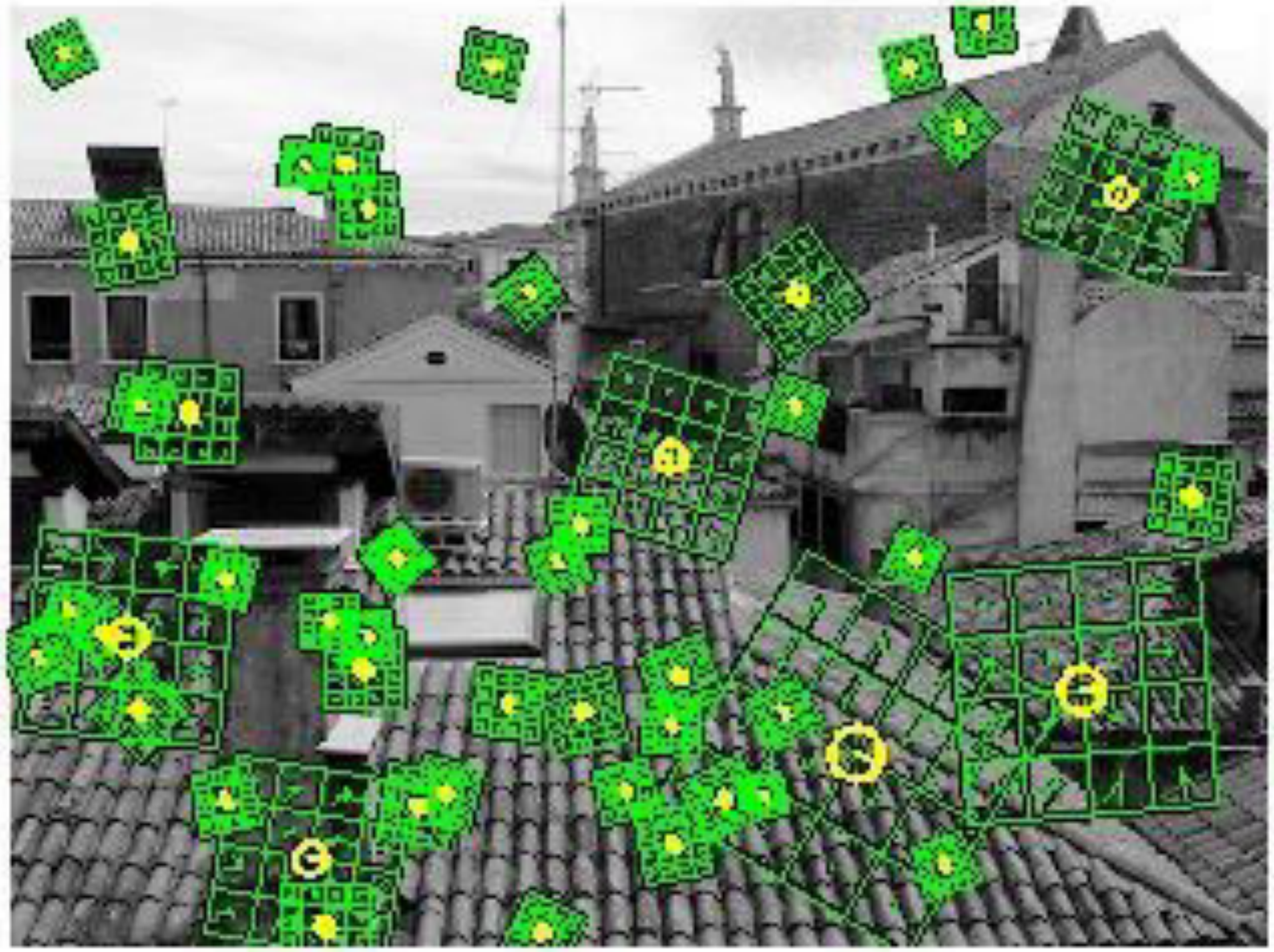


To extract a feature descriptor from a cell

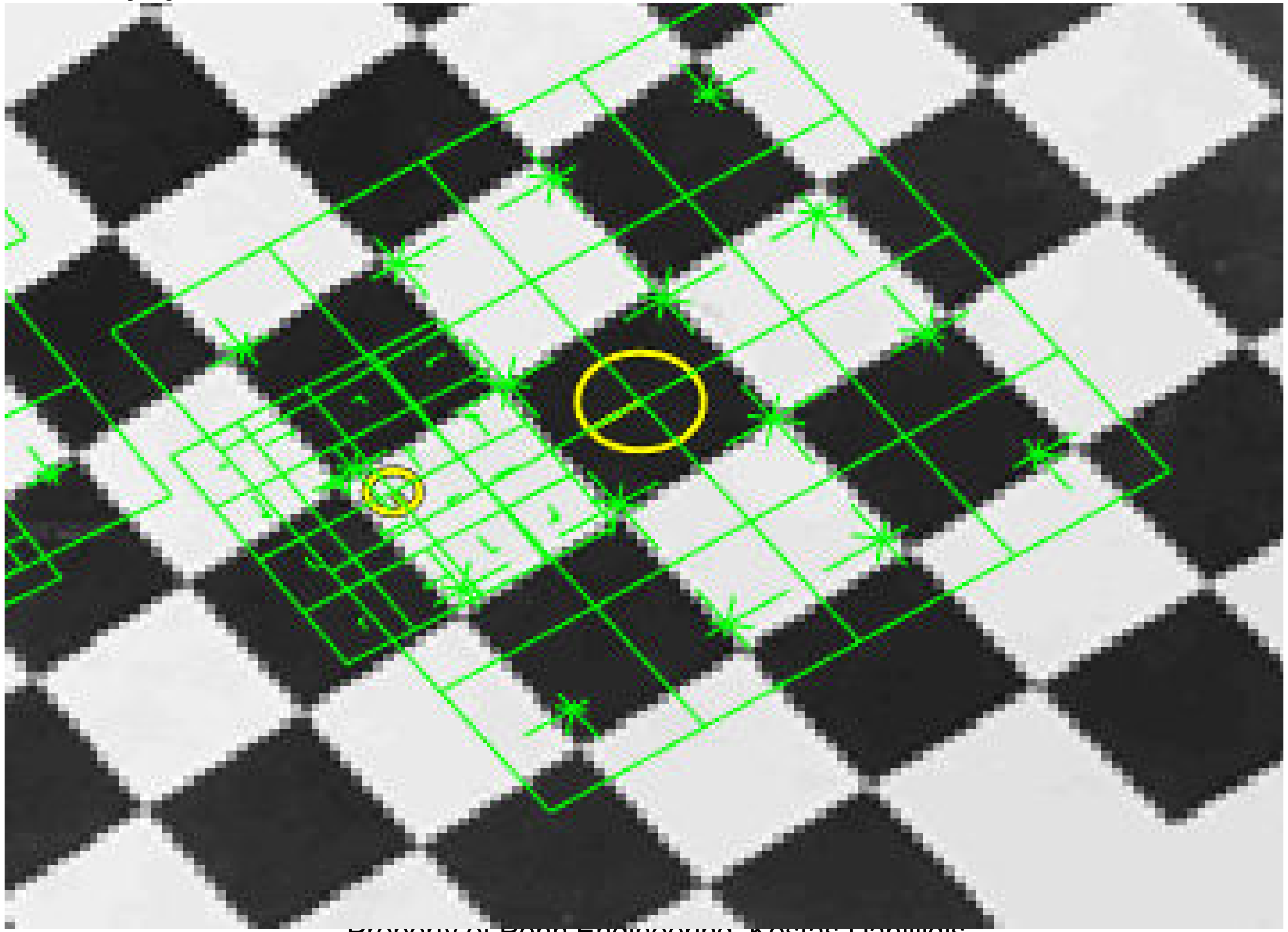
- Compute Image Gradients
- Accumulate gradients along cells
- Form image descriptor



As a matter of fact it is
a 4x4 grid of histograms at each keypoint

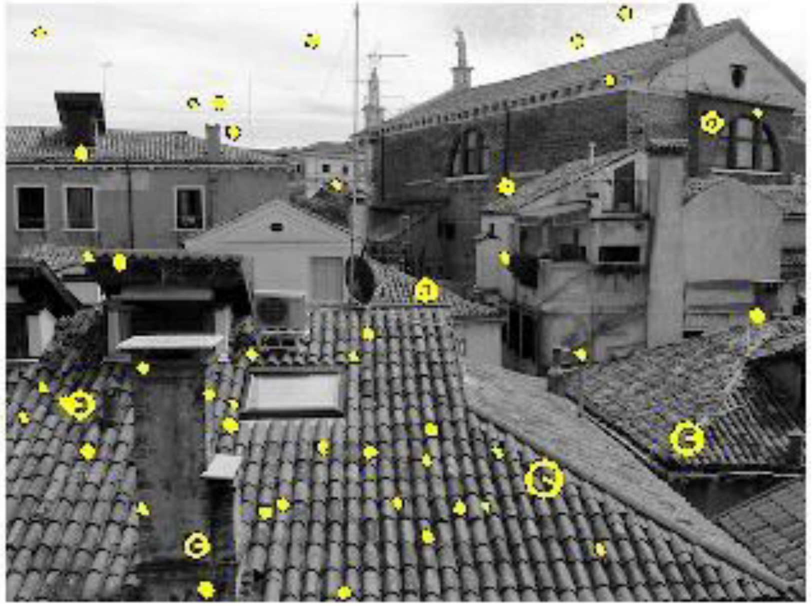


The descriptor is an 128x1 vector which together with σ,θ characterize the keypoint.



