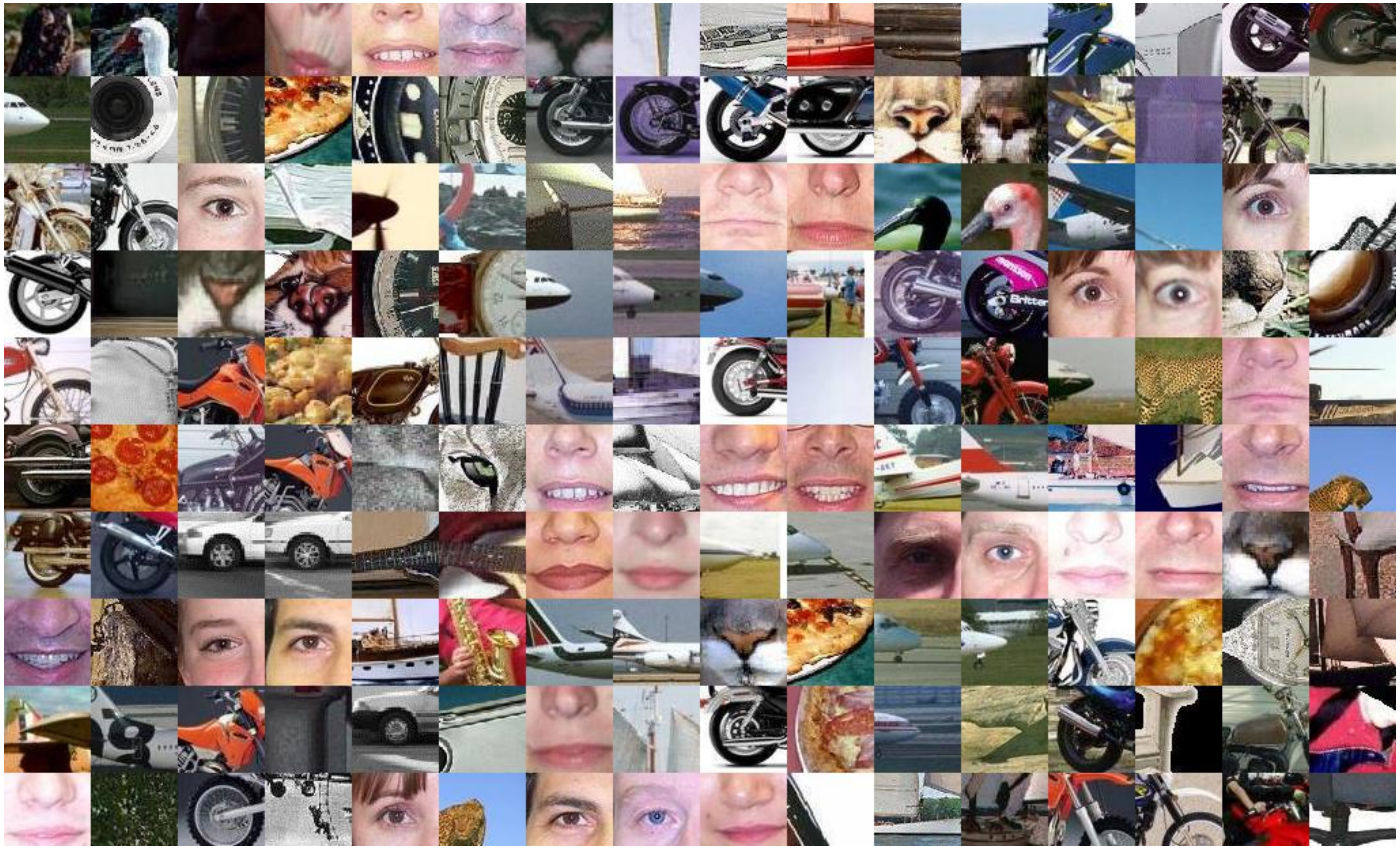
Feature detectors and descriptors



16-385 Computer Vision
Spring 2019, Lecture 6

http://www.cs.cmu.edu/~16385/

Course announcements

- Homework 1 is due tonight at 23:59!
- Homework 2 will be posted tonight and will be due on Monday, February 25th.
- There are additional office hours today: 3-5pm, covered by Abhay, at the graphics lounge in Smith Hall.

Overview of today's lecture

Leftover from lecture 5:

- Finish Harris corner detector.
- Multi-scale detection.

New in lecture 6:

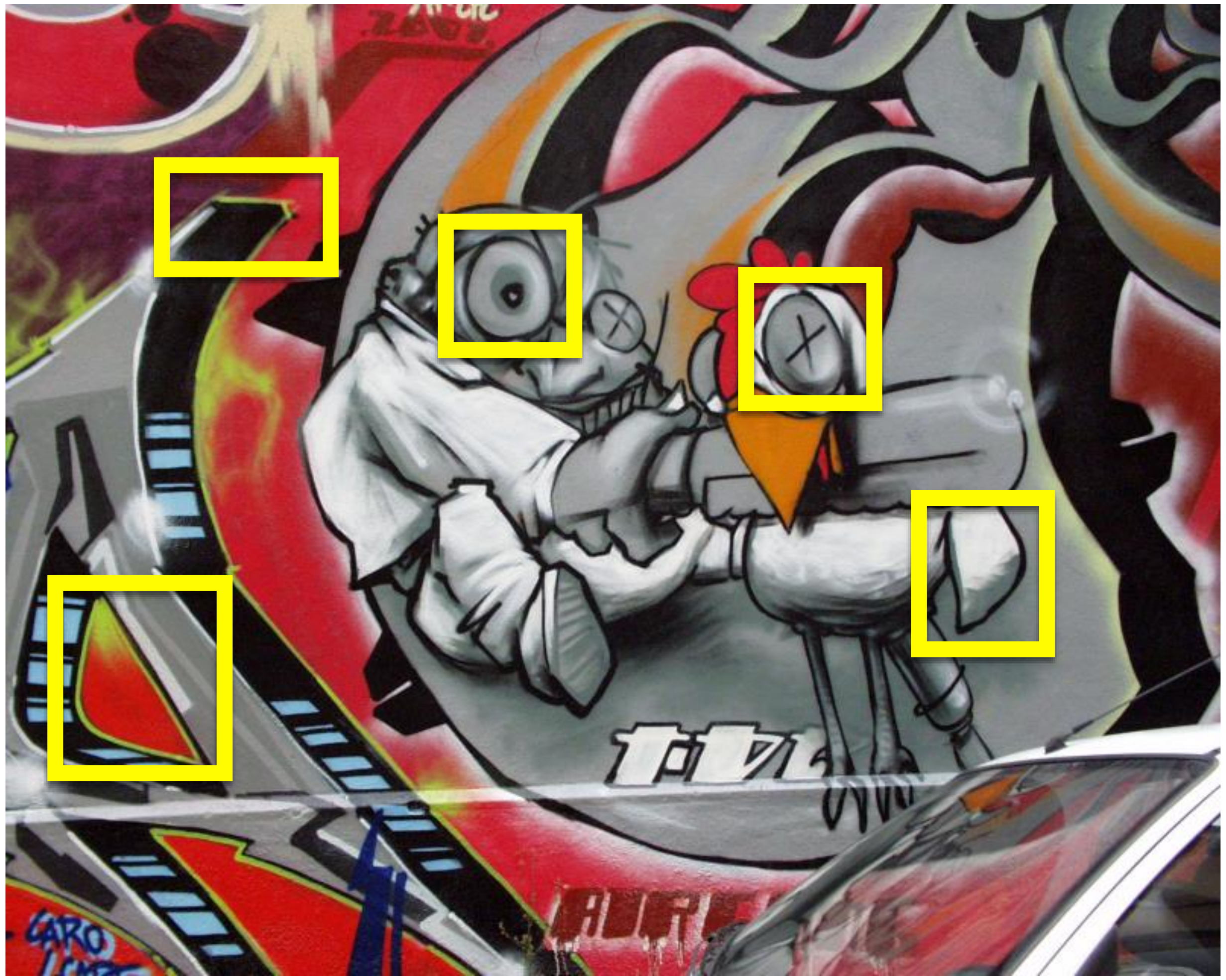
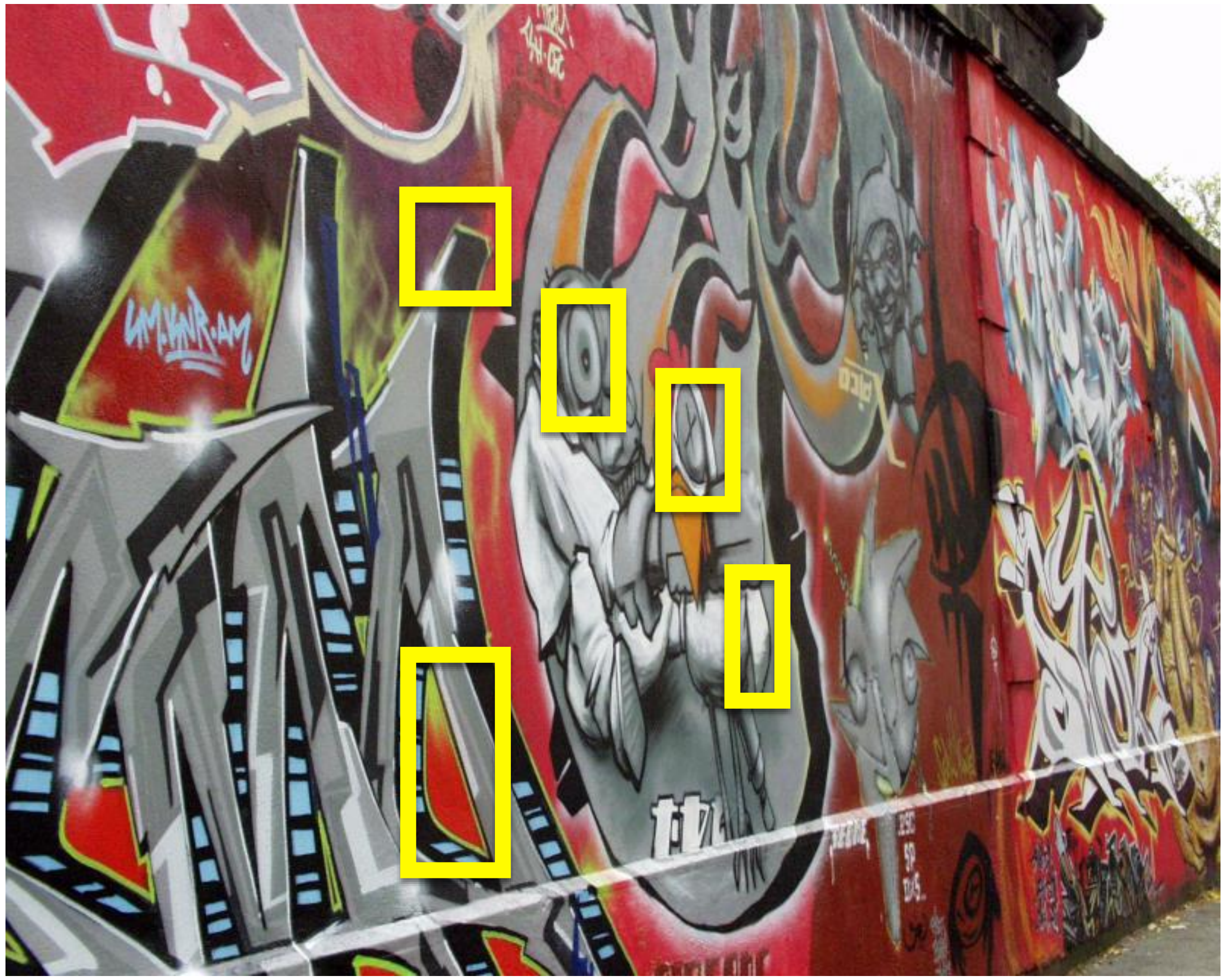
- Why do we need feature descriptors?
- Designing feature descriptors.
- MOPS descriptor.
- GIST descriptor.
- Histogram of Textons descriptor.
- HOG descriptor.
- SIFT.

Slide credits

Most of these slides were adapted from:

- Kris Kitani (16-385, Spring 2017).

Why do we need feature descriptors?

If we know where the good features are,
how do we match them?

Object instance recognition

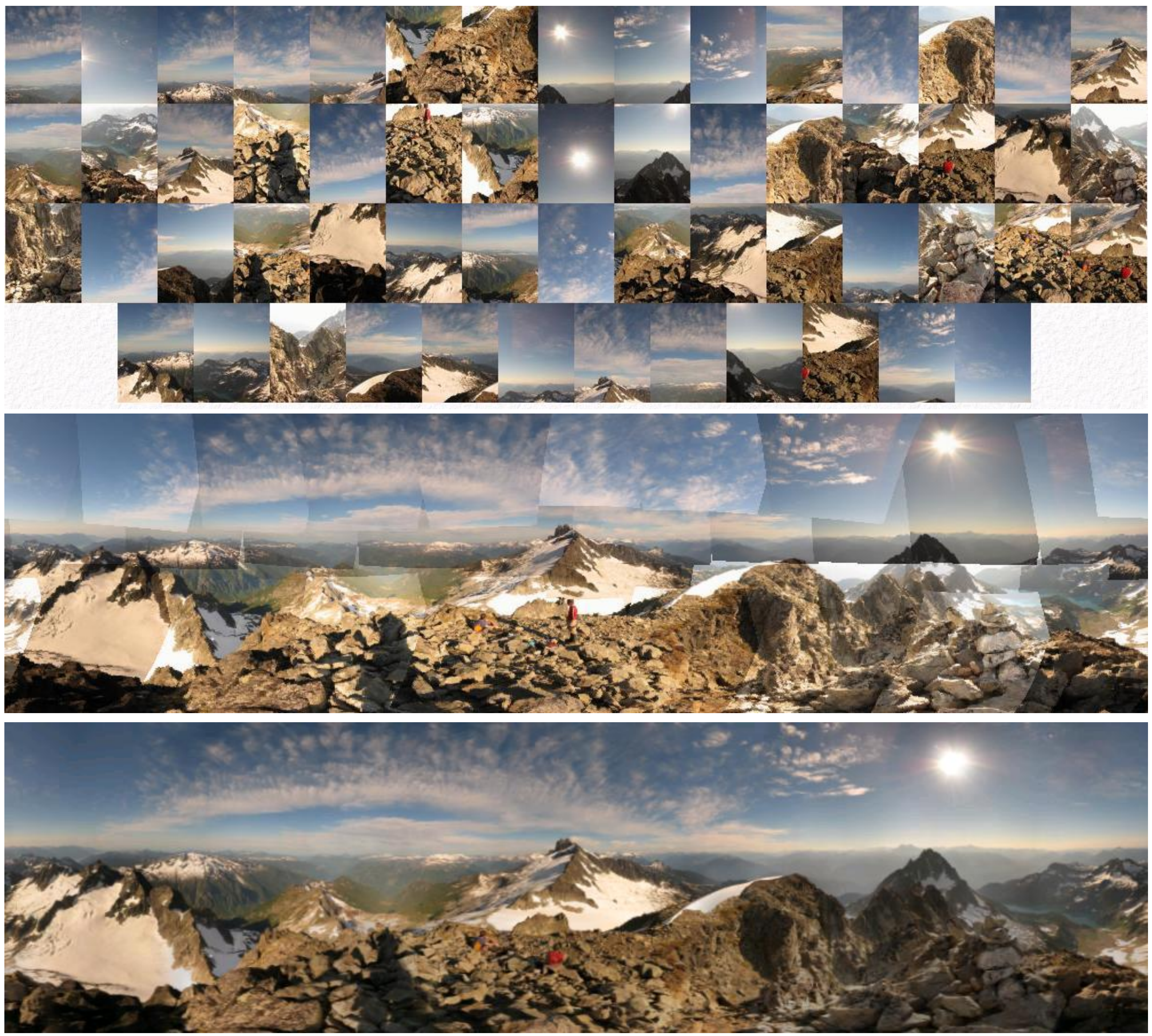
Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Image mosaicing



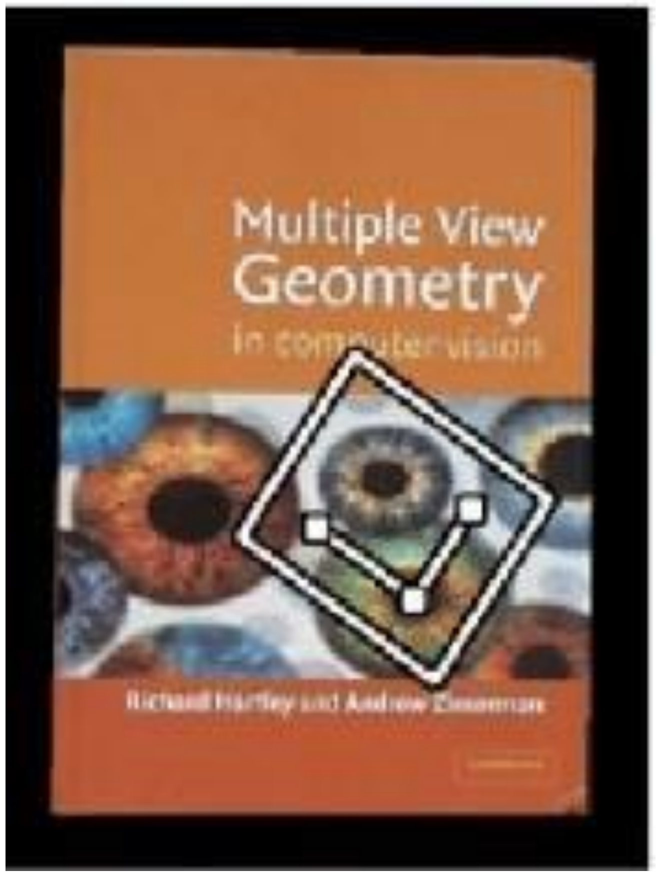
How do we describe an image patch?

Patches with similar content should have similar descriptors.

Designing feature descriptors

Photometric transformations

Geometric transformations

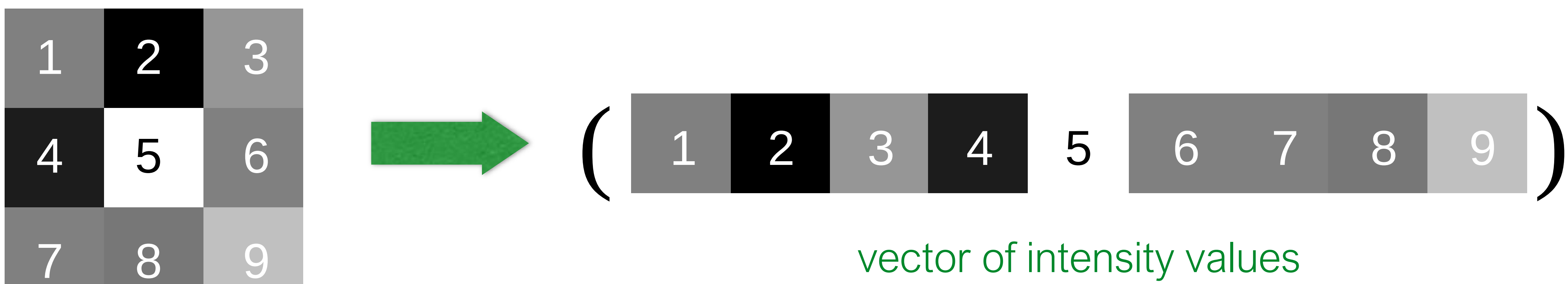


objects will appear at different scales,
translation and rotation

What is the best descriptor for an image feature?