Example of SIFT detections and feature extraction

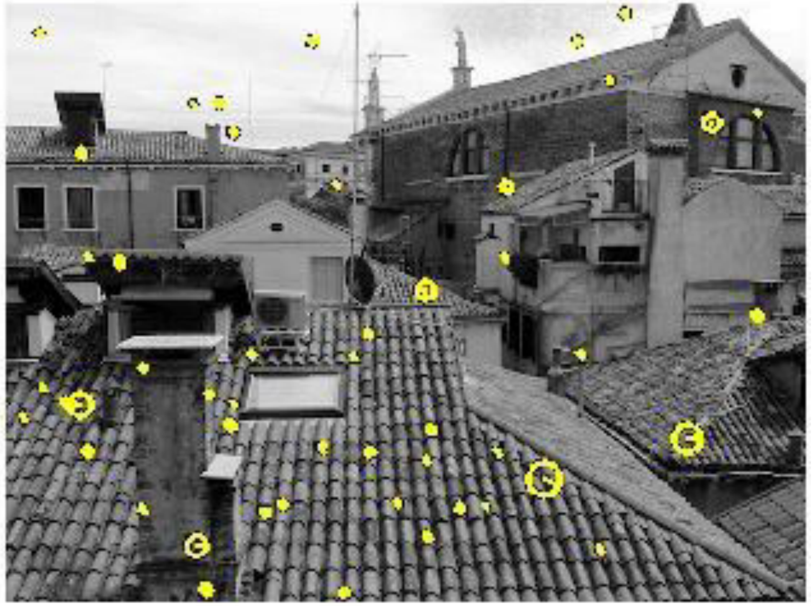
Input Image



Example Detections

Example of SIFT detections and feature extraction

Input Image



Example Detections

Extracted Feature Descriptors

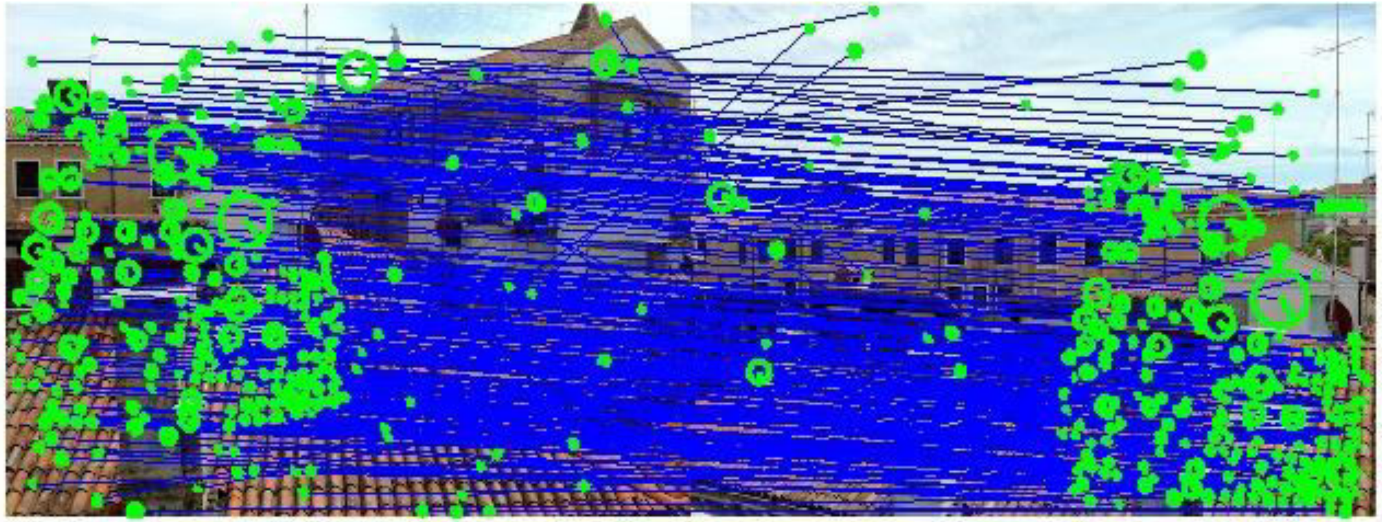
Property of Penn Engineerin

Using SIFT for image matching

Original Image Pair

Using SIFT for image matching

Original Image Pair

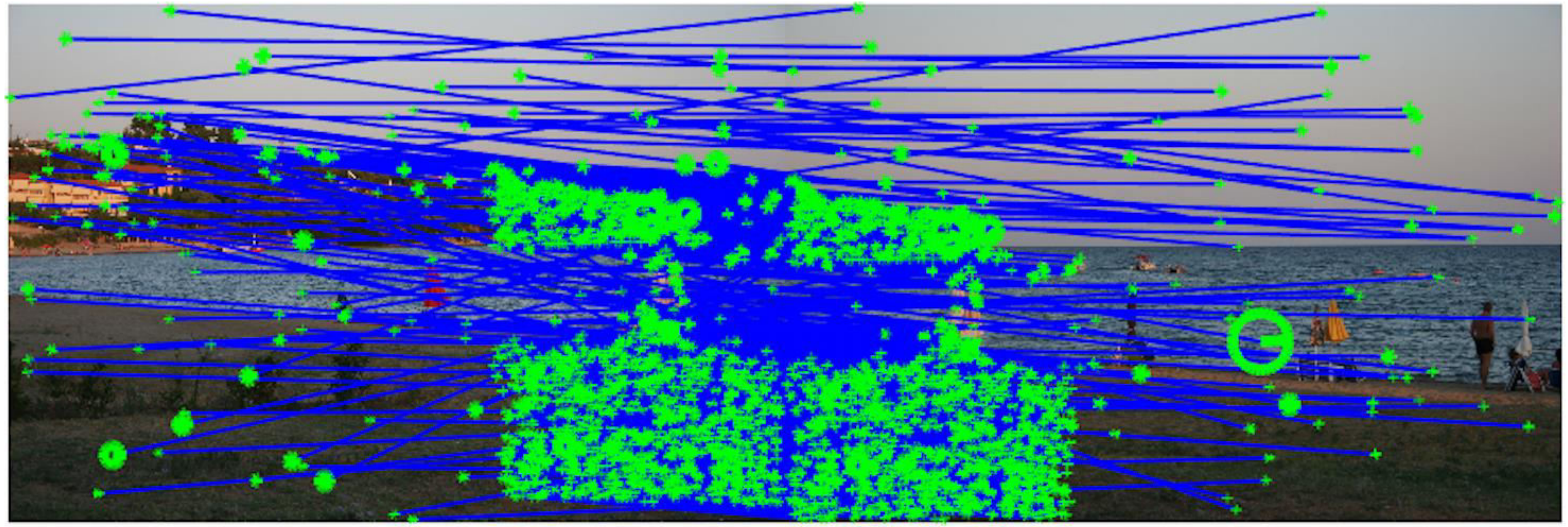


Matched features

Property of Penn Engineering, Kostas Daniilidis

Create Image Mosaic

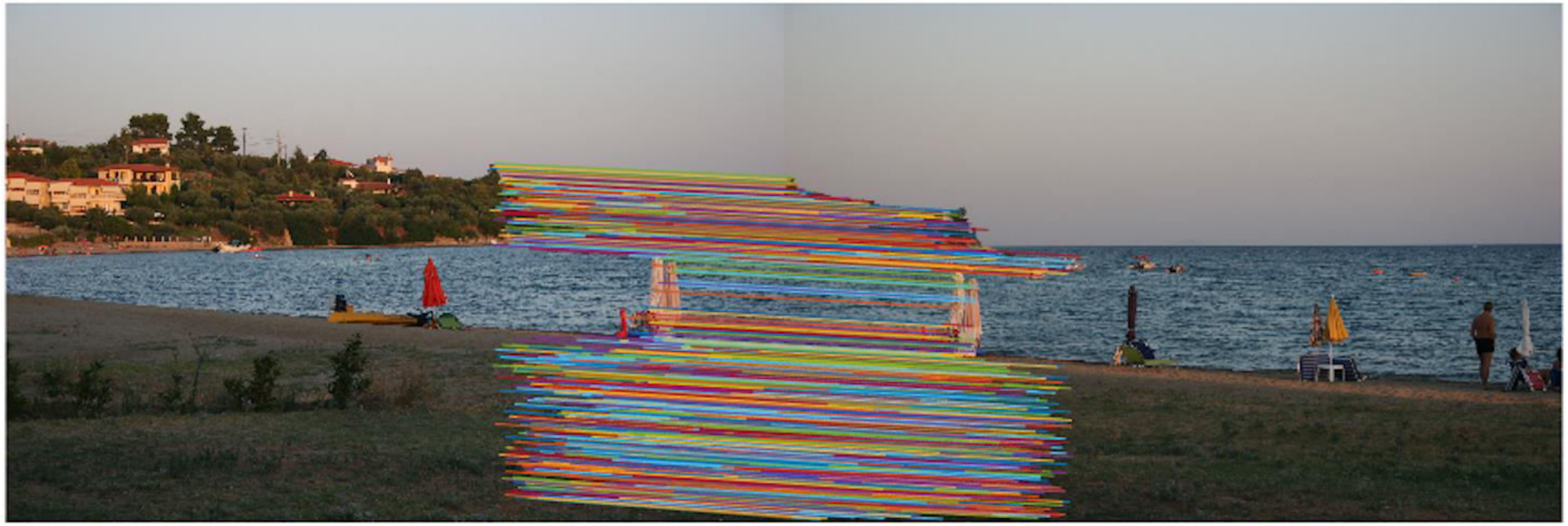
1. Get an image pair



2. Establish correspondences between matching features

Property of Penn Engineering, Kostas Daniilidis

Create Image Mosaic



3. Keep only consistent matches (inliers)

4. Compute homography and warp 2nd image

Property of Penn Engineering, Kostas Daniilidis

Create Image Mosaic

5. Repeat to extend the mosaic

Find Location

Query Image

We want to find a match in a dataset of given images

Find Location

Query Image

Find Location

Query Image

Good Matches

Medium Matches

Bad Matches

Property of Penn Engineering, Kostas Daniilidis

Find Location

Good Match

Medium Match

Bad Match

Property of Penn Engineering, Kostas Daniilidis

SIFT Features

- SIFT detector can automatically
  - Select scale
  - Compute dominant rotation
- SIFT descriptor
  - Is a grid of histogram of gradient orientations
  - On a region normalized with respect to scale and rotation

Video 4.4
Kostas Daniilidis

Property of Penn Engineering, Kostas Daniilidis

Image features

Why do we want features?