Image patch

Just use the pixel values of the patch



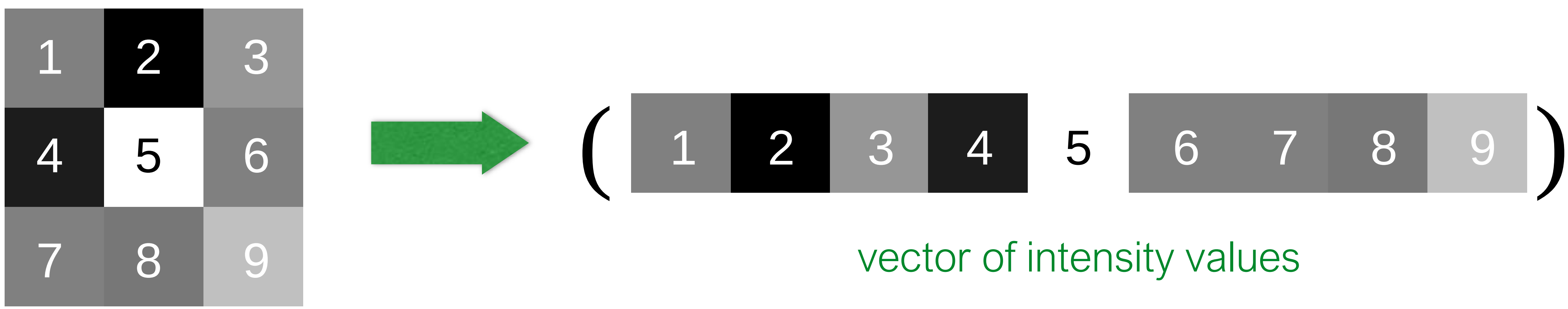
Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

Tiny Images

Just down-sample it!
Simple, fast, robust to small affine transforms.

Image patch

Just use the pixel values of the patch



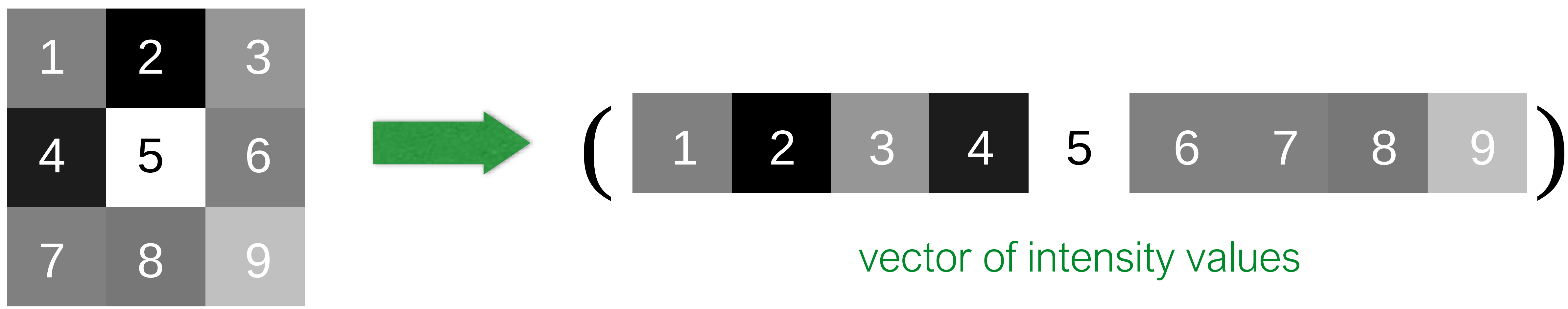
vector of intensity values

Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

What are the problems?

Image patch

Just use the pixel values of the patch



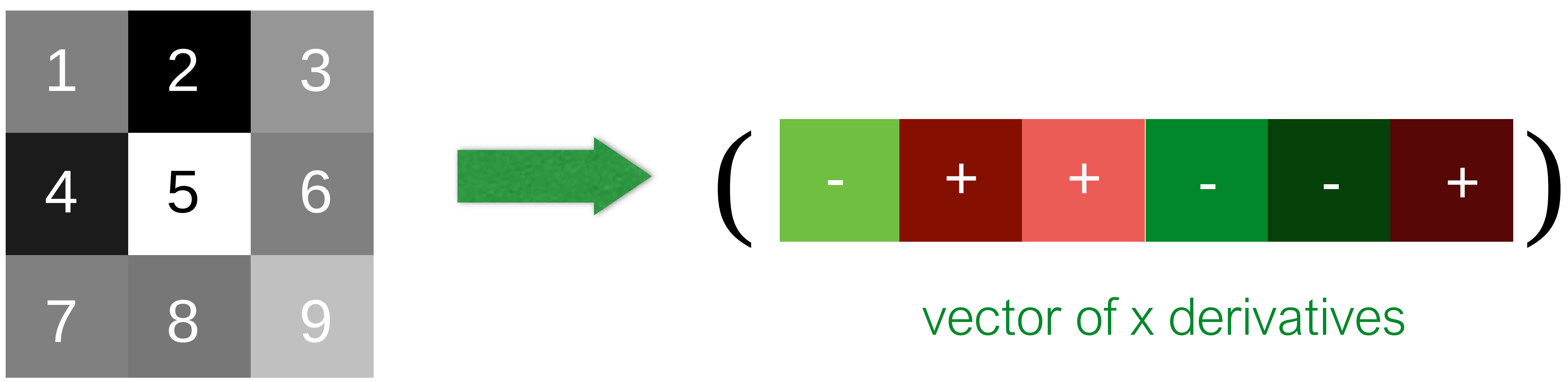
Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

What are the problems?

How can you be less sensitive to absolute intensity values?

Image gradients

Use pixel differences



vector of x derivatives

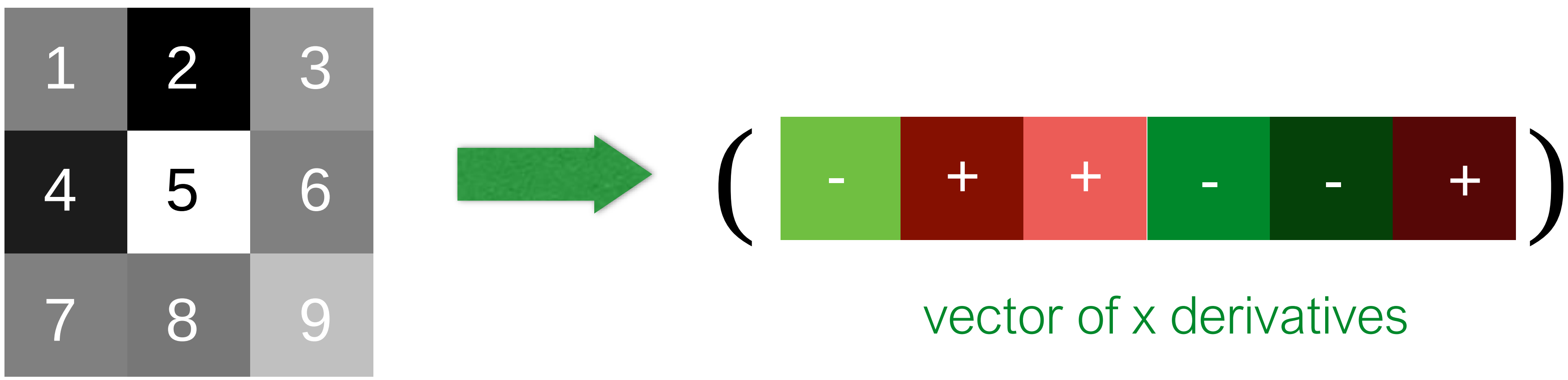
'binary descriptor'

Feature is invariant to absolute intensity values

What are the problems?

Image gradients

Use pixel differences



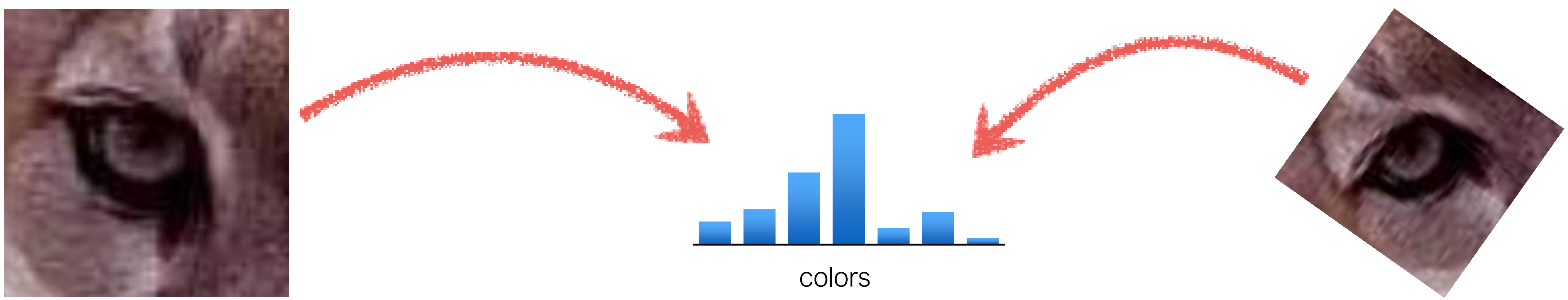
Feature is invariant to absolute intensity values

What are the problems?

How can you be less sensitive to deformations?

Color histogram

Count the colors in the image using a histogram

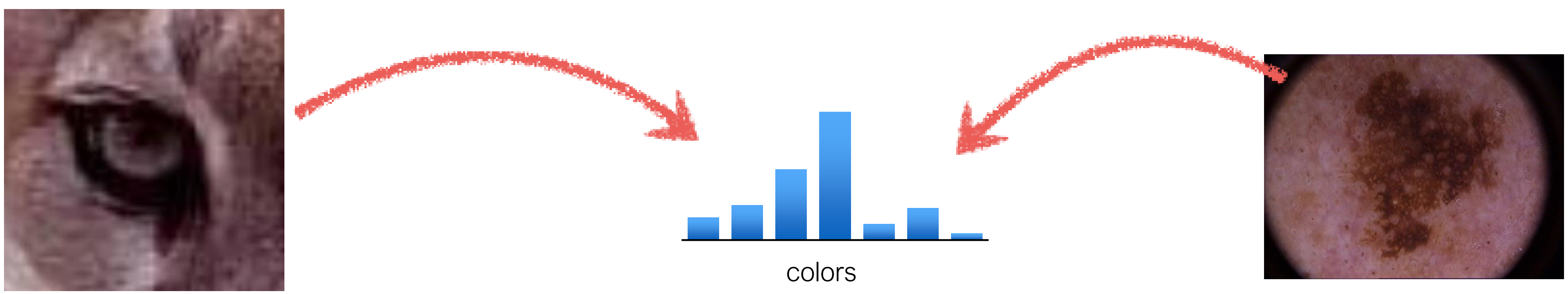


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram

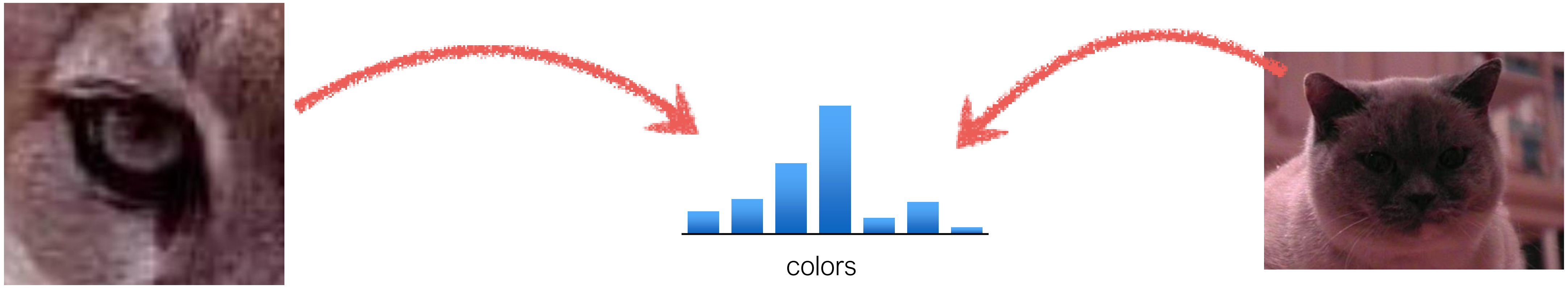


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram



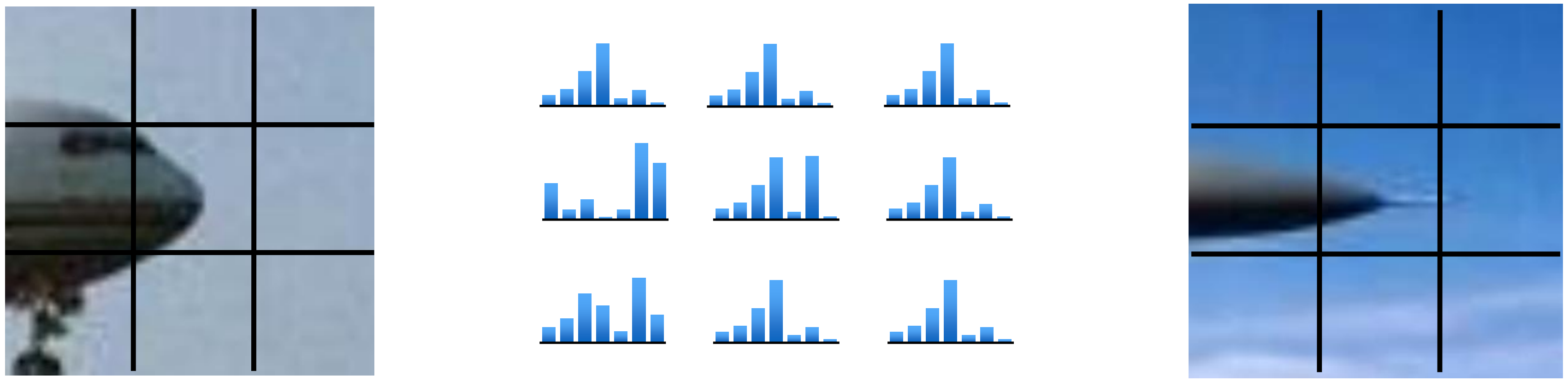
Invariant to changes in scale and rotation

What are the problems?

How can you be more sensitive to spatial layout?

Spatial histograms

Compute histograms over spatial 'cells'



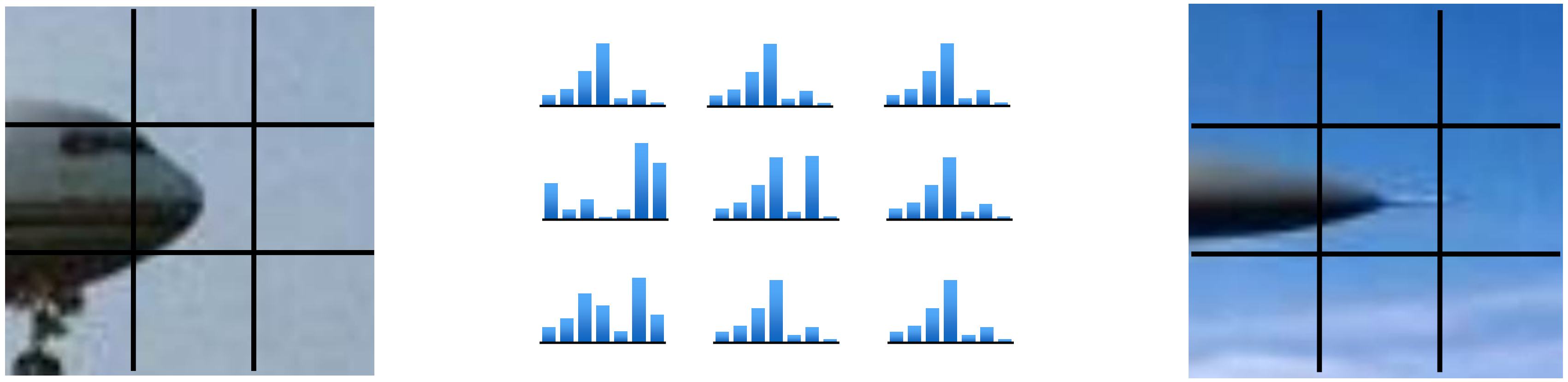
Retains rough spatial layout

Some invariance to deformations

What are the problems?

Spatial histograms

Compute histograms over spatial ‘cells’



Retains rough spatial layout

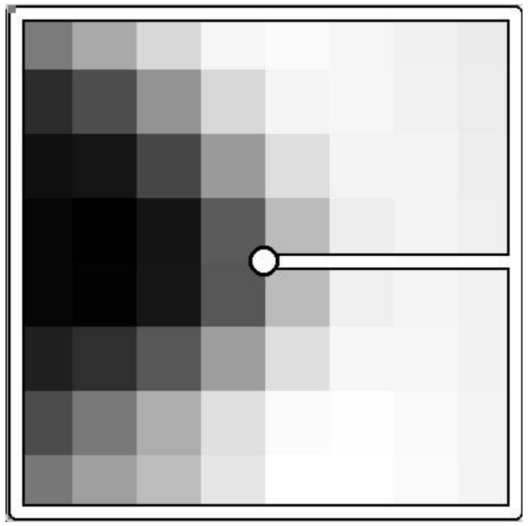
Some invariance to deformations

What are the problems?

How can you be completely invariant to rotation?

Orientation normalization

Use the dominant image gradient direction to normalize the orientation of the patch



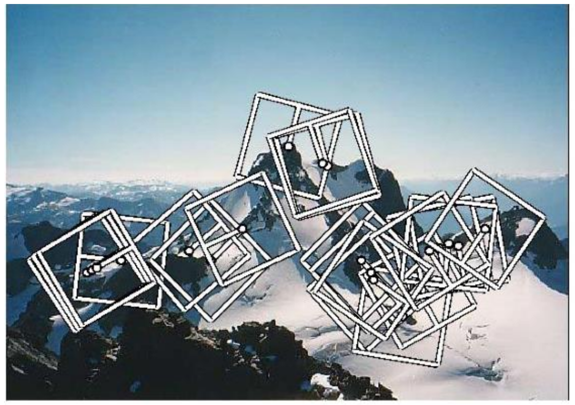
save the orientation angle θ along with (x, y, s)

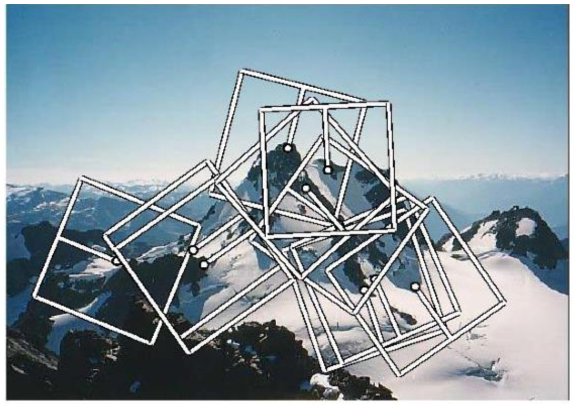
What are the problems?