- As an image representation for
  - Matching two images
  - Image Statistics
- For what purpose?
  - AR tracking
  - Mosaicking
  - 3D models
  - Visual odometry
  - Video tracking

What is a good feature?

- A feature that enables matching when image undergoes
  - Geometric deformations
  - Photometric changes
  - Intra-class changes

- but at the same time is distinctive
- can be computed efficiently requires low storage

A set of features is an image representation!

Geometry: zoom and rotation

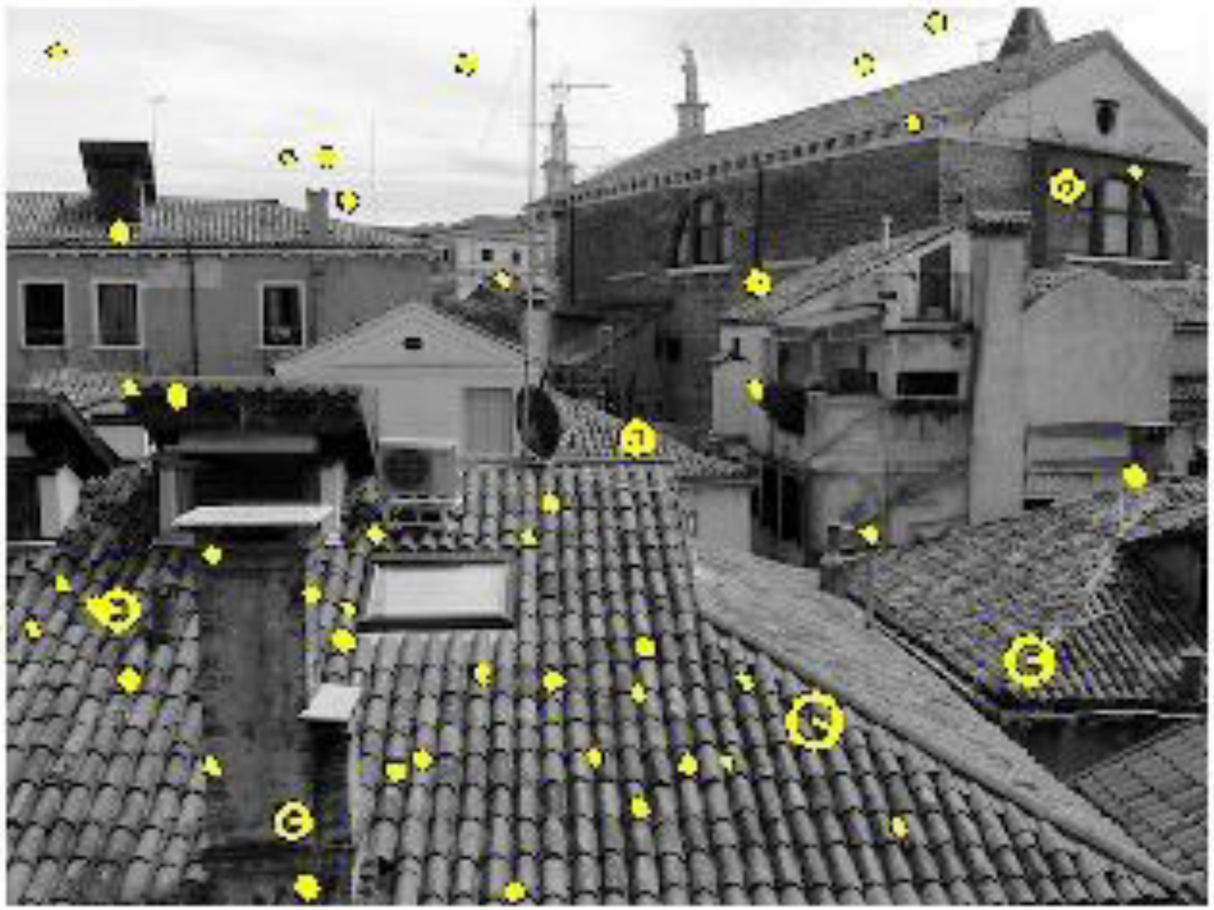
Photometry: Light and Color Changes

Photometry: Blurring

Intra-class changes

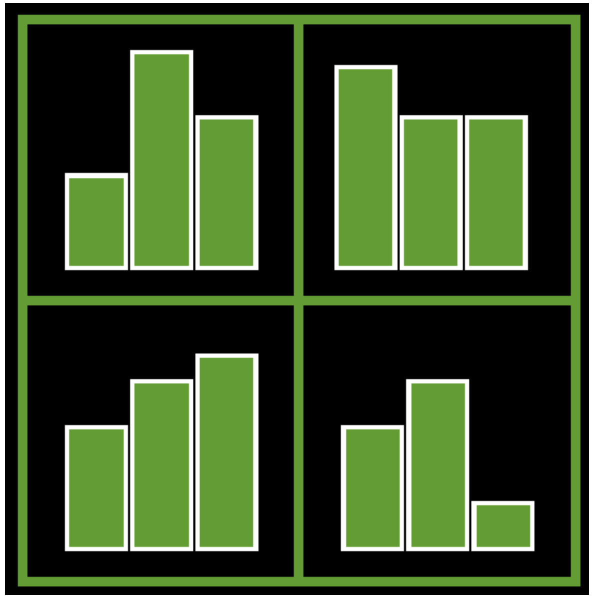
Detector and Descriptor

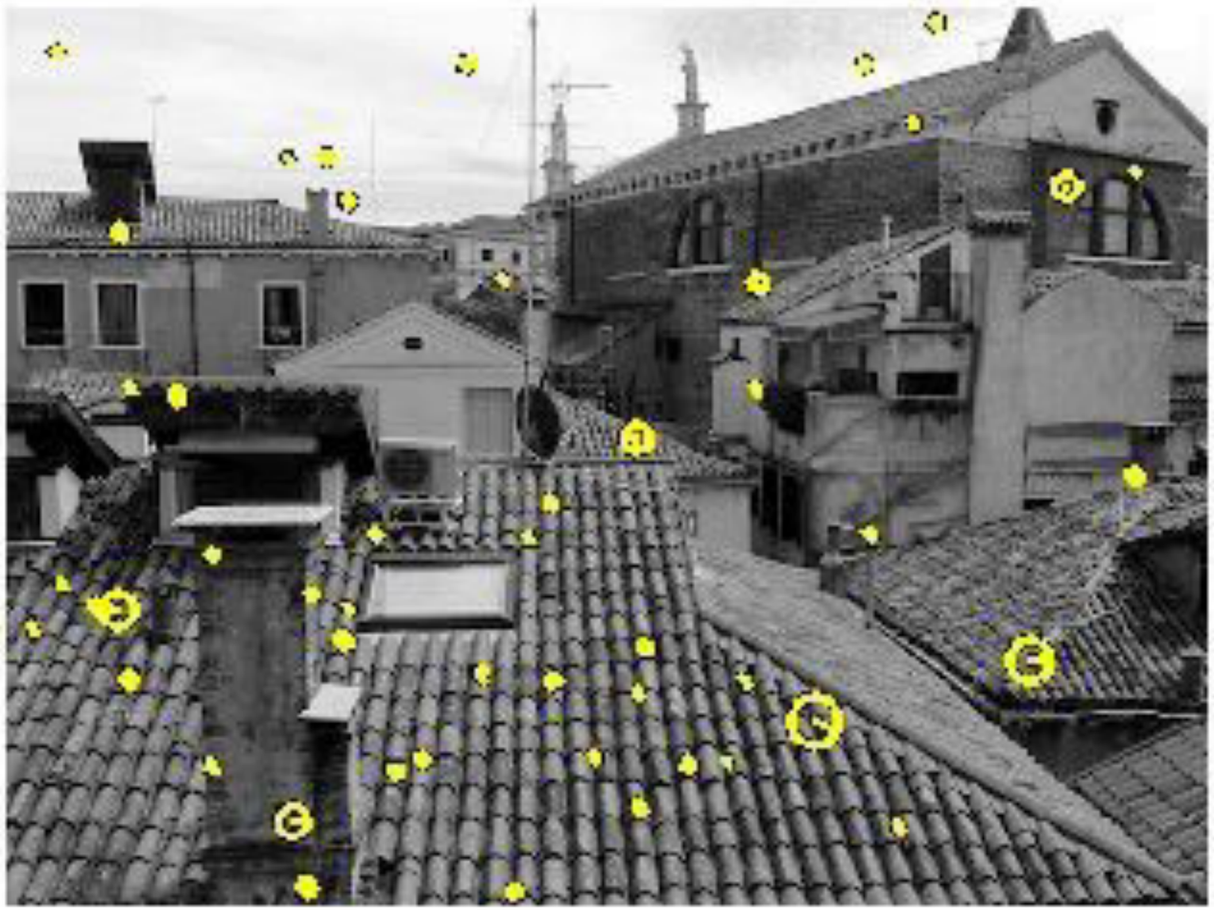
- Output of a detector are
  - Location
  - And possibly scale, orientation, affine, and other geometrics parameters