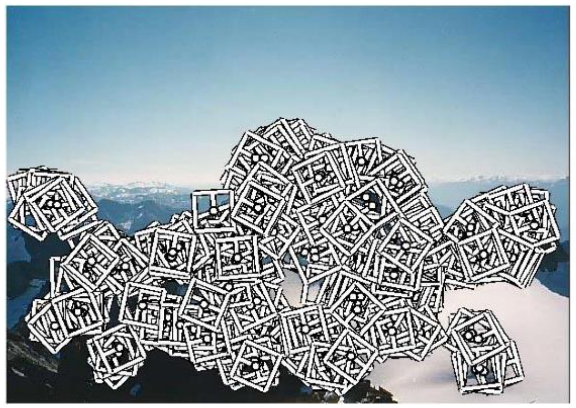
MOPS descriptor

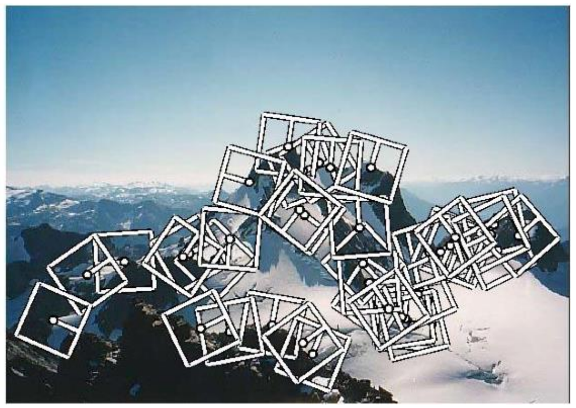
Multi-Scale Oriented Patches (MOPS)

Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517







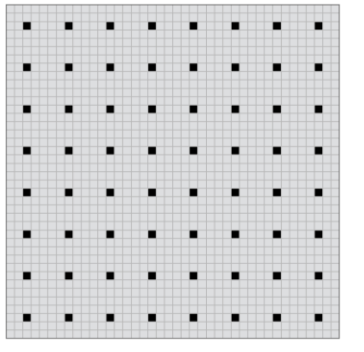


Multi-Scale Oriented Patches (MOPS)

Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

Get 40 x 40 image patch, subsample every 5th pixel
(*what's the purpose of this step?*)



Subtract the mean, divide by standard deviation
(*what's the purpose of this step?*)

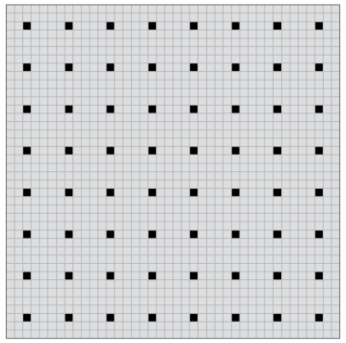
Haar Wavelet Transform
(*what's the purpose of this step?*)

Multi-Scale Oriented Patches (MOPS)

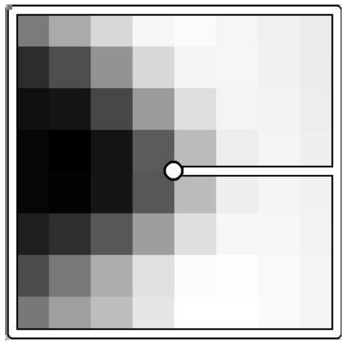
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder.
International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



Subtract the mean, divide by standard deviation
(*what's the purpose of this step?*)



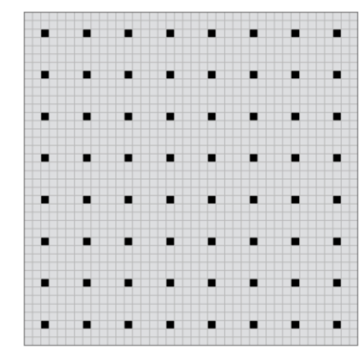
Haar Wavelet Transform
(*what's the purpose of this step?*)

Multi-Scale Oriented Patches (MOPS)

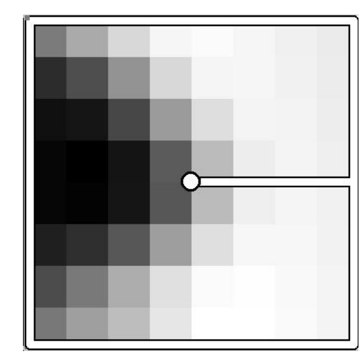
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder.
International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

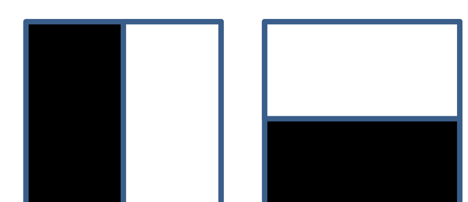
Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



Subtract the mean, divide by standard deviation
(removes bias and gain)



Haar Wavelet Transform
(*what's the purpose of this step?*)

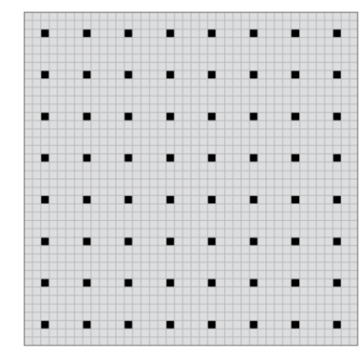


Multi-Scale Oriented Patches (MOPS)

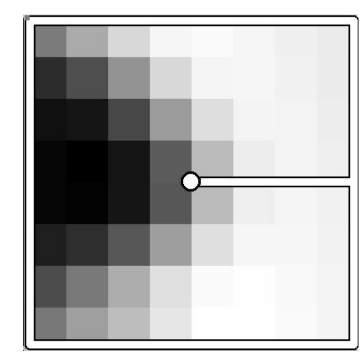
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder.
International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

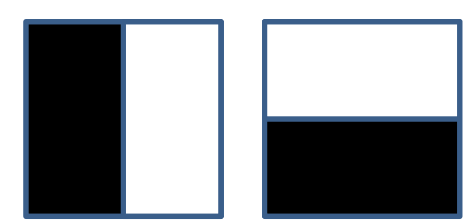
Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



Subtract the mean, divide by standard deviation
(removes bias and gain)



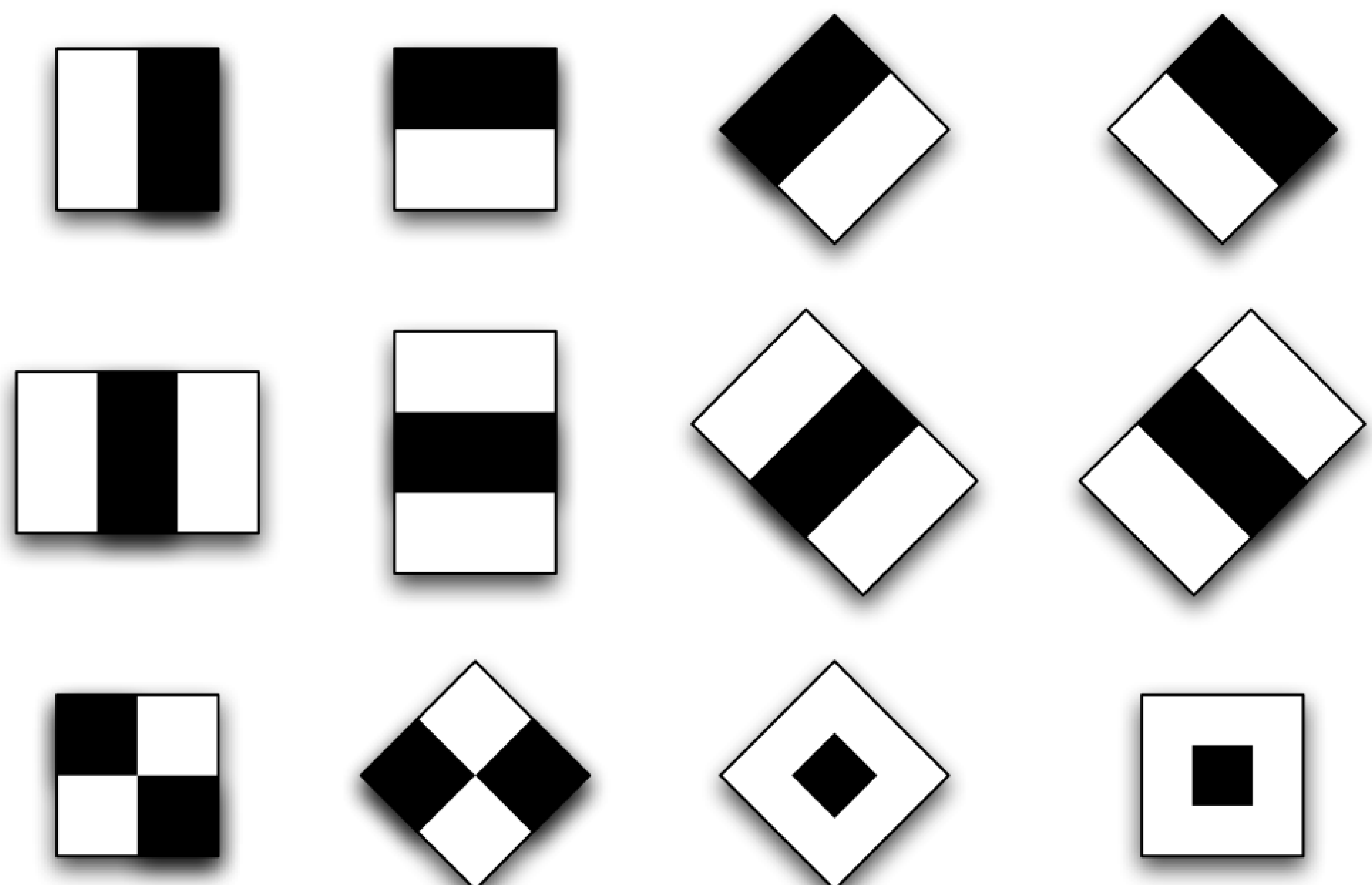
Haar Wavelet Transform
(low frequency projection)



Haar Wavelets

(actually, Haar-like features)

Use responses of a bank of filters as a descriptor



We will see later in class how to compute Haar wavelet responses **efficiently** (in constant time) with integral images

GIST descriptor

GIST

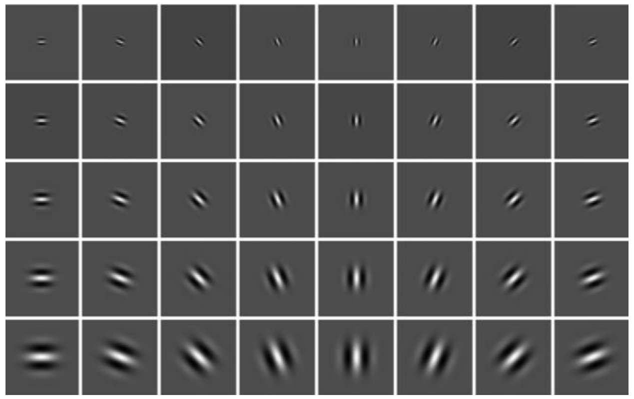
1. Compute filter responses (filter bank of Gabor filters)

2. Divide image patch into 4 x 4 cells

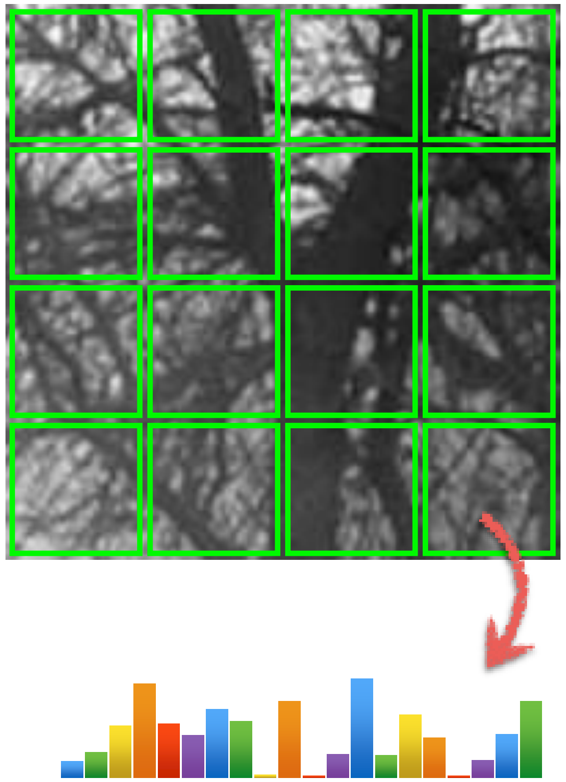
3. Compute filter response averages for each cell

4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

Filter bank



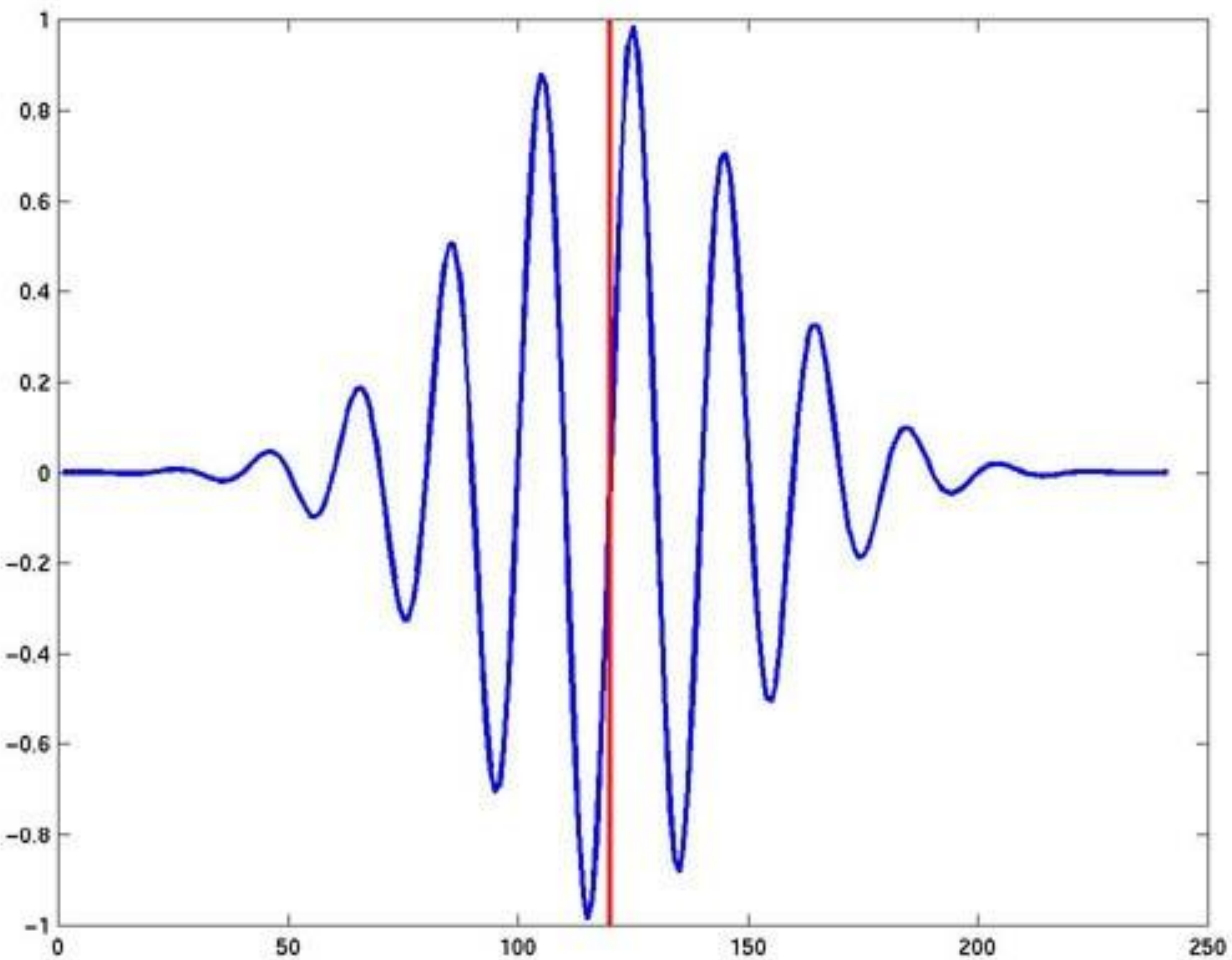
4 x 4 cell

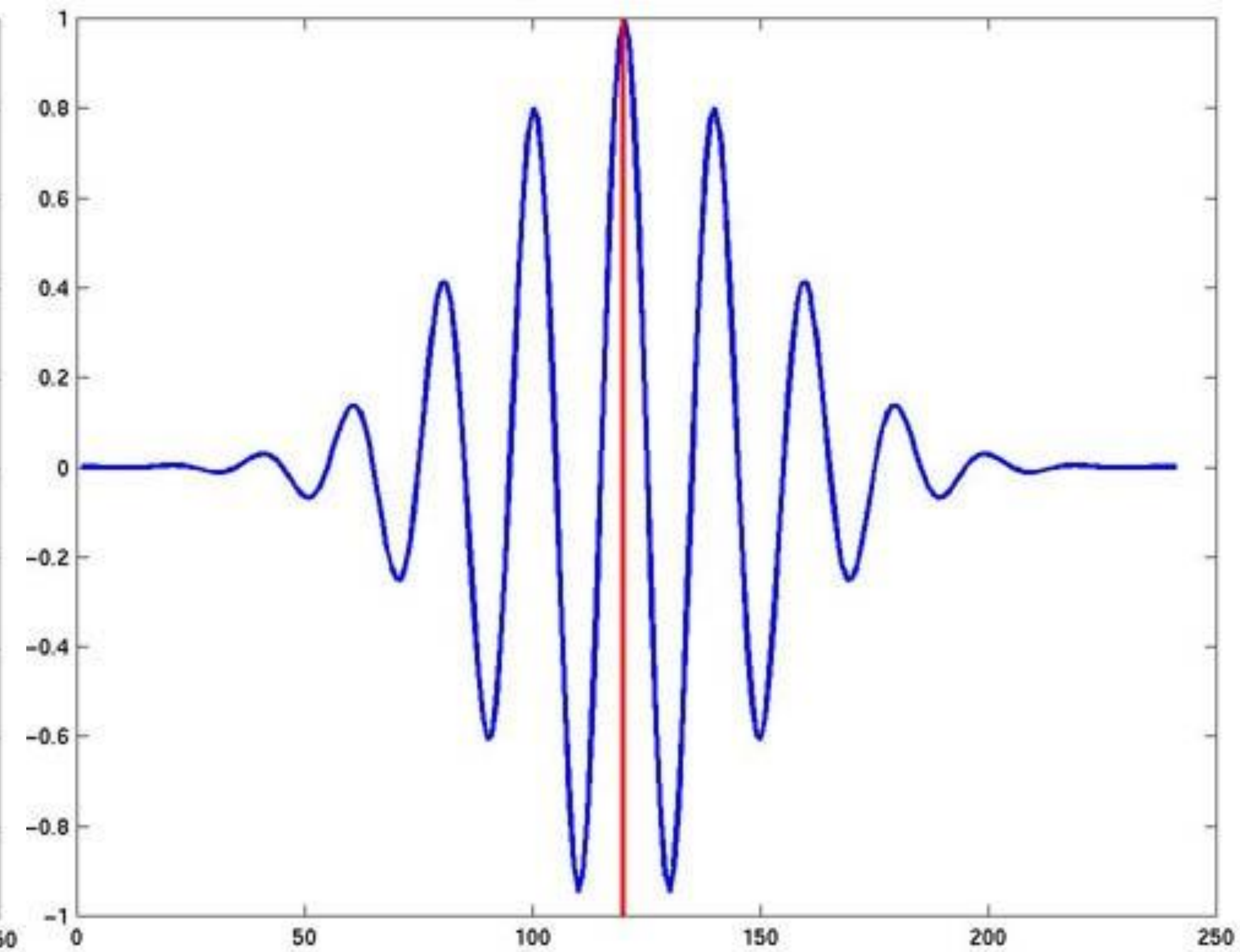


averaged filter responses

Gabor Filters

(1D examples)



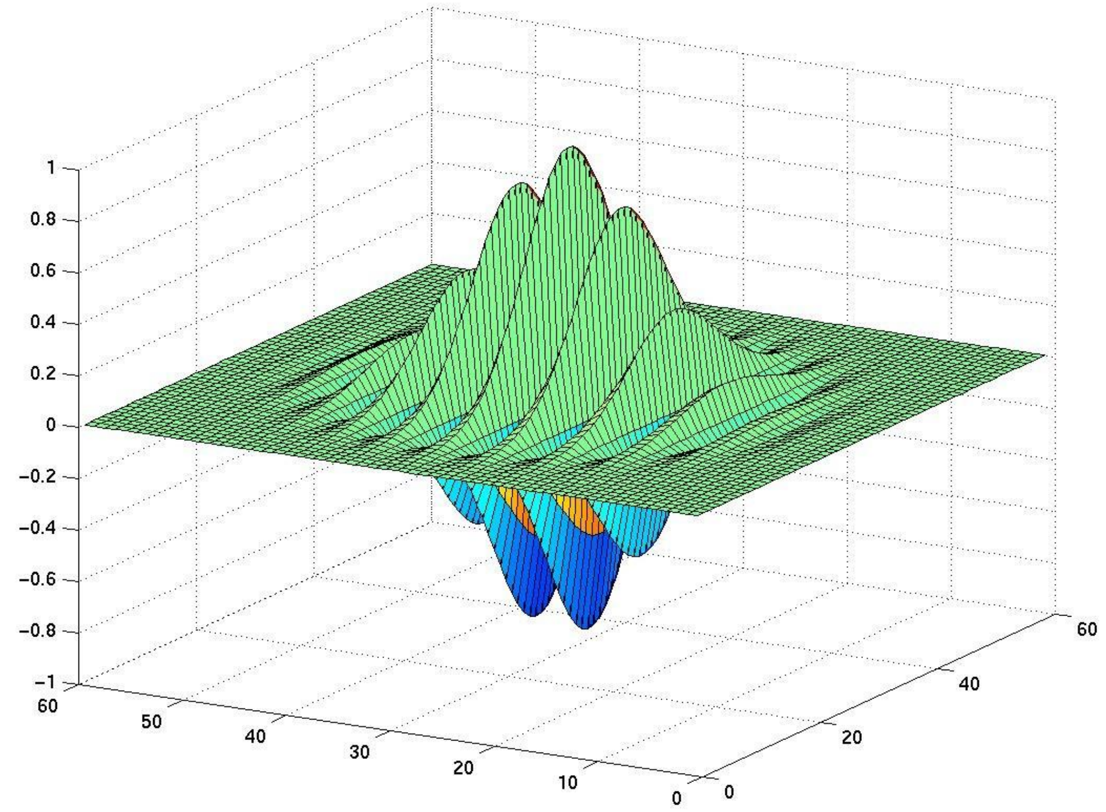


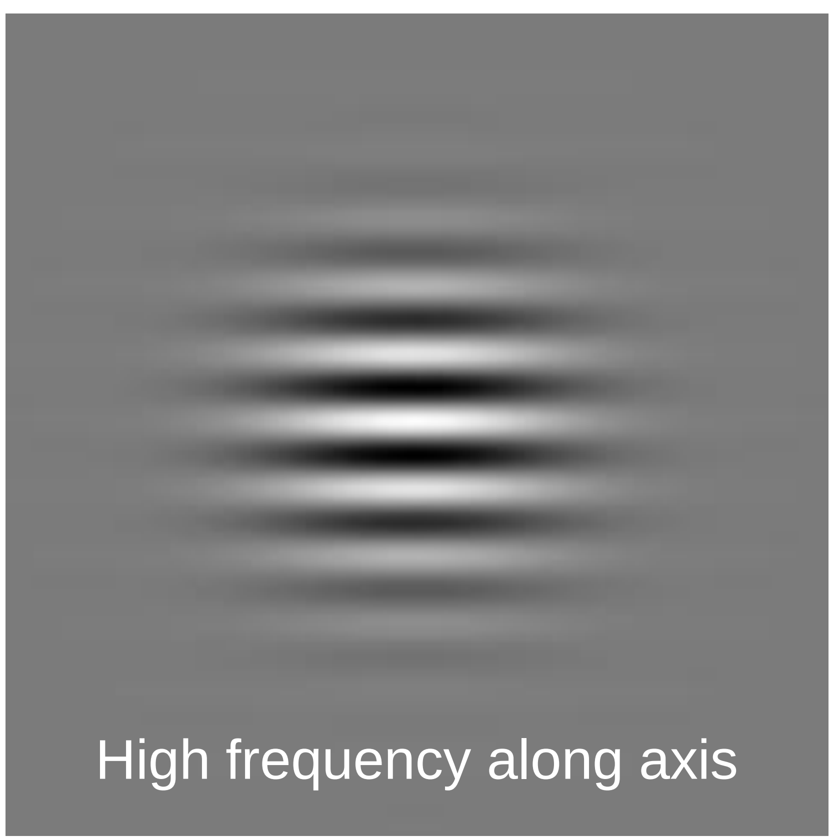
$$e^{-\frac{x^2}{2\sigma^2}} \sin\left(2\pi\omega x\right)$$

$$e^{-\frac{x^2}{2\sigma^2}} \cos\left(2\pi\omega x\right)$$

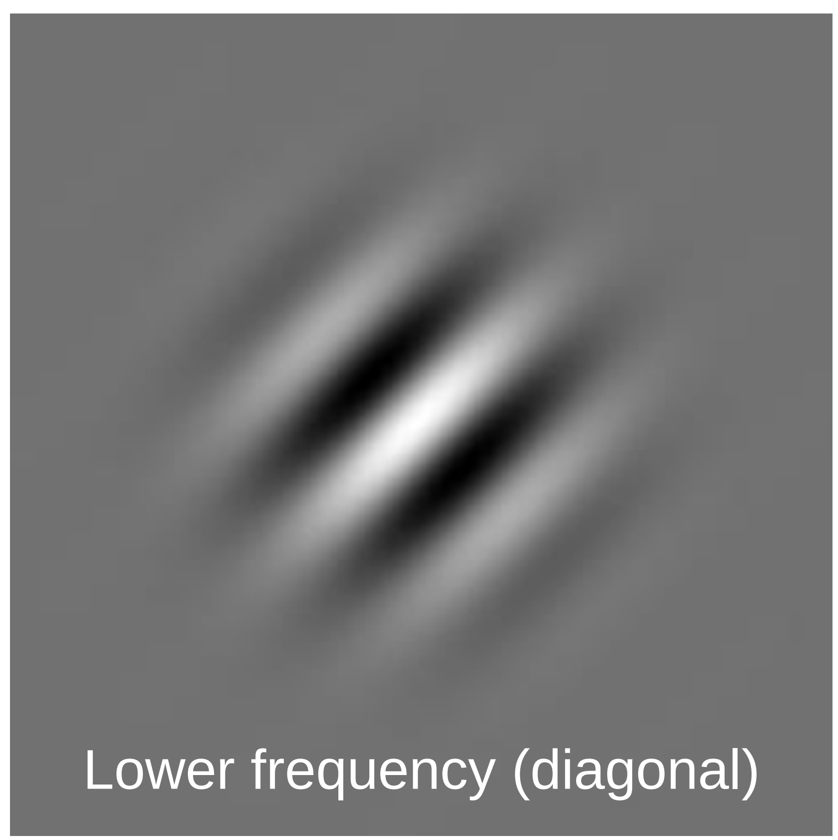
2D Gabor Filters

$$e^{-\frac{x^2+y^2}{2\sigma^2}} \cos\left(2\pi\left(k_x x + k_y y\right)\right)$$

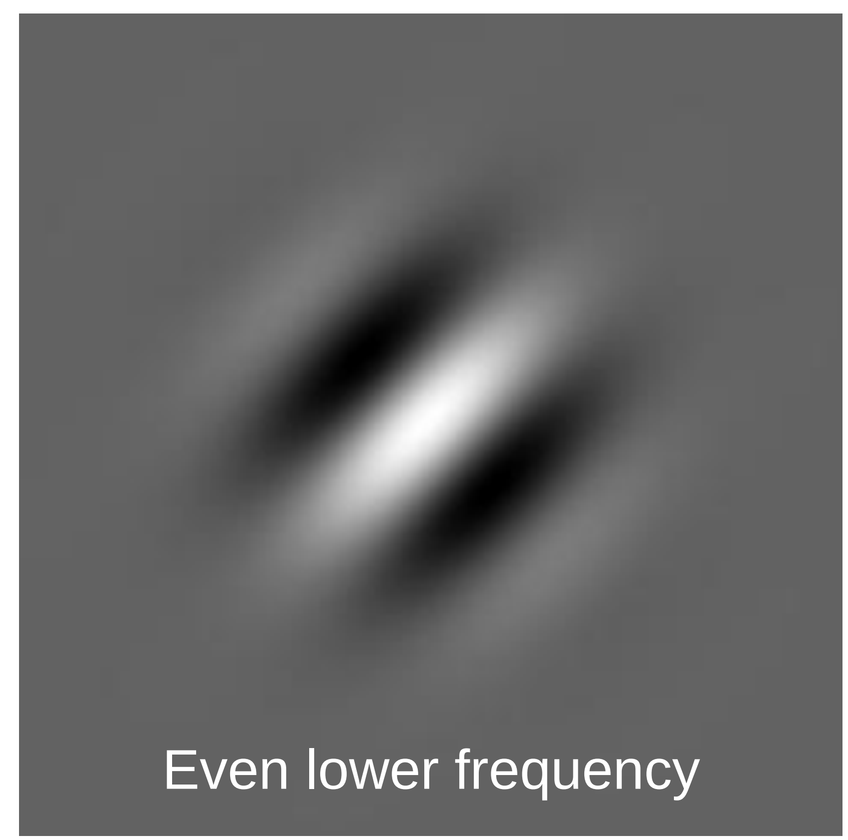




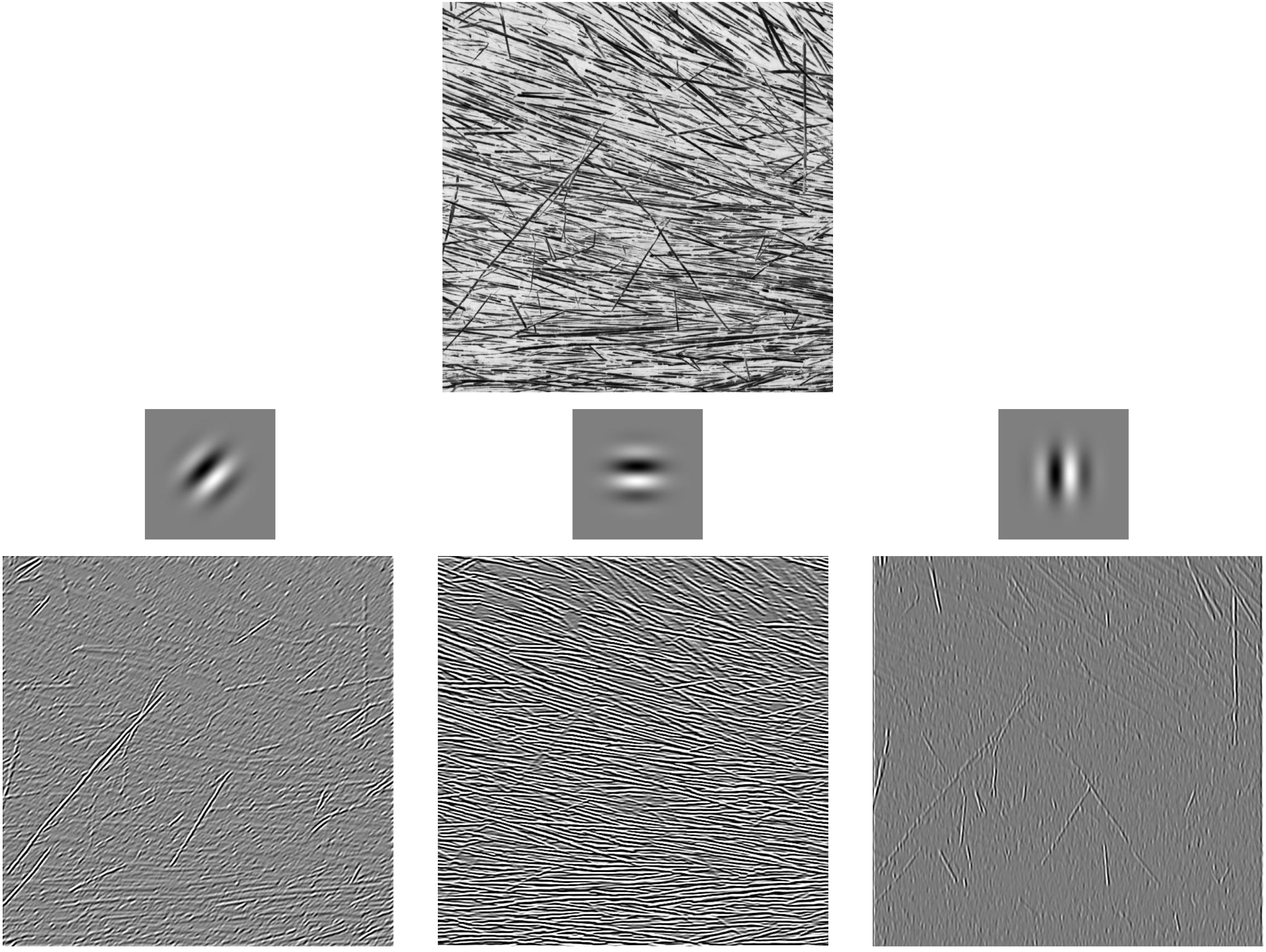
High frequency along axis



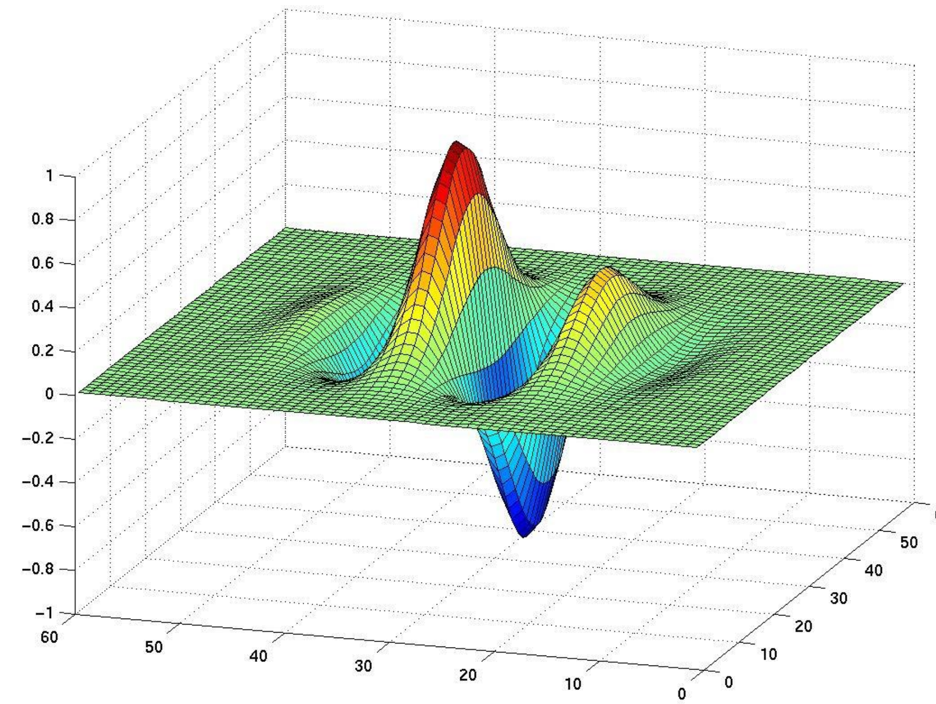
Lower frequency (diagonal)