Detector and Descriptor

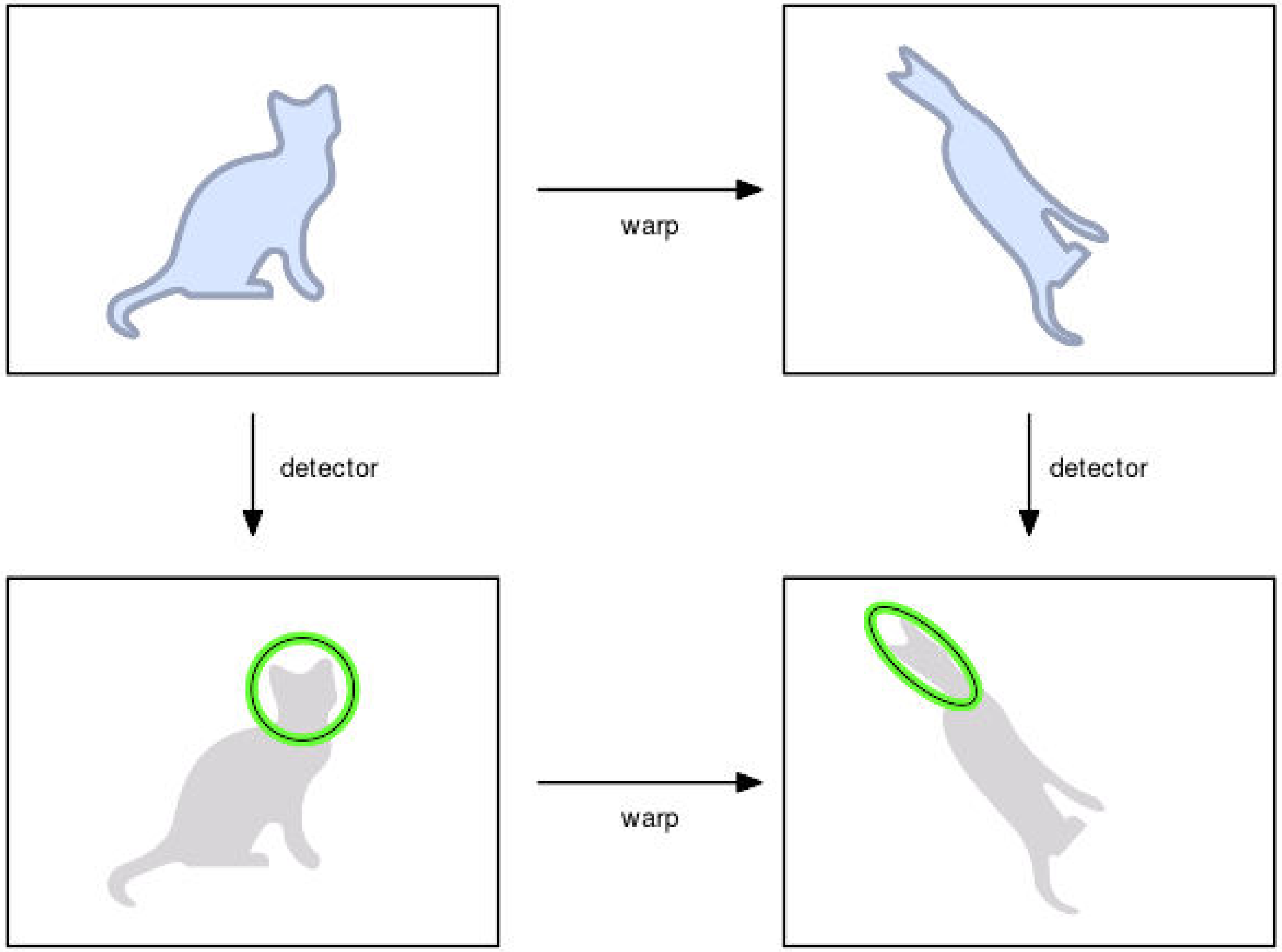
- Output of a descriptor is
  - A vector describing the neighborhood
  - For example a histogram!





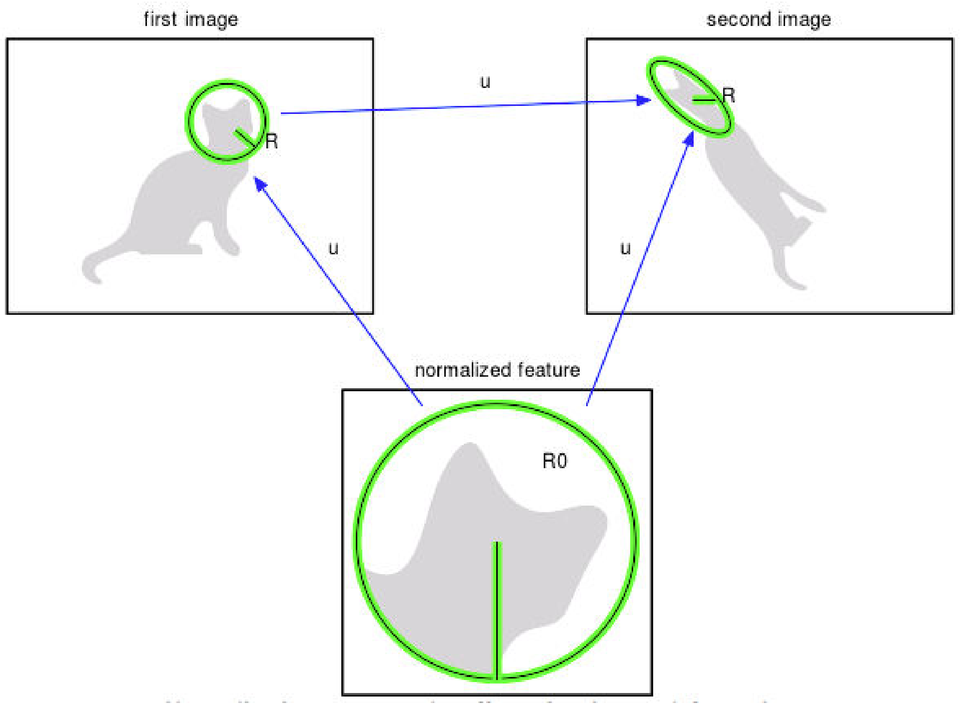
Geometric deformations: equivariance

- A feature has to be equivariant to deformations



Geometric deformations: invariance

A feature is invariant to deformations if its descriptor remains the same detected .



If the feature is equivariant, then invariance can be achieved with normalization.

Video 4.5
Kostas Daniilidis

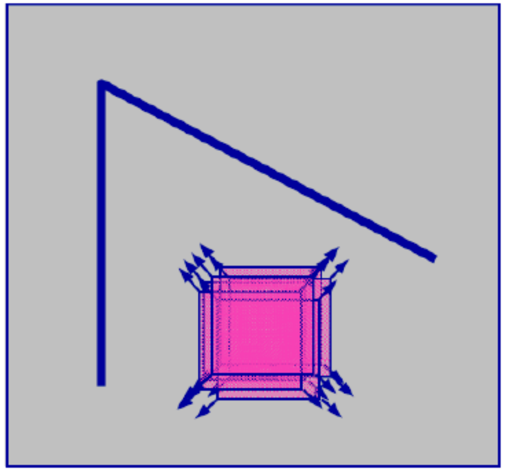
Harris Corners

A COMBINED CORNER AND EDGE DETECTOR

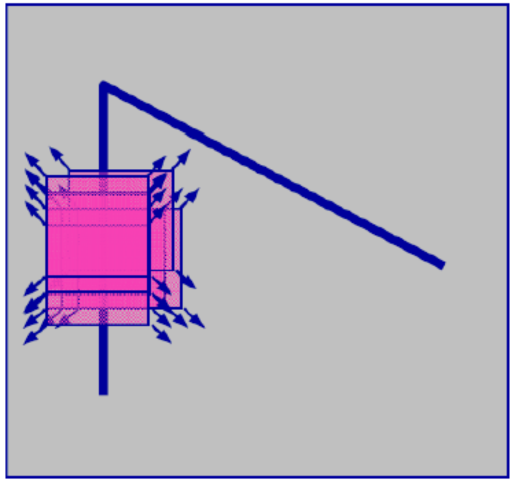
Chris Harris & Mike Stephens

Plessey Research Roke Manor, United Kingdom
© The Plessey Company plc. 1988

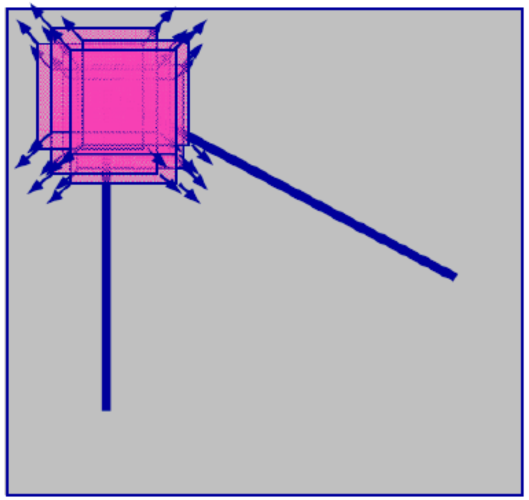
Harris corner detector: basic idea



flat region:
no change in any direction



edge:
no change along
the edge direction



corner:
change in all directions

Harris corner detector: derivation

Variation of intensity (expressed as weighted sum of squared distances) for a shift Δx around a fixed point x_0, using first order Taylor expansion:

$$
\begin{aligned}
E(\Delta x) &= \sum_{x \in \mathcal{N}(x_0)} w(x)\,[I(x+\Delta x) - I(x)]^2 \\
&\approx \sum_{x \in \mathcal{N}(x_0)} w(x)\,\left[(I(x) + \nabla I(x)^T \Delta x) - I(x)\right]^2 \\
&= \sum_{x \in \mathcal{N}(x_0)} w(x)\,\left[\nabla I(x)^T \Delta x\right]^2 \\
&= \sum_{x \in \mathcal{N}(x_0)} w(x) \Delta x^T \nabla I(x) \nabla I(x)^T \Delta x \\
&= \Delta x^T A \Delta x
\end{aligned}
$$

where A is the autocorrelation matrix, w is a weighting function (e.g. Gaussian centered at x_0) and $\mathcal{N}(x_0)$ a neighborhood of x_0.

Property of Penn Engineering, Kostas Daniilidis

Harris corner detector: autocorrelation matrix

The autocorrelation matrix A (evaluated at x_0) captures the variation of the gradients within the local patch. It can be rewritten as

$$\begin{aligned} A(x_0) &= \sum_{x \in \mathcal{N}(x_0)} w(x) \nabla I(x) \nabla I(x)^T \\ &= \sum_{x \in \mathcal{N}(x_0)} w(x) \begin{pmatrix} I_x(x) \\ I_y(x) \end{pmatrix} \begin{pmatrix} I_x(x) & I_y(x) \end{pmatrix} \\ &= \sum_{x \in \mathcal{N}(x_0)} w(x) \begin{pmatrix} I_x(x)^2 & I_x(x) I_y(x) \\ I_x(x) I_y(x) & I_y(x)^2 \end{pmatrix} \end{aligned}$$

where $I_x = \partial I / \partial x$ and $I_y = \partial I / \partial y$.

Property of Penn Engineering, Kostas Daniilidis

Harris corner detector: autocorrelation matrix

By replacing the weighted summations with discrete convolutions with the weighting kernel w, the autocorrelation matrix can be written as a convolution:

$$A = w * \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

where $*$ denotes convolution. Observe that A is positive semidefinite since

$$\Delta x^T A \Delta x = E(\Delta x) \geq 0$$

for all $\Delta x \in \mathbb{R}^2$ since E is a sum of square distances.
But why do we care if A is positive semidefinite after all? (*Hint*: think of eigenvalues!)

Harris corner detector: autocorrelation matrix

Since $A \succeq 0$ it follows that it can be decomposed as following:

$$A = Q\Lambda Q^T$$

where $Q = [\ q_1 \mid q_2\ ]$ is the matrix with columns the eigenvectors of A and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2)$ where λ_1, λ_2 are the two *nonnegative* eigenvalues of A.

$$\begin{aligned} E(\Delta x) &= \Delta x^T A \Delta x \\ &= \Delta x^T Q\Lambda Q^T \Delta x \\ &= \Delta y^T \Lambda \Delta y \\ &= \frac{\Delta y_1^2}{(1/\sqrt{\lambda_1})^2} + \frac{\Delta y_2^2}{(1/\sqrt{\lambda_2})^2} \end{aligned}$$

where $\Delta y = Q^T \Delta x$. Note that $\Delta y_1 = q_1^T \Delta x$ and $\Delta y_2 = q_2^T \Delta x$.

Harris corner detector: autocorrelation matrix

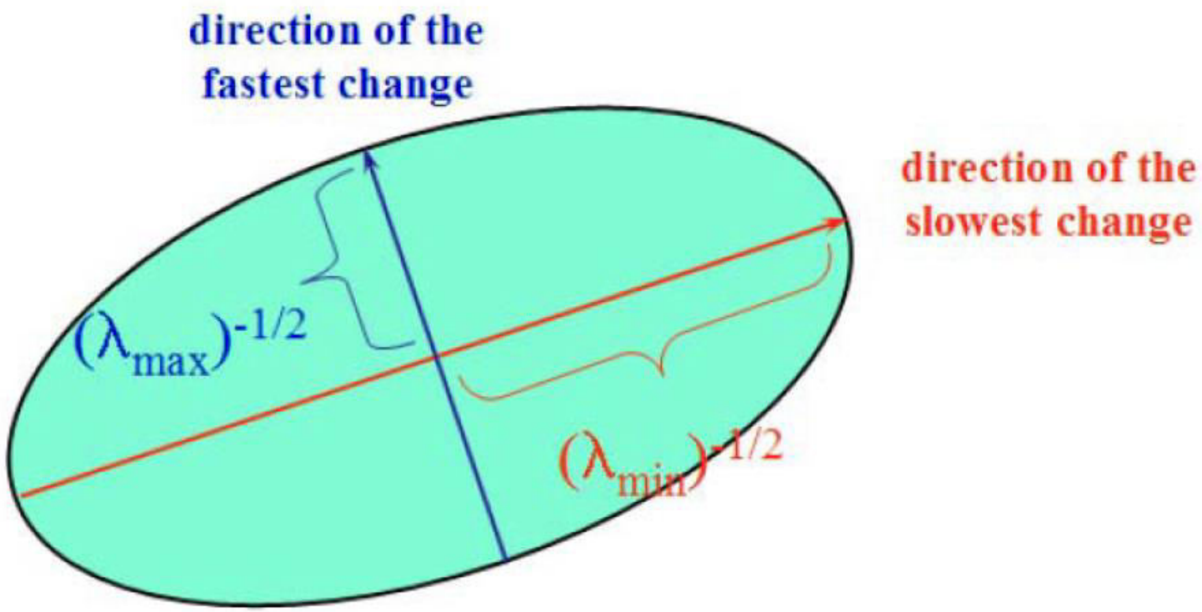
Note that always Q can be a rotation matrix in the spectral decomposition of A. Why? (Hint: the negated version of an eigenvector is an eigenvector too.) Let

$$Q = R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Since $\Delta y = Q^T \Delta x$ it follows that the directions Δy_1 and Δy_2 result from Δx_1 and Δx_2 by a rotation of θ.

Harris corner detector

What does the above formula of $E(\Delta x)$ in terms of Δy_1 and Δy_2 reminds you of? Yes, an ellipse!



Uncertainty ellipse corresponding to an eigenvalue analysis of the autocorrelation matrix A.

Property of Penn Engineering, Kostas Daniilidis

Harris corner detector

Observations:

1. If λ_1, λ_2 are very small, then a large shift does not produce a significant variation of intensity and thus, the patch corresponds to a flat region.
2. If $\lambda_1 >> \lambda_2$, then a shift along the direction Δy_1 produces a significant variation of intensity and thus, the patch corresponds to an edge.
3. If λ_1, λ_2 are large, then a shift along any direction produces a significant variation of intensity and thus, the patch corresponds to a corner.

Harris corner detector

How can we quantify the above observations?

$$\begin{aligned} R &= \det(A) - k \cdot \mathrm{trace}(A)^2 \\ &= \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 \end{aligned}$$

where $k = 0.06$ is an empirical constant.

1. If λ_1, λ_2 are very small, then R is very small eliminating flat regions.
2. If $\lambda_1 >> \lambda_2$, then R is very small eliminating edge responses.
3. If λ_1, λ_2 are large, R is large as desired.

Is the Harris "autocorrelation matrix" an equivariant feature?

$$\sum_{x \in \mathcal{N}(x_0)} w(x) \begin{pmatrix} I_x(x)^2 & I_x(x)I_y(x) \\ I_x(x)I_y(x) & I_y(x)^2 \end{pmatrix}$$

Rotation: If the image is rotated will the matrix rotate as well?
Yes! And the eigendecomposition will yield rotation.

Scaling: If the image is rotated will the matrix scale as well?
Yes but the detection threshold will have to be adapted.
Only the descriptor matrix is equivariant.

Harris corner detector: implementation sketch

1. Read input image and turn it to grayscale if necessary.
2. Smooth input image with a gaussian filter.
3. Convolve input image with filter $h = \begin{bmatrix} 1/2 & 0 & -1/2 \end{bmatrix}$ to obtain the first order derivatives:

$$I_x = I * h \qquad I_y = I * h^T$$

4. Compute images A, B, C at pixel (u, v) as defined below:

$$A(u,v) = I_x(u,v)^2 \quad B(u,v) = I_y(u,v)^2 \quad C(u,v) = I_x(u,v) \cdot I_y(u,v)$$