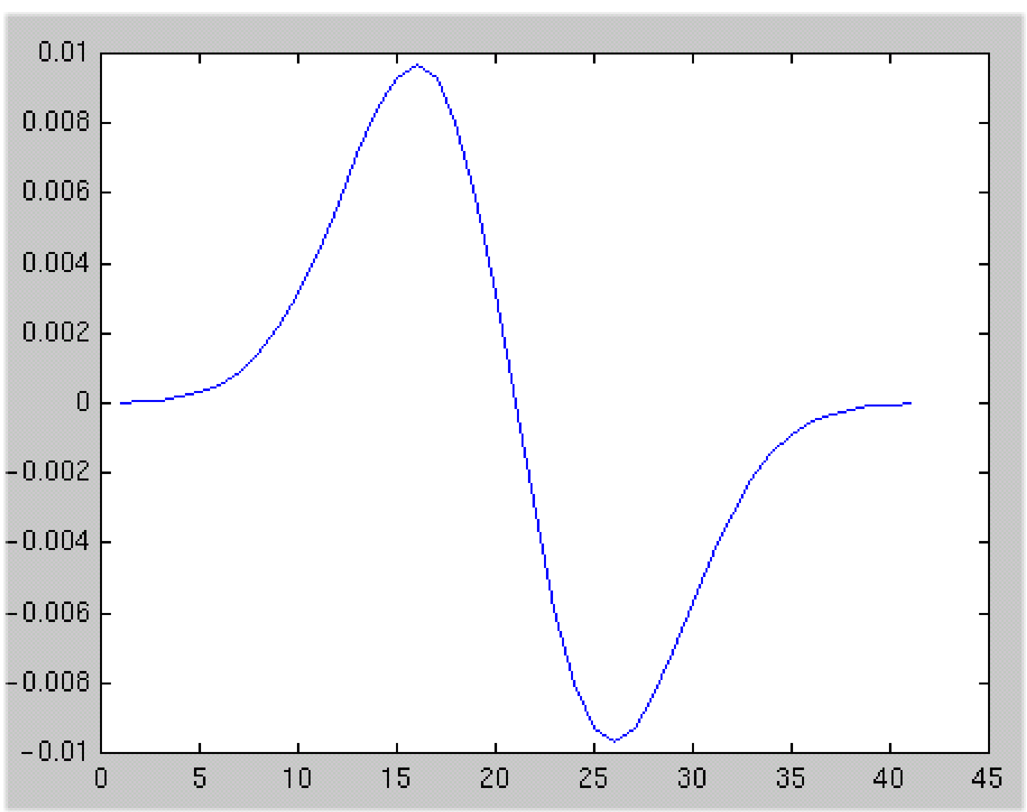
Even lower frequency



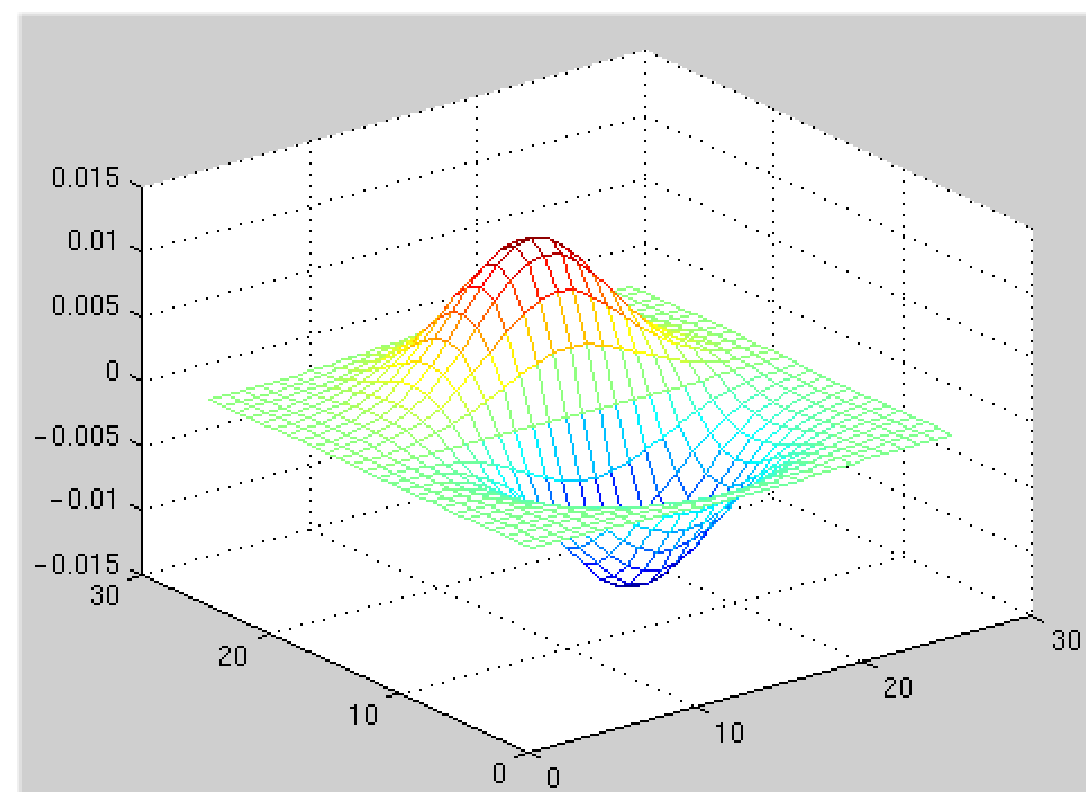
Odd
Gabor
filter

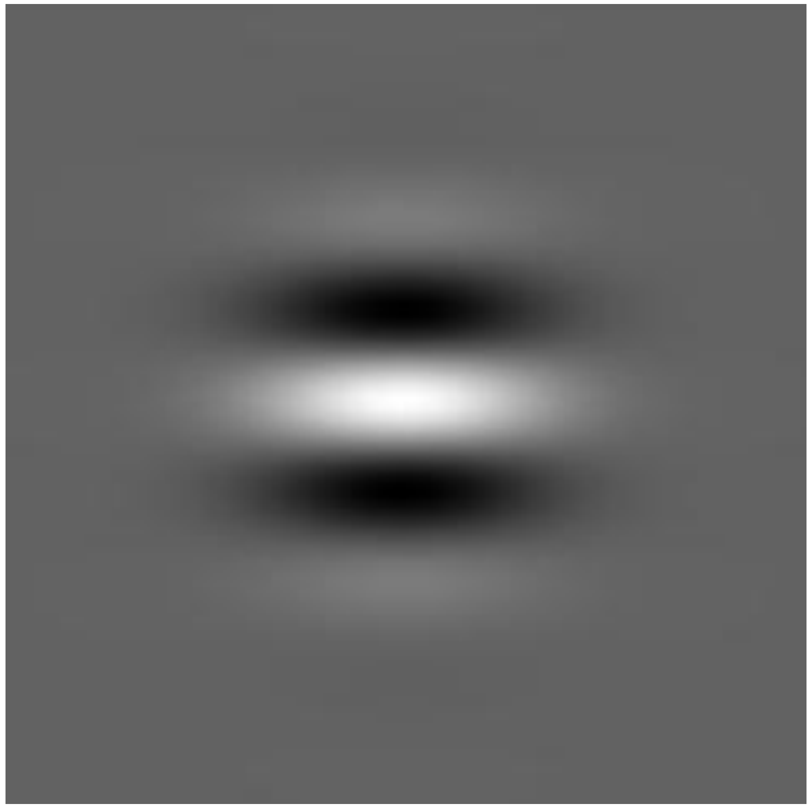


… looks a lot like…

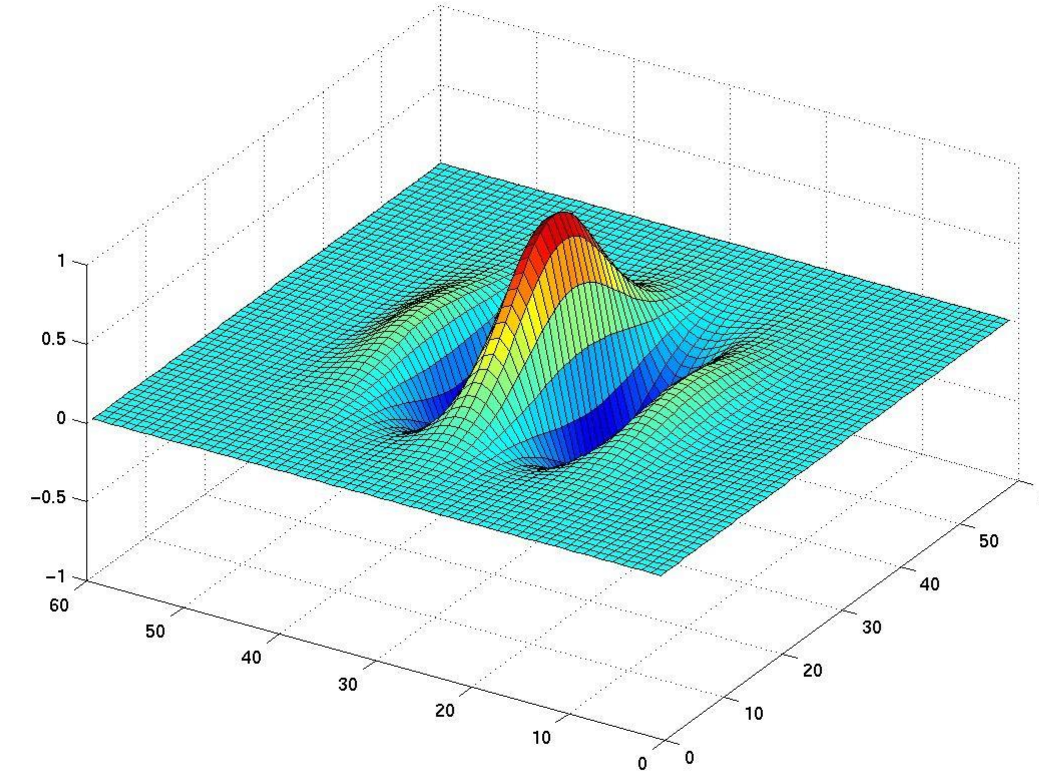


Gaussian
Derivative

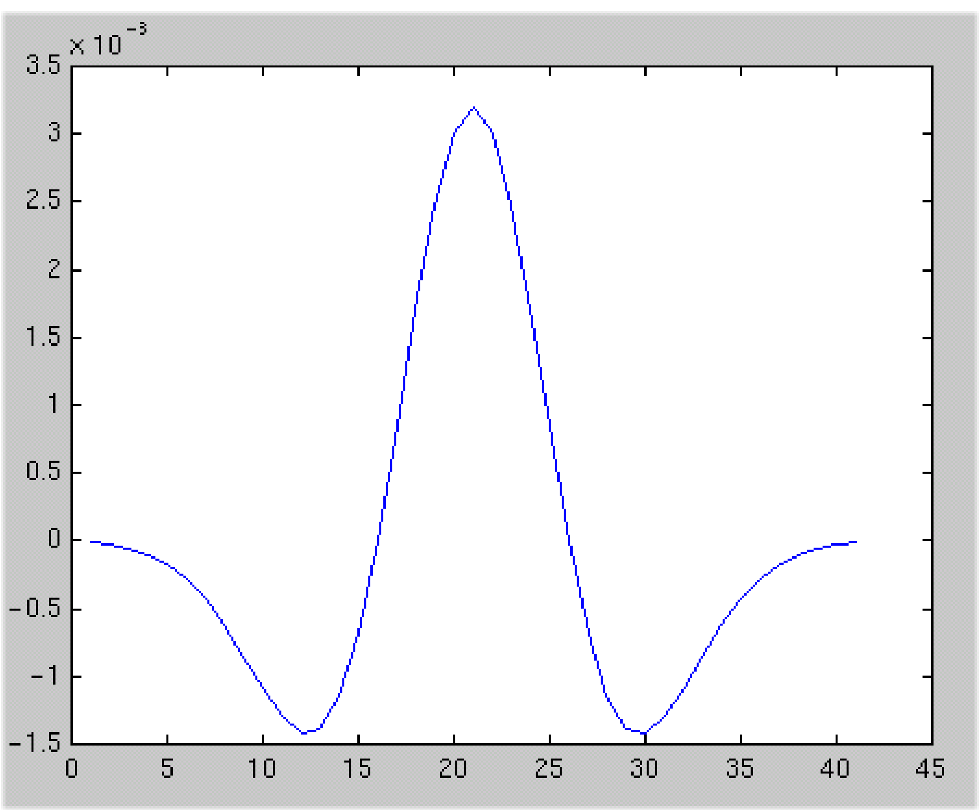




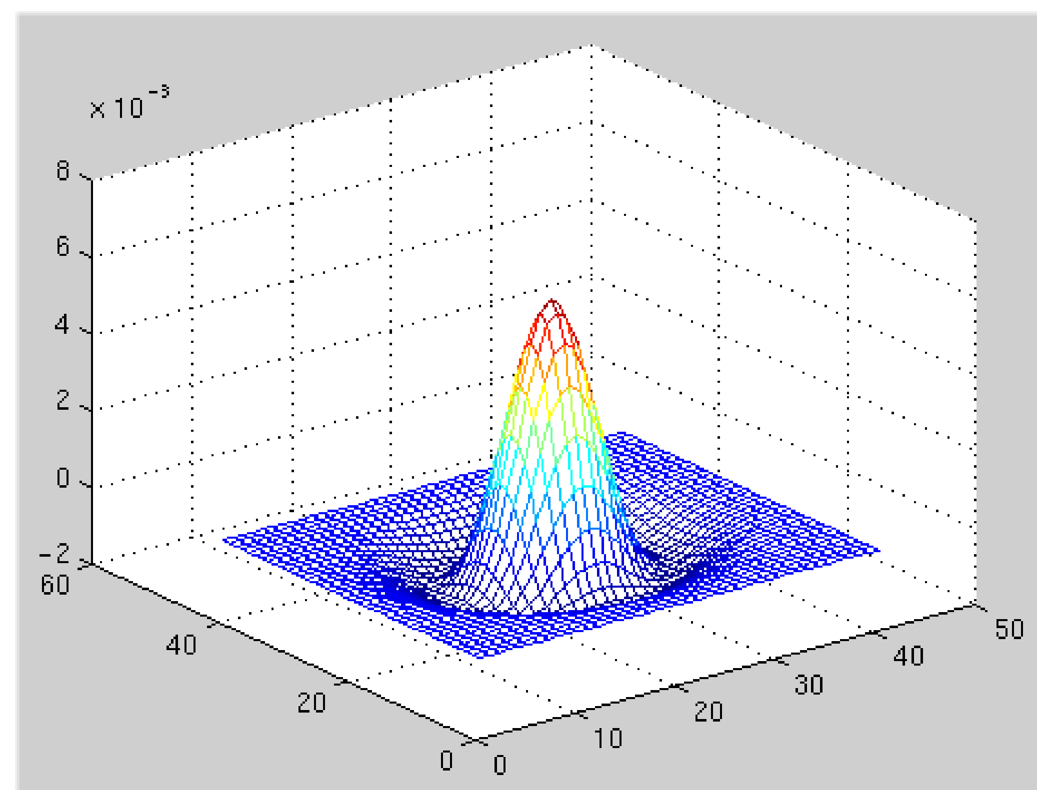
Even Gabor filter



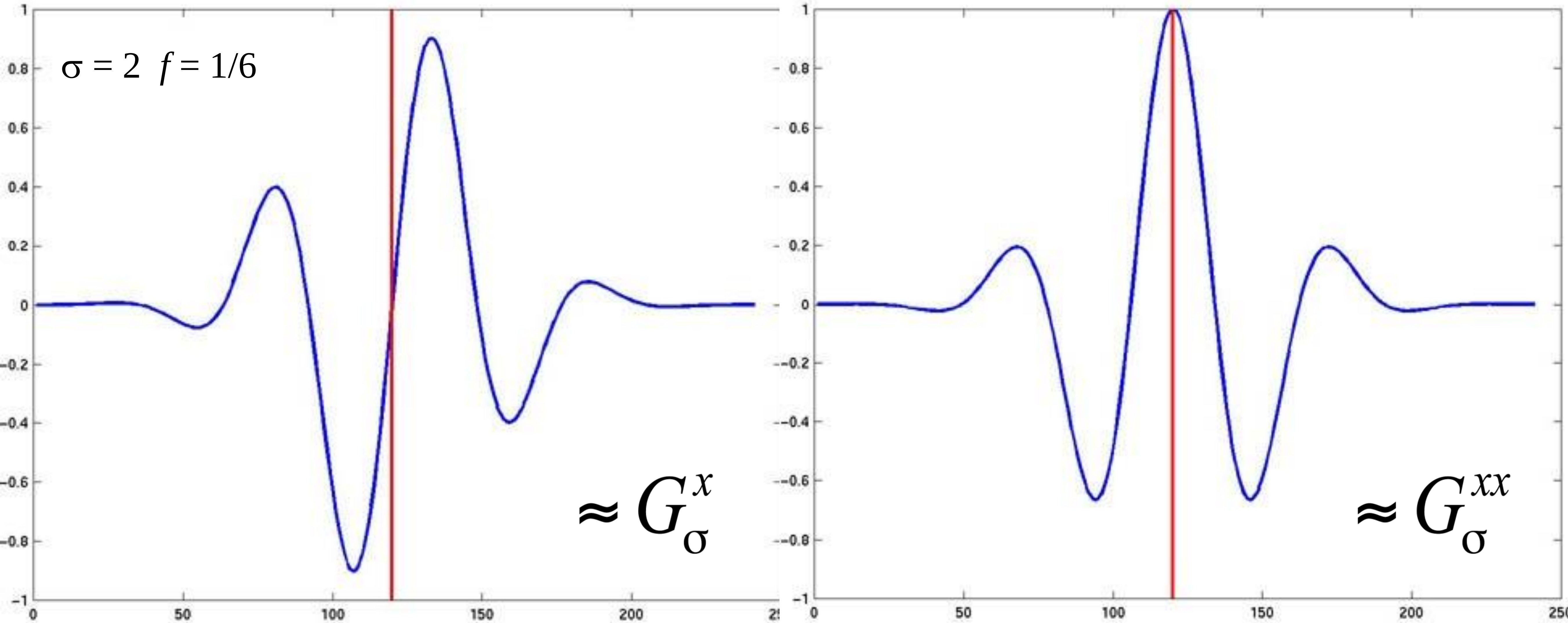
… looks a lot like…



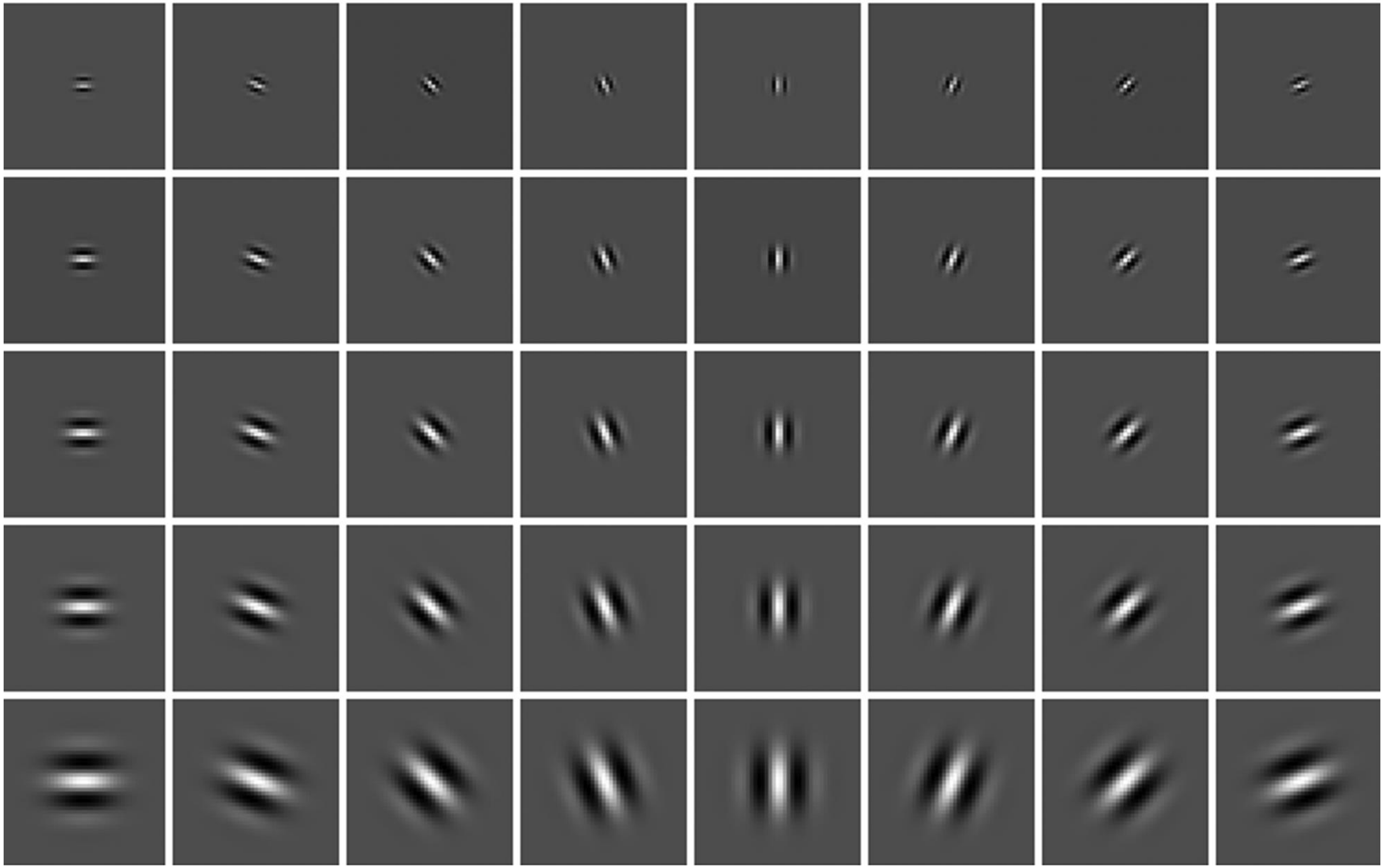
Laplacian



If scale small compared to inverse frequency, the Gabor filters become derivative operators



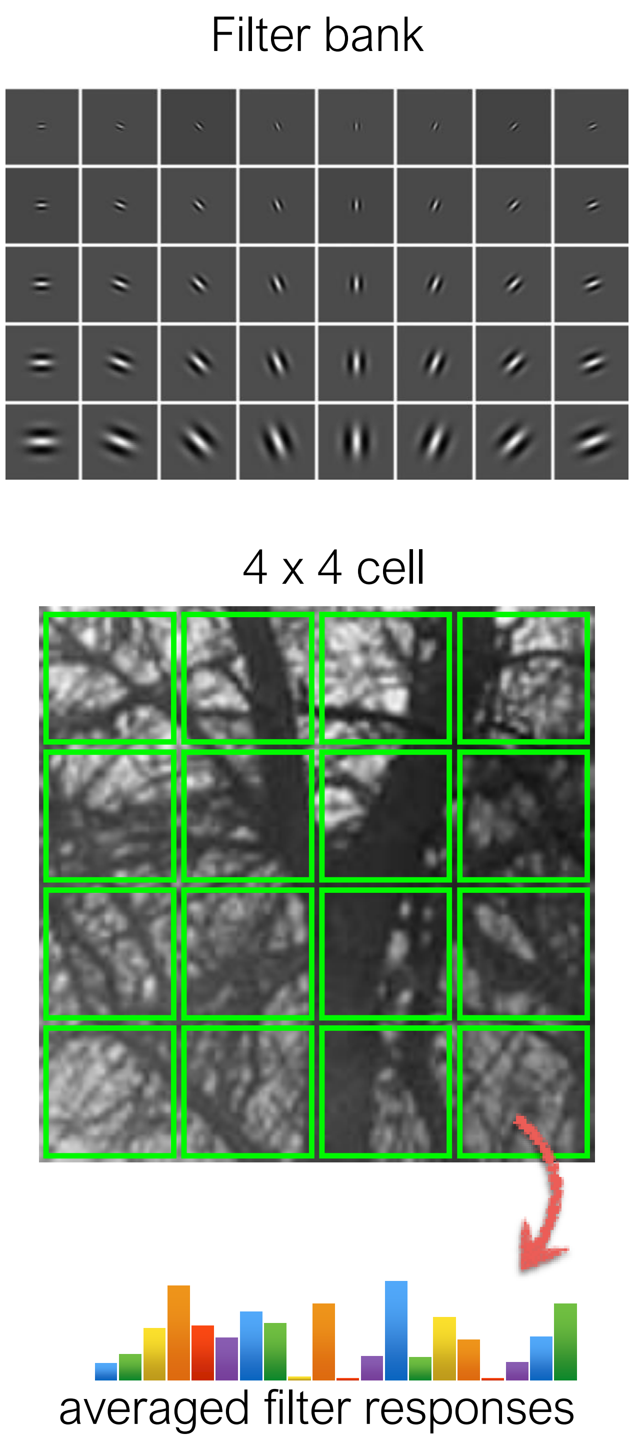
Directional edge detectors



GIST

1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

What is the GIST descriptor encoding?