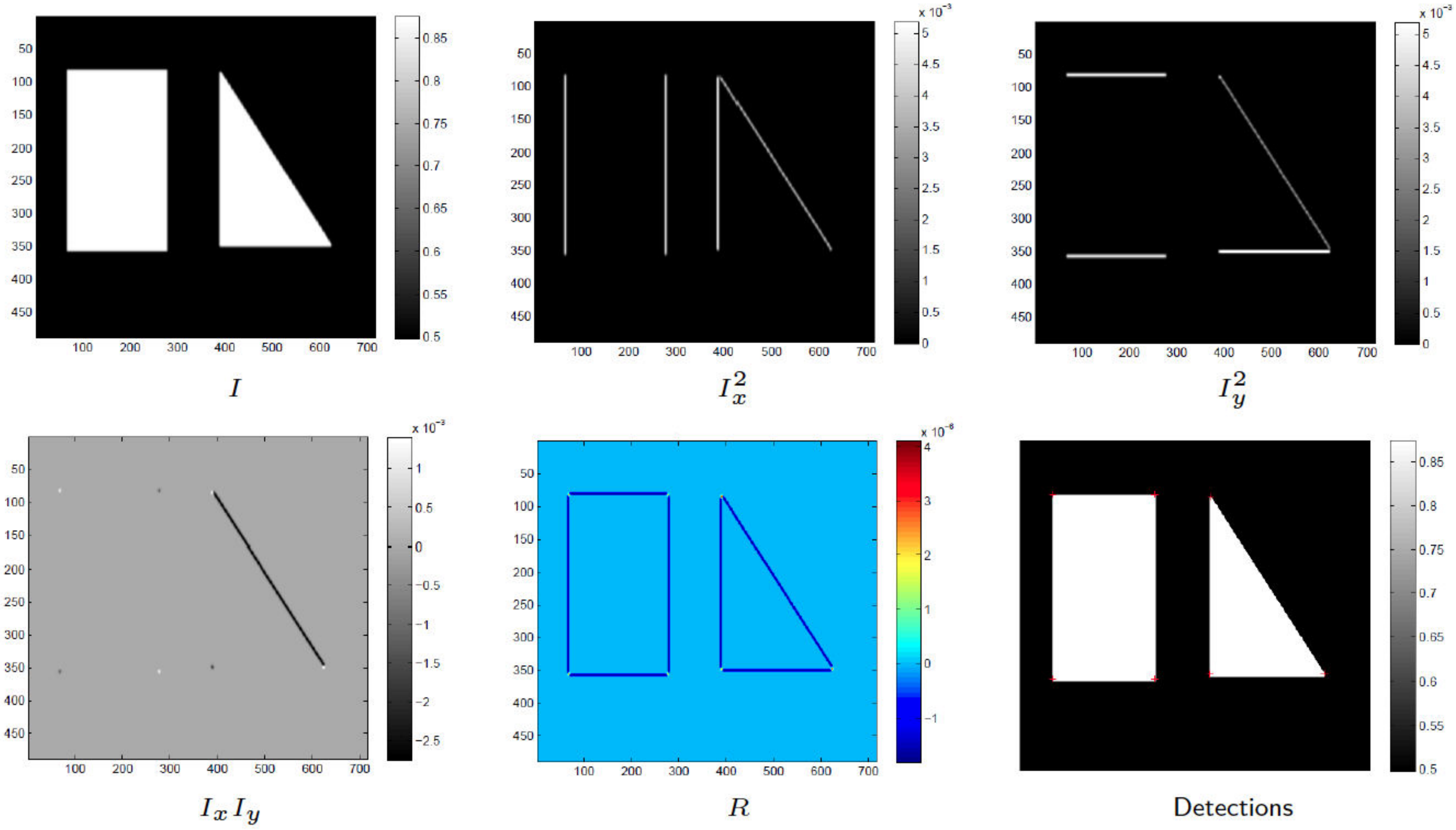
5. Filter A, B, C with a gaussian filter.
6. Compute corner response at each pixel (u, v)

$$R(u,v) = (A(u,v) \cdot B(u,v) - C(u,v)^2) - k(A(u,v) + B(u,v))^2$$

with $k = 0.06$.

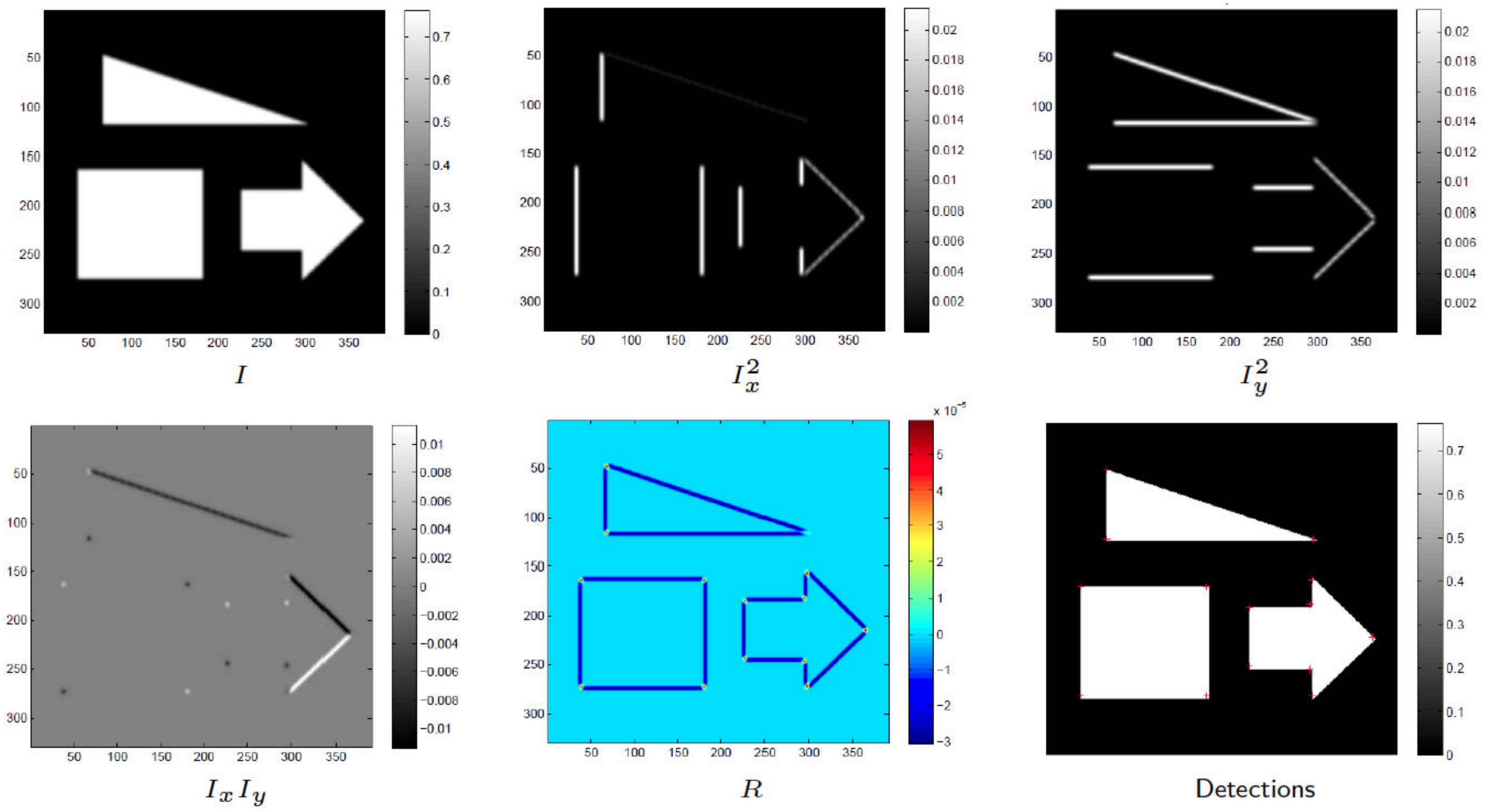
7. Detect local maxima of R.
8. Keep local maxima of R whose values are above a threshold.

Harris corner detector: visualization

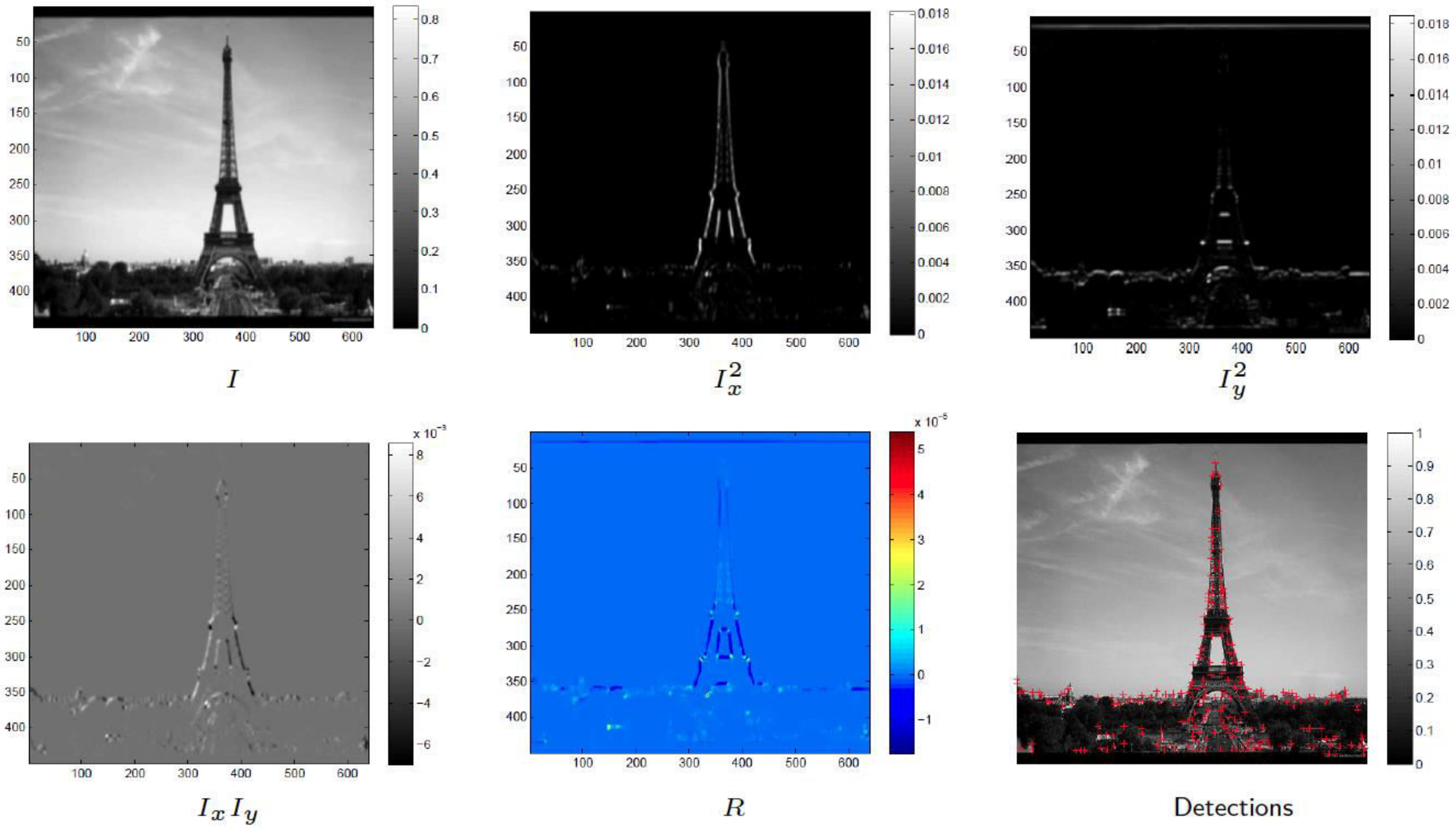


Property of Penn Engineering, Kostas Daniilidis

Harris corner detector: visualization



Harris corner detector: visualization



Video 4.6
Kostas Daniilidis

Property of Penn Engineering, Kostas Daniilidis

MSER: Maximally Stable Extremal Regions, Matas et al. 2001

Robust wide-baseline stereo from maximally stable extremal regions

J. Matas[a,b,*], O. Chum[a], M. Urban[a], T. Pajdla[a]

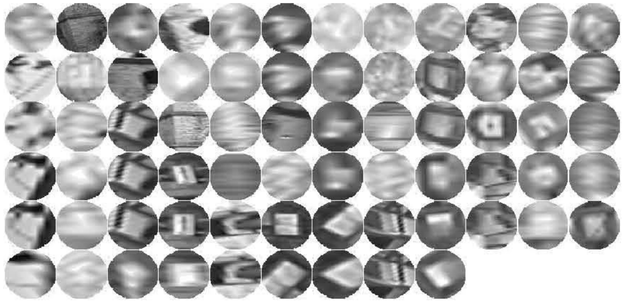
[a]*Department of Cybernetics, Center for Machine Perception, CTU Prague, Karlovo nám 13 CZ 121 35, Czech Republic*
[b]*CVSSP, University of Surrey, Guildford GU2 7XH, UK*

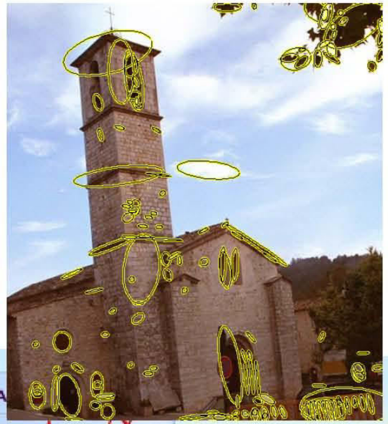
Received 12 March 2003; received in revised form 11 February 2004; accepted 12 February 2004

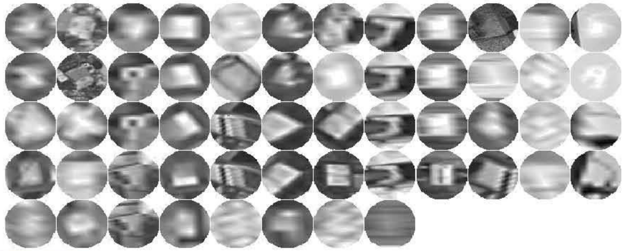
MSER: Maximally Stable Extremal Regions, Matas et al. 2001

“uniform blobs”

Regions unchanged over many thresholds

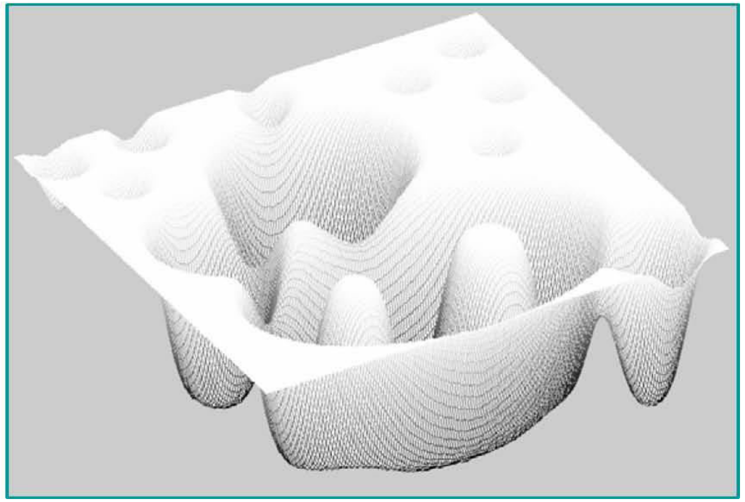






MSER Construction (1)

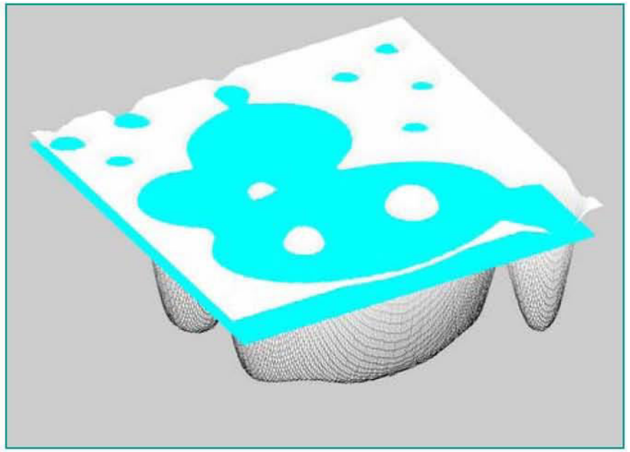
intensity image



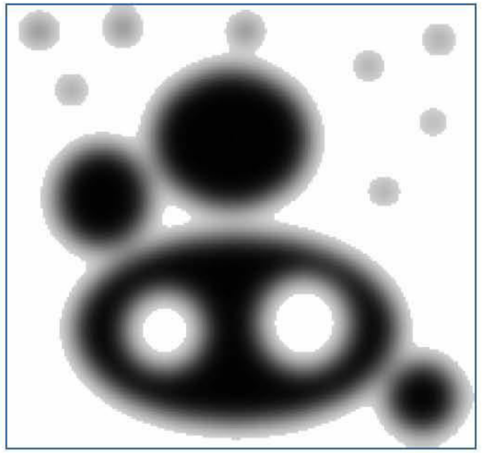
shown as a surface function

Watershed segmentation algorithms come from the concept of filling a basin with water to different levels.

MSER Construction (2)



Threshold simulation



Extremal Regions (represented by their original lumiance values)

For each region, and for each threshold value, the region area is saved.

IMAGE AND SOUND ISPG PROCESSING GROUP

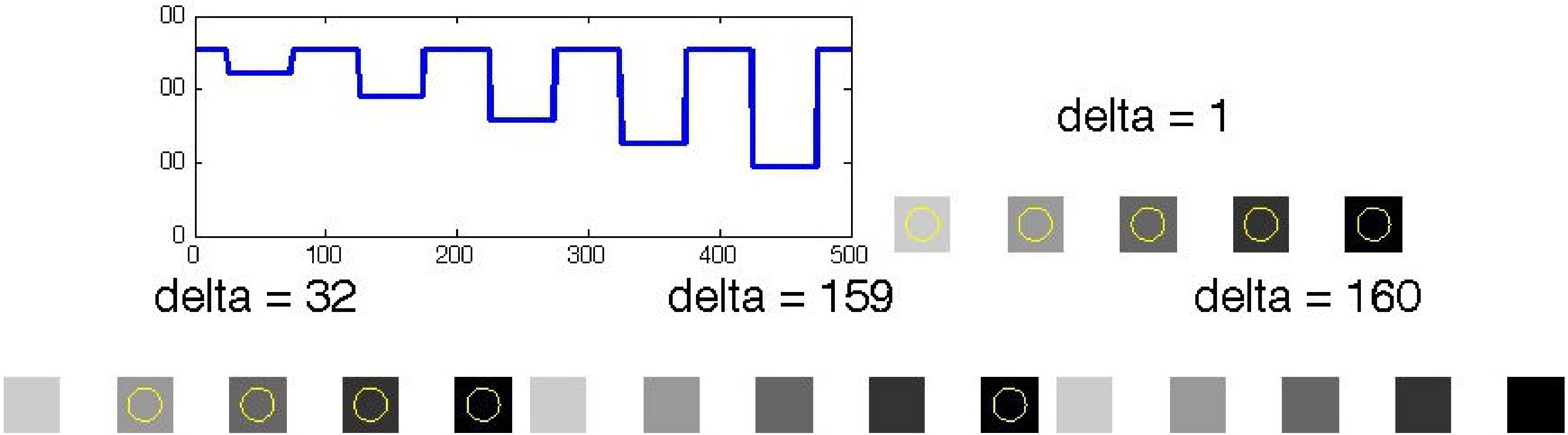
MSER Computation (3)

- For each threshold, compute the connected binary regions.

- Compute a function, area A(i), at each threshold value i.

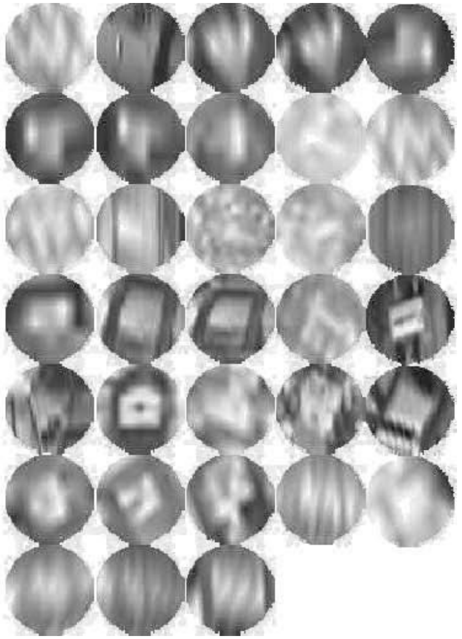
- Analyze this function for each potential region to determine those that persist with similar function value over multiple thresholds.