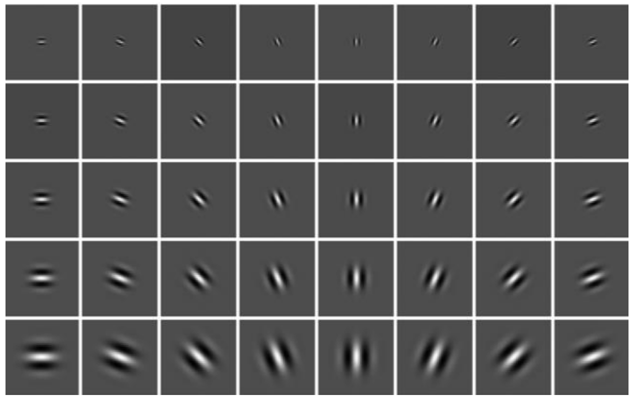
GIST

1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

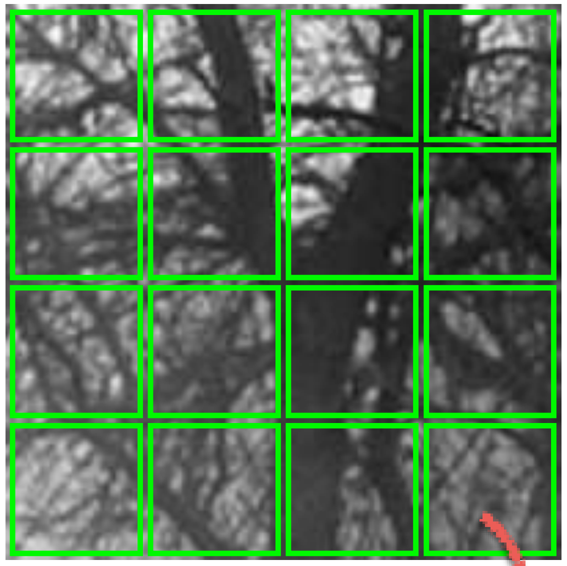
What is the GIST descriptor encoding?

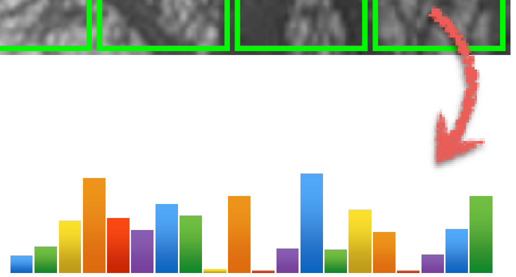
Rough spatial distribution of image gradients

Filter bank



4 x 4 cell





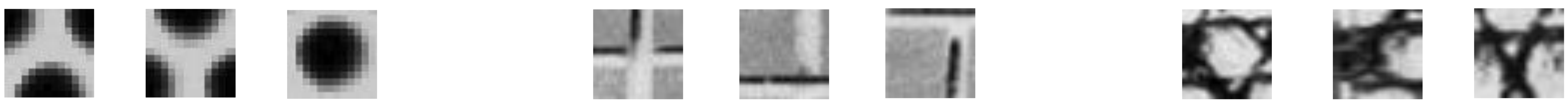
averaged filter responses

Histogram of Textons descriptor

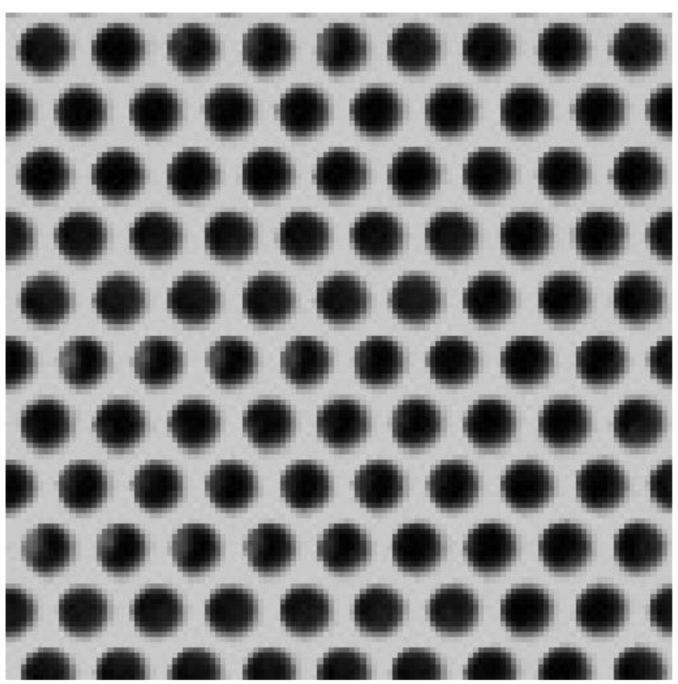
Textons

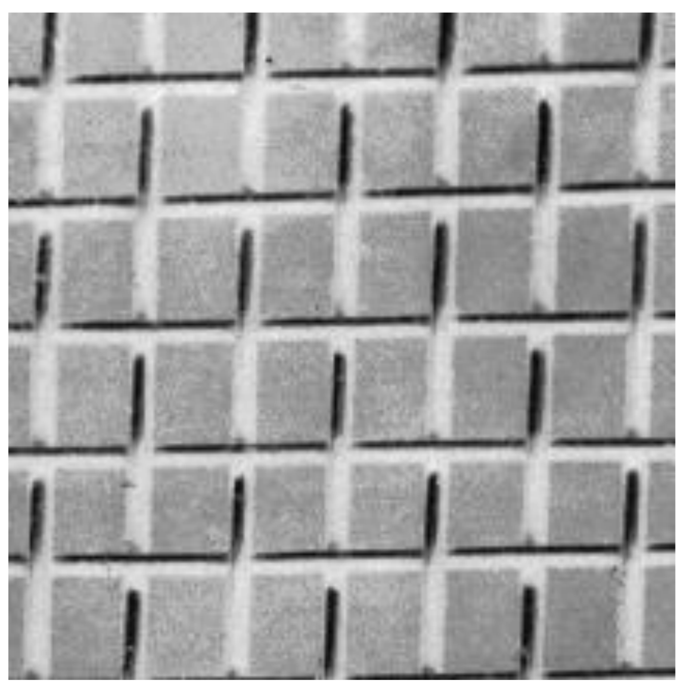
Julesz. Textons, the elements of texture perception, and their interactions. Nature 1981

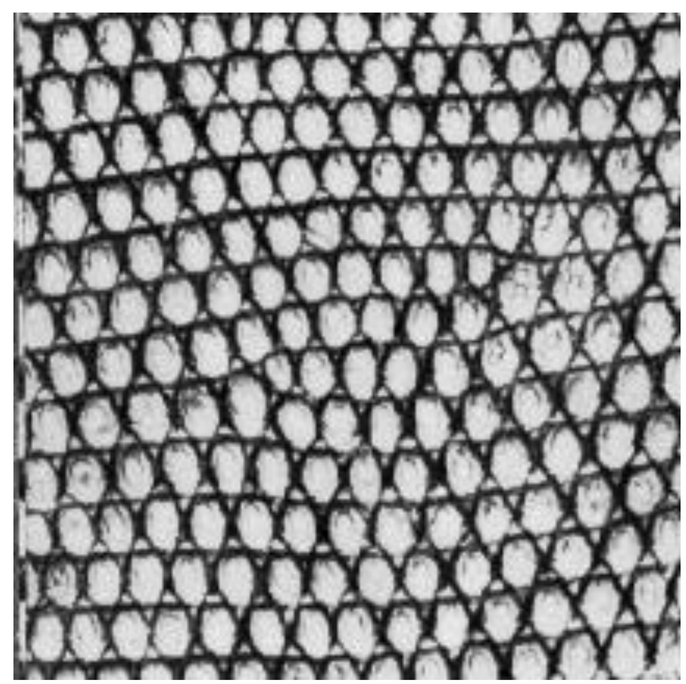
Texture is characterized by the repetition of basic elements or ***textons***



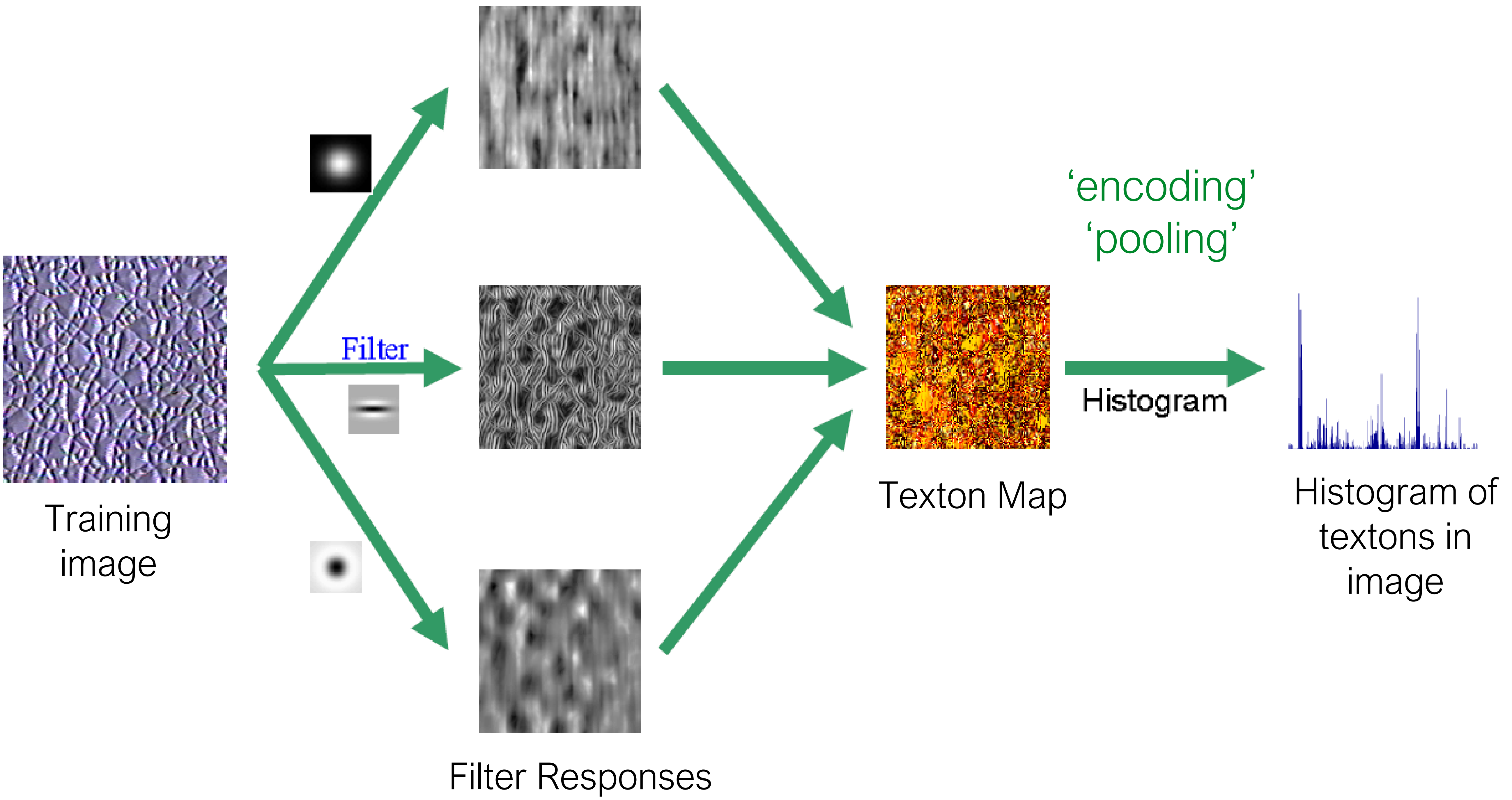
For stochastic textures, it is the identity of the ***textons***, not their spatial arrangement, that matters





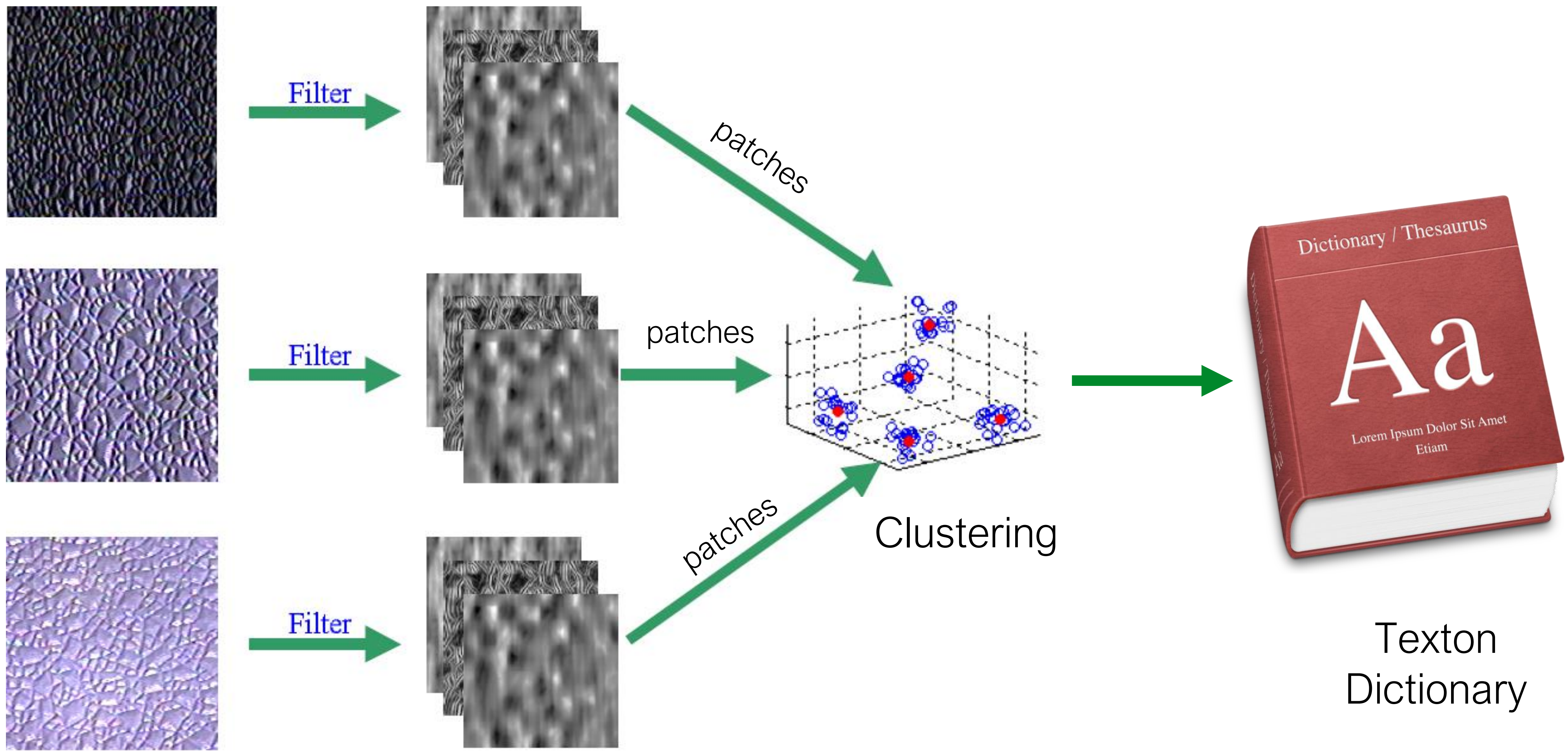


Histogram of Textons descriptor



Malik, Belongie, Shi, Leung. Textons, Contours and Regions: Cue Integration in Image Segmentation. ICCV 1999.