A single threshold

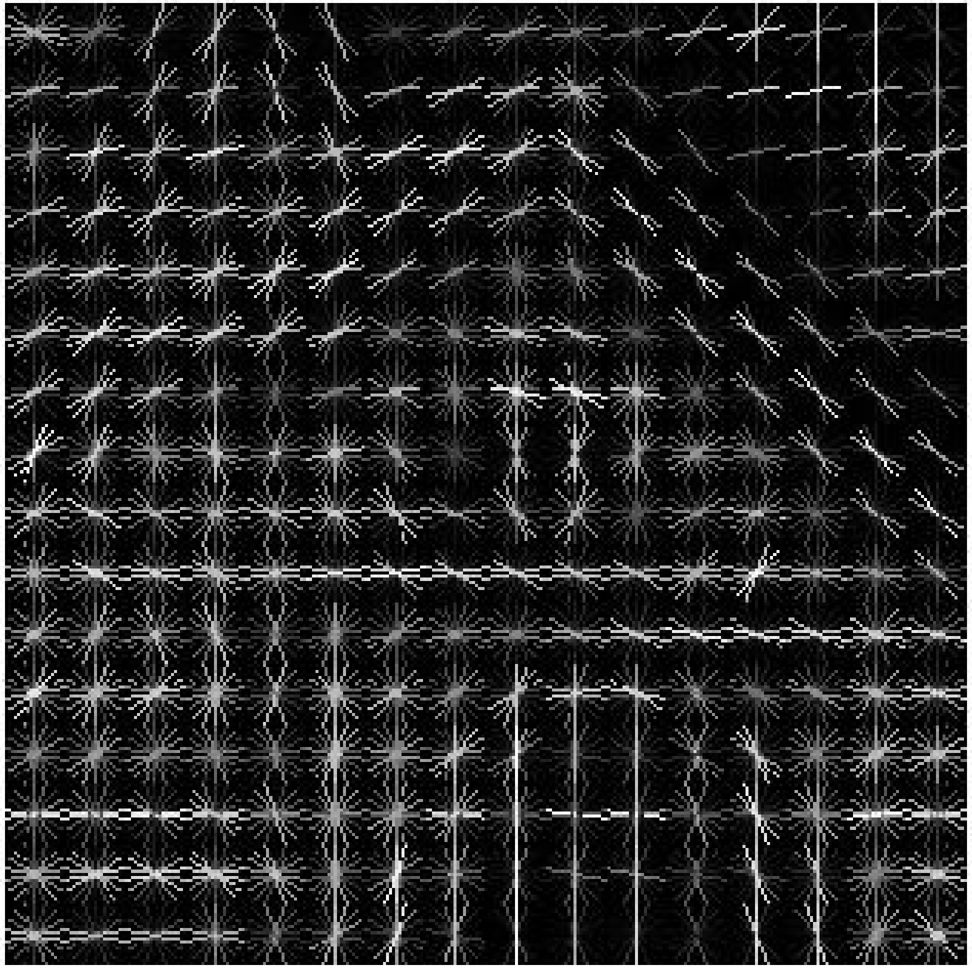


Effect of Δ. *We start with a synthetic image which has an intensity profile as shown. The bumps have heights equal to 32, 64, 96, 128 and 160. As we increase Δ, fewer and fewer regions are detected until finally at Δ=160 there is no region R which is stable at R(+Δ).*

Affine transformation from ellipses to circular regions plus intensity normalization



Histogram of gradients (HOG) descriptor (Dalal and Triggs)



Property of Penn Engineering, Kostas Daniilidis

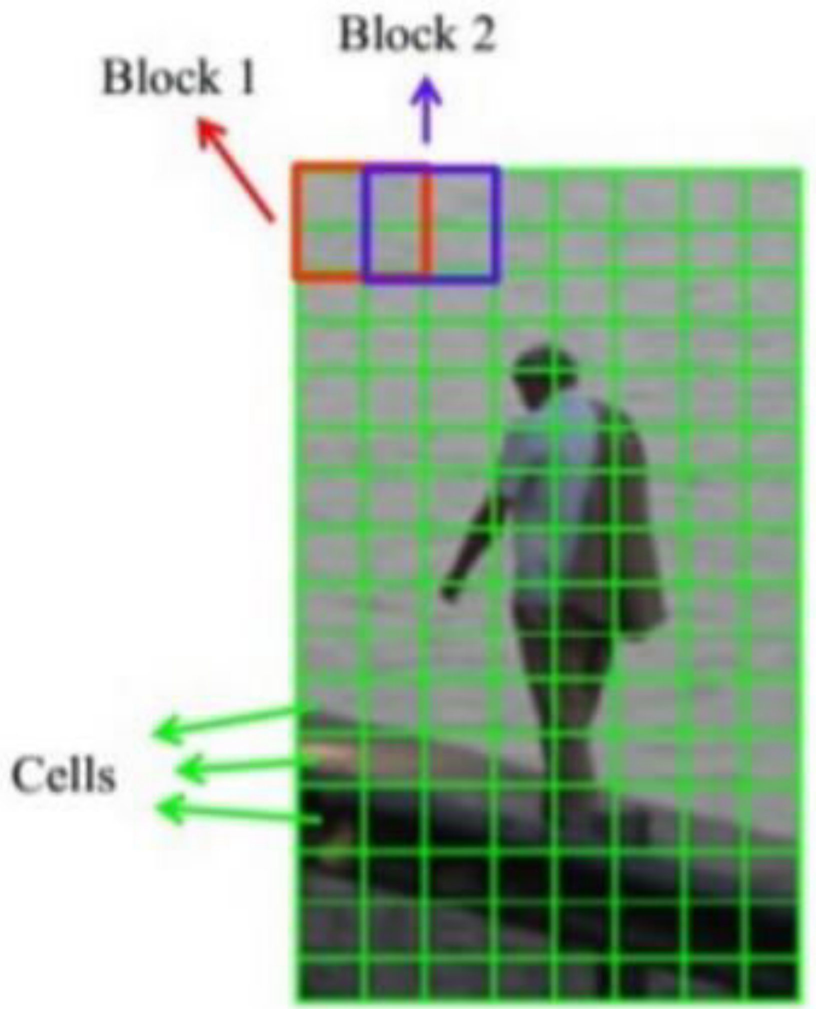
Histogram of gradients (HOG) descriptor

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Image is split in overlapping blocks of cells

- Each block is 16x16
- 50% overlap

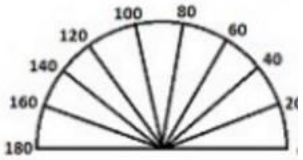


- Each block has 2x2 cells
- Each cell is 8x8

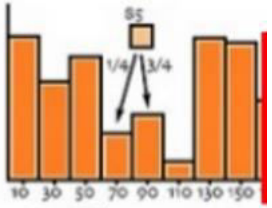
Histogram quantization

Tri-linear Interpolation

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins $\boxed{(0\text{-}180)}$



9 Bins



Bin centers

  - The vote is the gradient magnitude
  - Interpolate votes linearly between neighboring bin centers.
    - Example: if θ=85 degrees.
    - Distance to the bin cente Bin 70 and Bin 90 are 15 and 5 degrees, respectively.
    - Hence, ratios are 5/20=1/4, 15/20=3/4.
  - The vote can also be weighted with Gaussian to downweight the pixels near the edges of the block.

VLFeat.org

Google Custom Search | Search ×

Home | Download | Tutorials | Applications | Documentation

Home

ACM OpenSource Award

The **VLFeat** open source library implements popular computer vision algorithms specializing in image understanding and local features extraction and matching. Algorithms include Fisher Vector, VLAD, SIFT, MSER, k-means, hierarchical k-means, agglomerative information bottleneck, SLIC superpixels, quick shift superpixels, large scale SVM training, and many others. It is written in C for efficiency and compatibility, with interfaces in MATLAB for ease of use, and detailed documentation throughout. It supports Windows, Mac OS X, and Linux. The latest version of VLFeat is `0.9.20`.

Download

- **VLFeat 0.9.20** (Windows, Mac, Linux)
- Source code and installation
- git repository, bug tracking.

Tutorials

- Features: Covariant detectors, HOG, SIFT, MSER, Quick shift, SLIC
- Clustering: IKM, HIKM, AIB
- Matching: Randomized kd-trees
- All tutorials

Example applications

- Caltech-101 classification
- SIFT matching for auto-stitching
- All example applications

Documentation

- MATLAB commands
- C API with algorithm descriptions
- Command line tools

Citing

```
@misc{vedaldi08vlfeat,
 Author = {A. Vedaldi and B. Fulkerson},
 Title = {{VLFeat}: An Open and Portable Library
          of Computer Vision Algorithms},
 Year  = {2008},
 Howpublished = {\url{http://www.vlfeat.org/}}
```

Acknowledgments

PASCAL2 Yandex UCLA Vision Lab Oxford VGG.

International Journal of Computer Vision 65(1/2), 43–72, 2005
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s11263-005-3848-x

A Comparison of Affine Region Detectors

K. MIKOLAJCZYK
University of Oxford, OX1 3PJ, Oxford, United Kingdom
km@robots.ox.ac.uk

T. TUYTELAARS
University of Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
tuytelaa@esat.kuleuven.be

C. SCHMID
INRIA, GRAVIR-CNRS, 655, av. de l'Europe, 38330, Montbonnot, France
schmid@inrialpes.fr

A. ZISSERMAN
University of Oxford, OX1 3PJ, Oxford, United Kingdom
az@robots.ox.ac.uk

J. MATAS
Czech Technical University, Karlovo Namesti 13, 121 35, Prague, Czech Republic
matas@cmp.felk.cvut.cz

F. SCHAFFALITZKY AND T. KADIR
University of Oxford, OX1 3PJ, Oxford, United Kingdom
fsm@robots.ox.ac.uk
tk@robots.ox.ac.uk

L. VAN GOOL
University of Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
vangool@esat.kuleuven.be

Received August 20, 2004; Revised May 3, 2005; Accepted May 11, 2005

First online version published in January, 2006