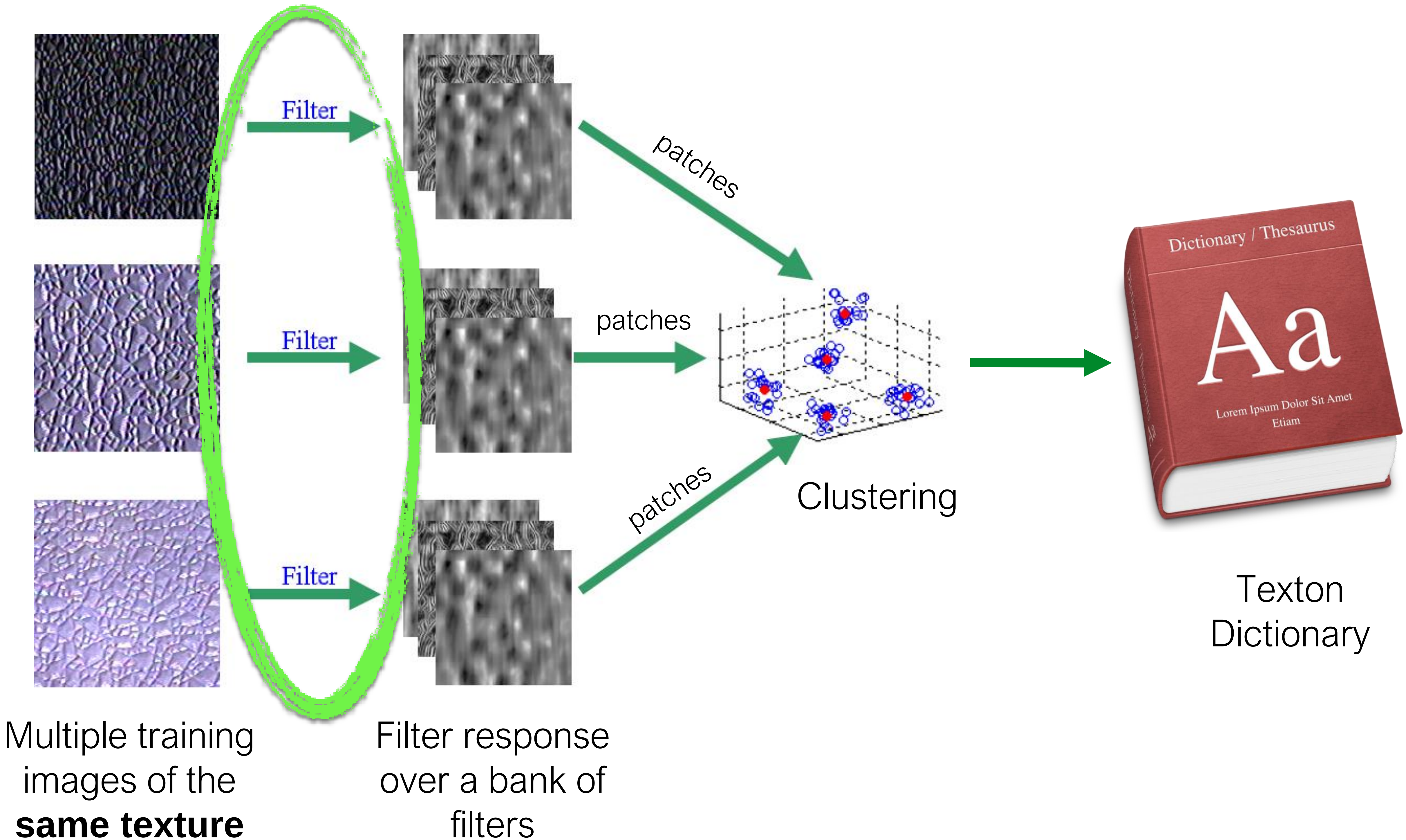
Learning Textons from data



Multiple training images of the **same texture**

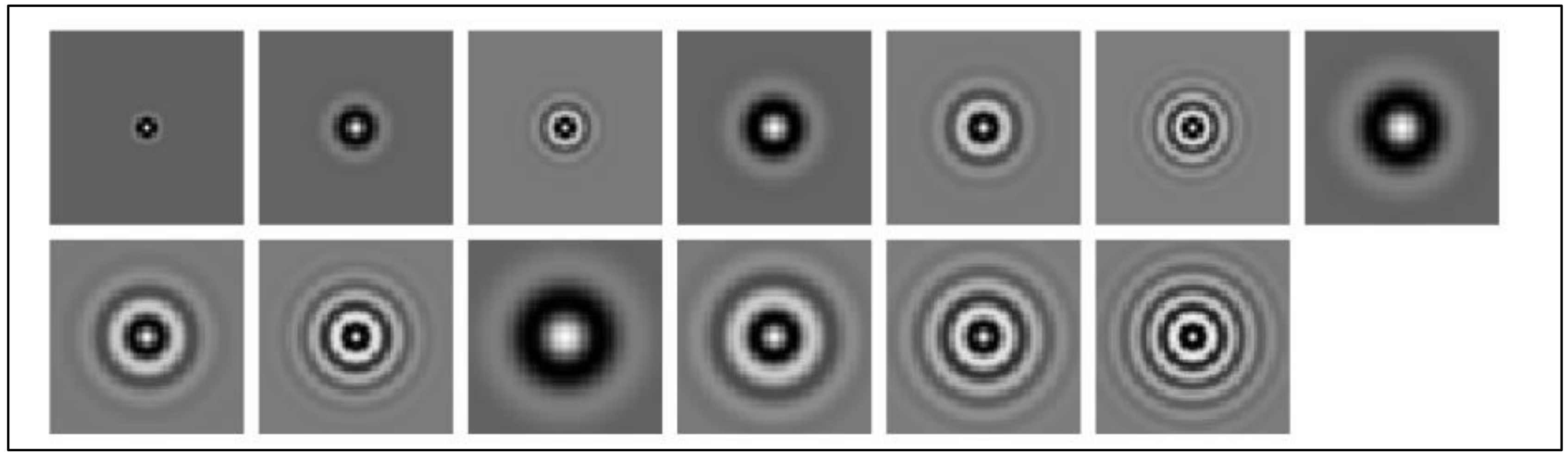
Filter response over a bank of filters

Learning Textons from data



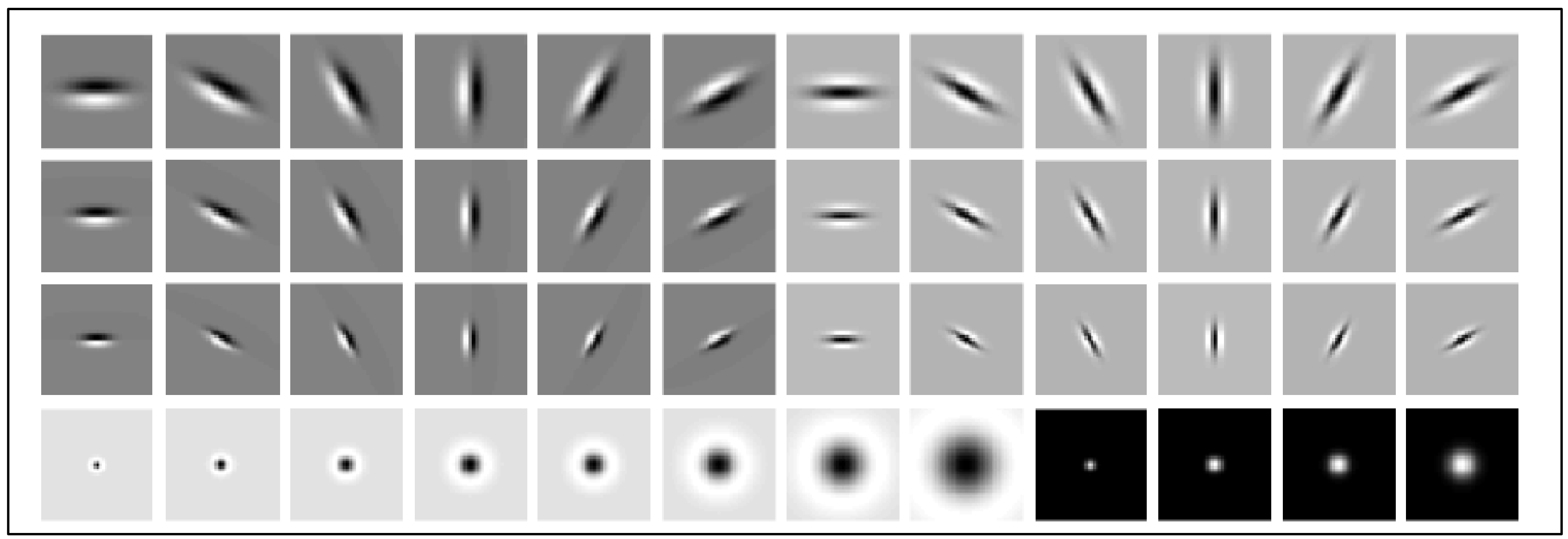
Example of Filter Banks

Isotropic Gabor

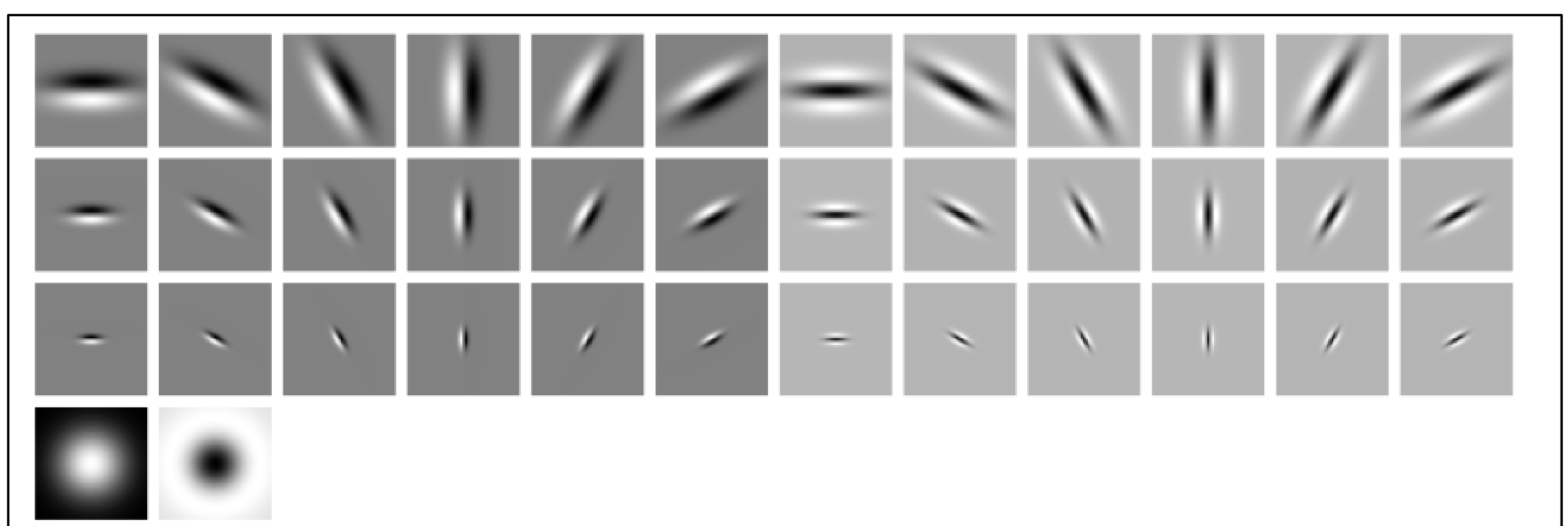


'S'

Gaussian derivatives at different scales and orientations

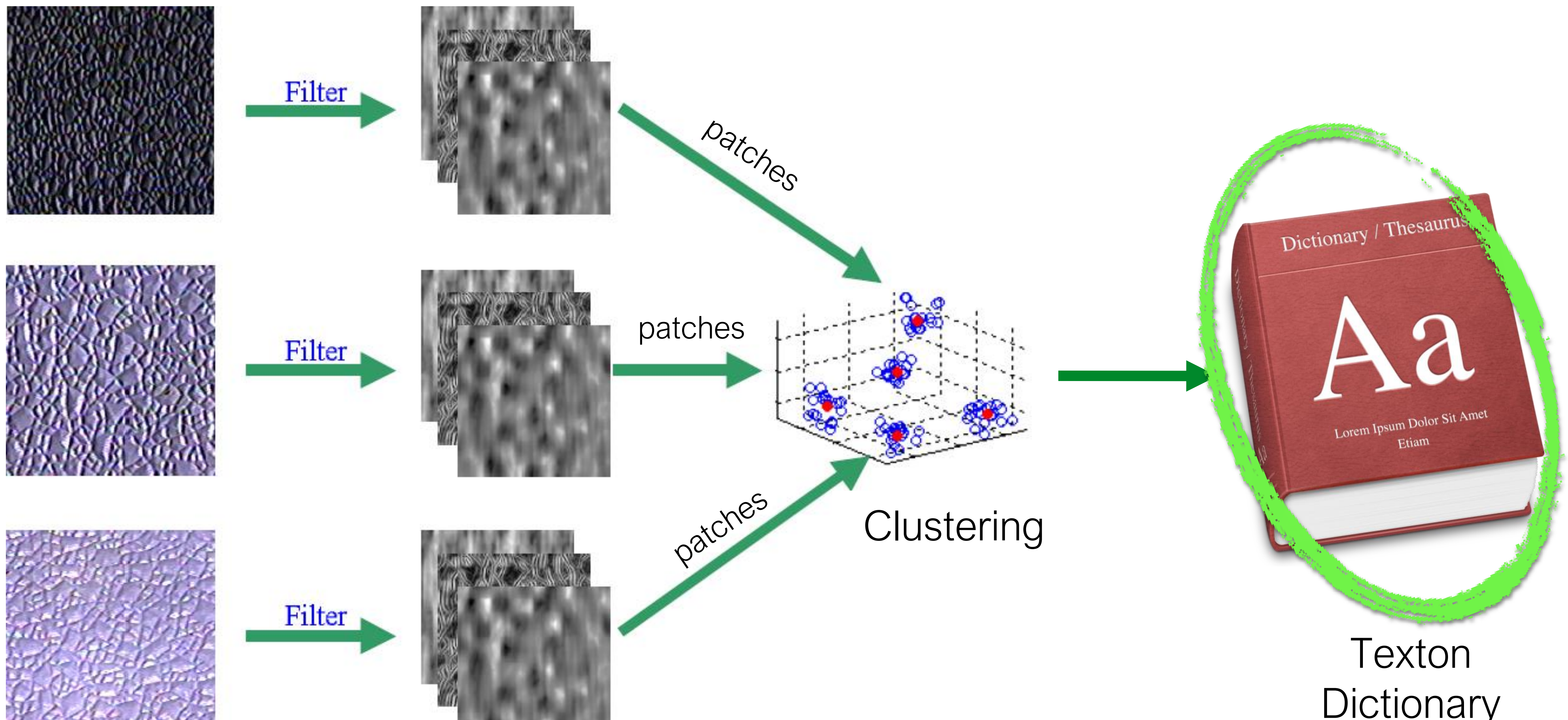


'LM'



'MR8'

Learning Textons from data

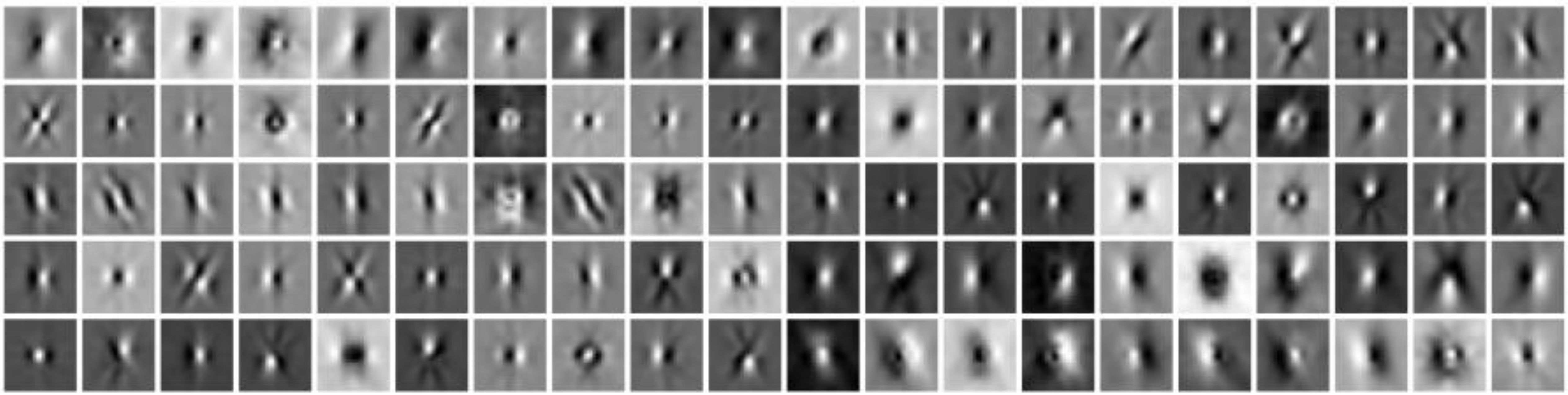


Multiple training images of the same texture

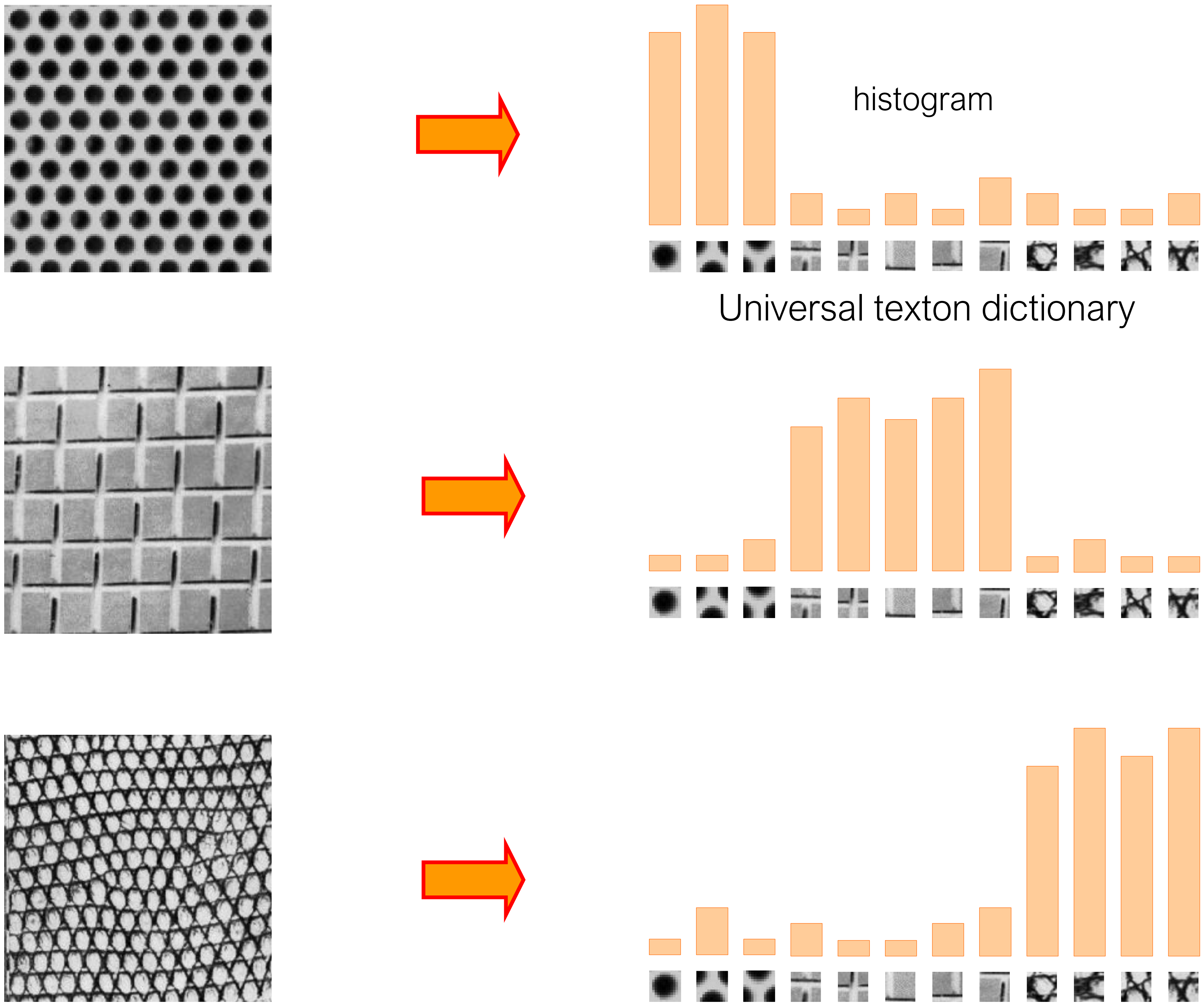
Filter response over a bank of filters

We will learn more about clustering later in class (Bag of Words lecture).

Texton Dictionary



Malik, Belongie, Shi, Leung. Textons, Contours and Regions: Cue Integration in Image Segmentation. ICCV 1999.



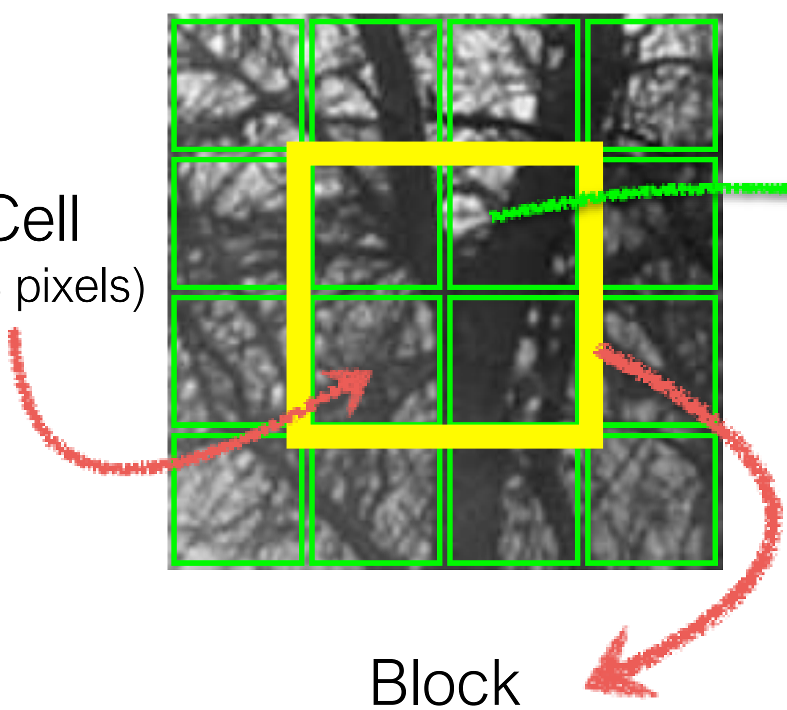
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

HOG descriptor

HOG

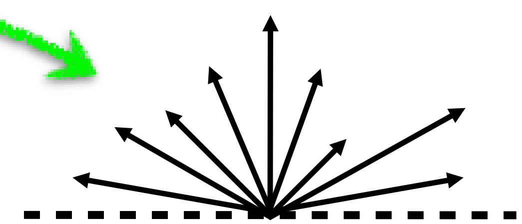
Dalal, Triggs. **Histograms of Oriented Gradients** for Human Detection. CVPR, 2005

Cell
(8x8 pixels)



Block
(2x2 cells)

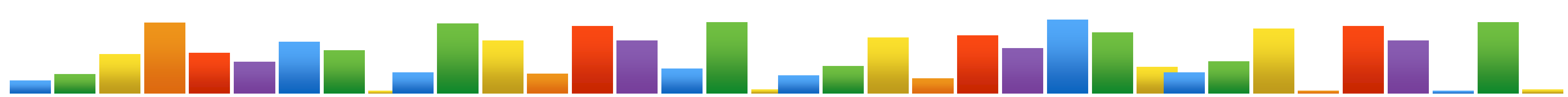
histogram of ‘unsigned’
gradients



soft binning

gradient magnitude histogram
(one for each cell)

Concatenate and L-2 normalization



Single scale, no dominant orientation

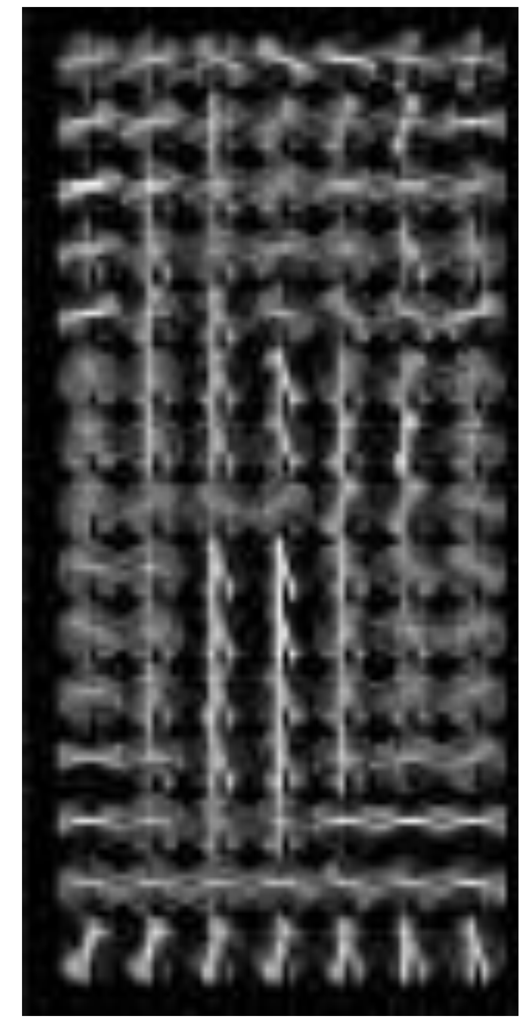
Pedestrian detection

1 cell step size

visualization

128 pixels
16 cells
15 blocks

15 x 7 x 4 x 36 =
3780



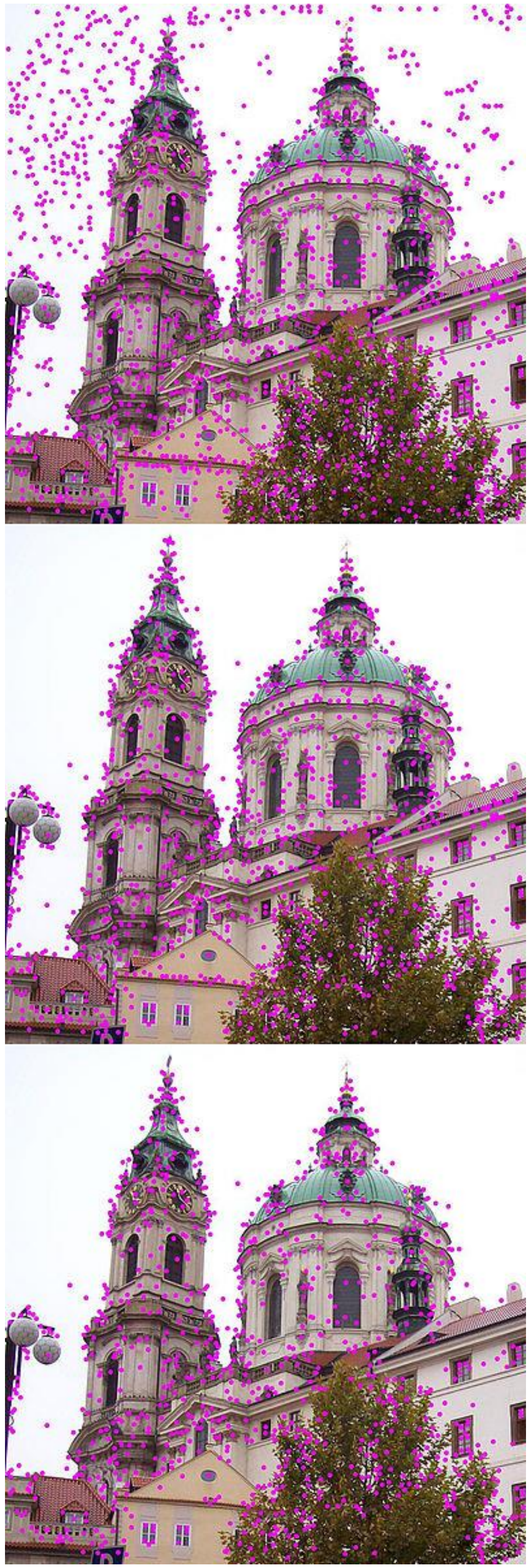
64 pixels
8 cells
7 blocks

Redundant representation due to overlapping blocks

How many times is each inner cell encoded?

http://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/

SIFT



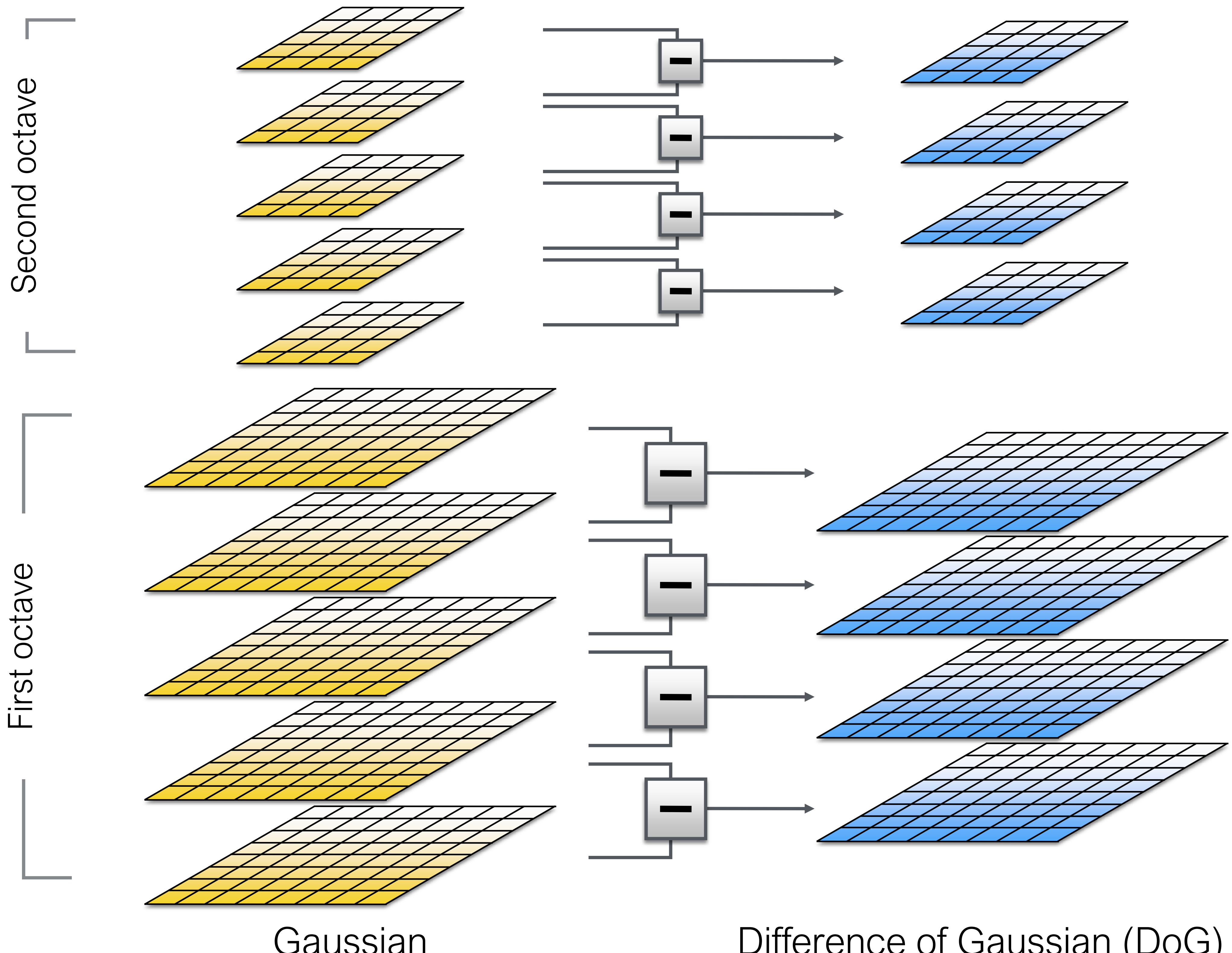
SIFT

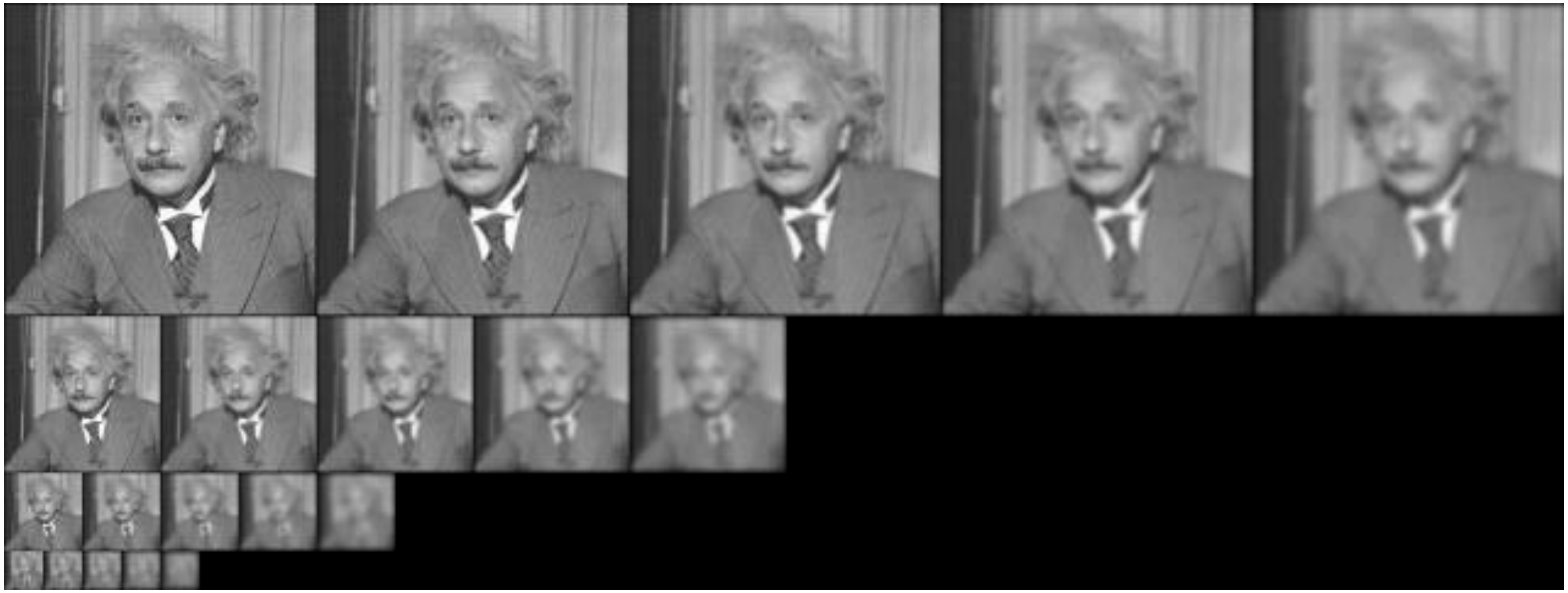
(Scale Invariant Feature Transform)

SIFT describes both a **detector** and **descriptor**

1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

1. Multi-scale extrema detection

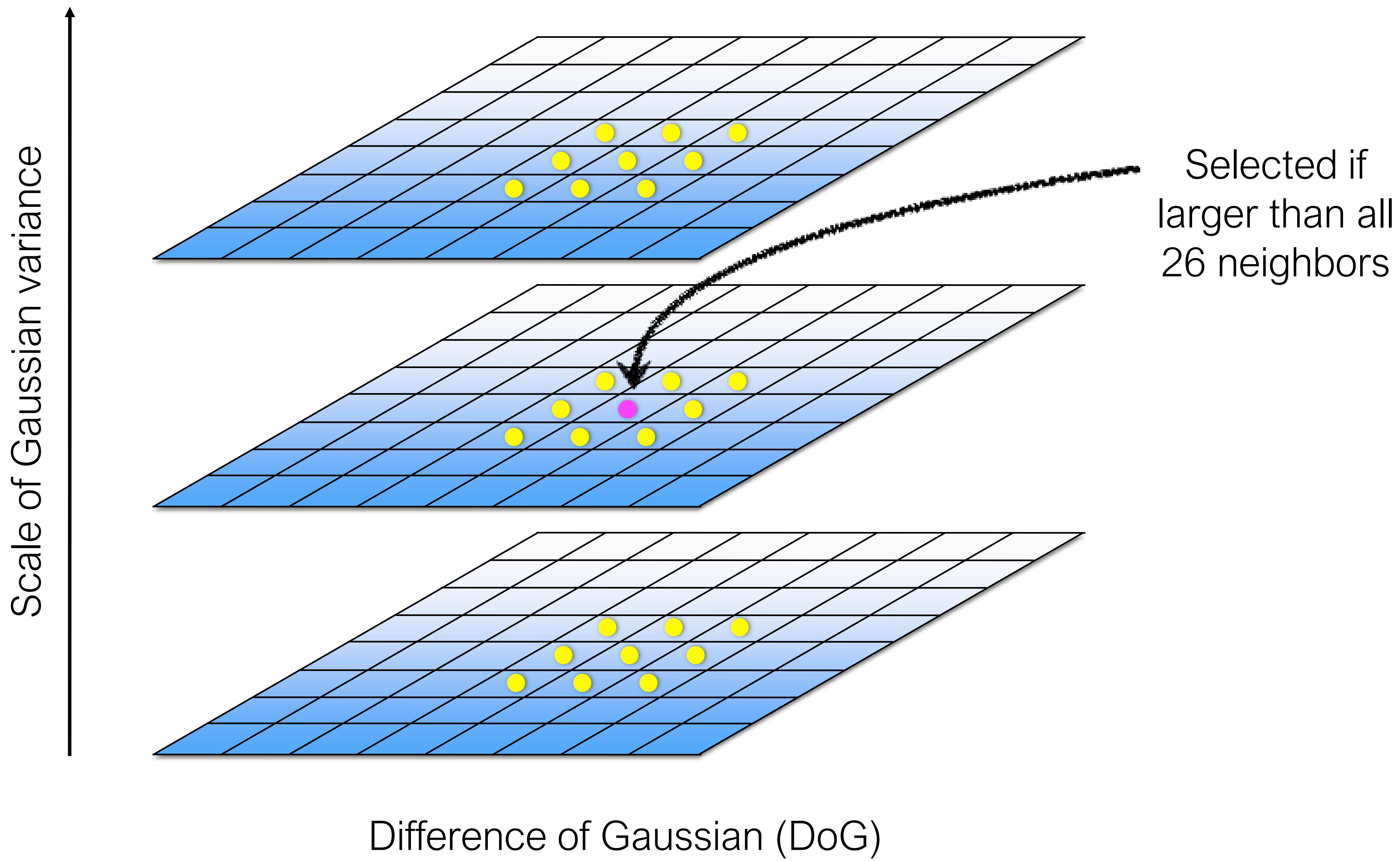




Gaussian

Laplacian

Scale-space extrema



2. Keypoint localization

2nd order Taylor series approximation of DoG scale-space

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\mathbf{x} = \{x, y, \sigma\}$$

Take the derivative and solve for extrema

$$\mathbf{x}_m = -\frac{\partial^2 f}{\partial \mathbf{x}^2}^{-1} \frac{\partial f}{\partial \mathbf{x}}$$

Additional tests to retain only strong features

3. Orientation assignment

For a keypoint, **L** is the **Gaussian-smoothed** image with the closest scale,

$$m(x,y) = \sqrt{\underbrace{(L(x+1,y) - L(x-1,y))^2}_{\text{x-derivative}} + \underbrace{(L(x,y+1) - L(x,y-1))^2}_{\text{y-derivative}}}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Detection process returns

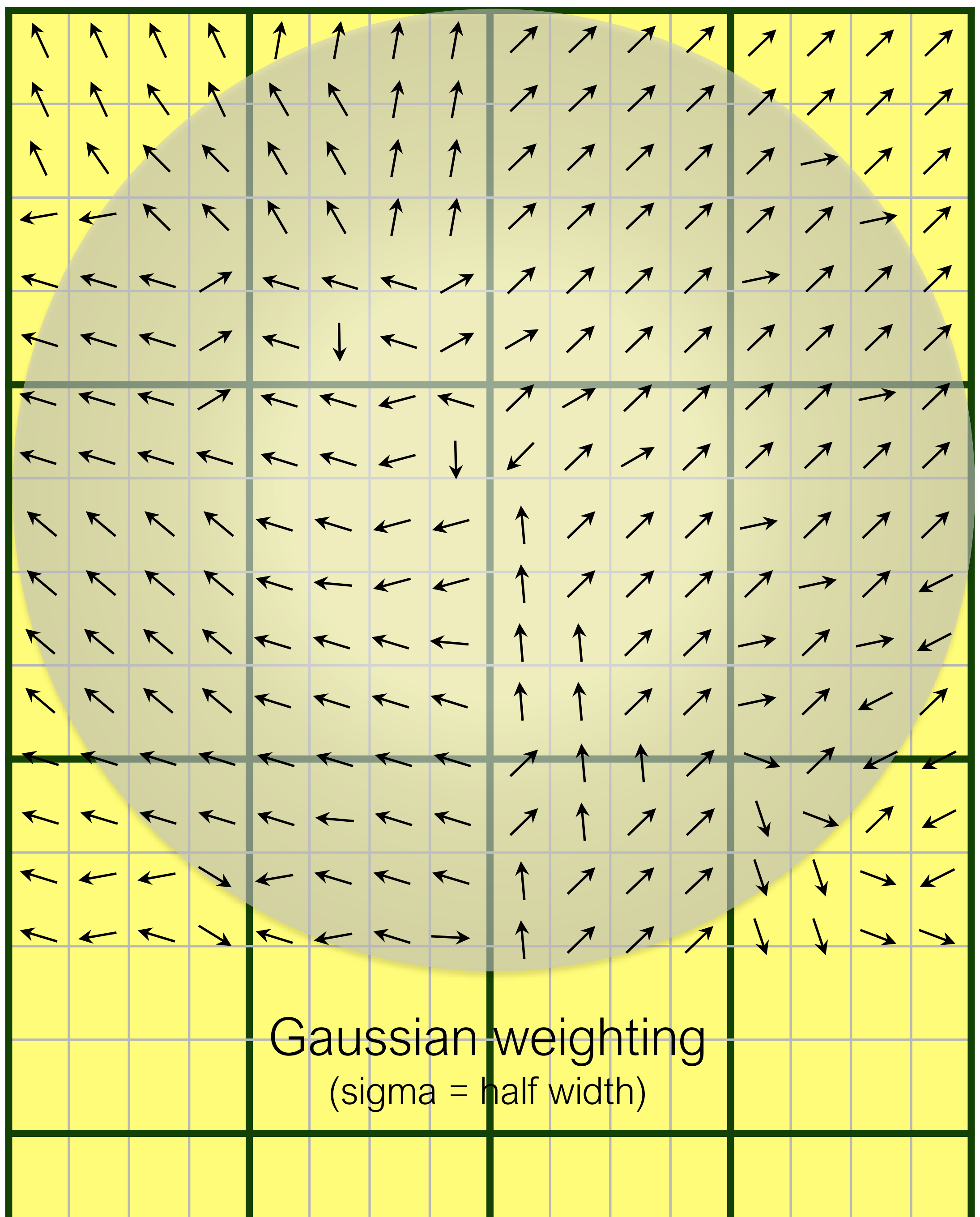
$$\{x, y, \sigma, \theta\}$$

location scale orientation

4. Keypoint descriptor

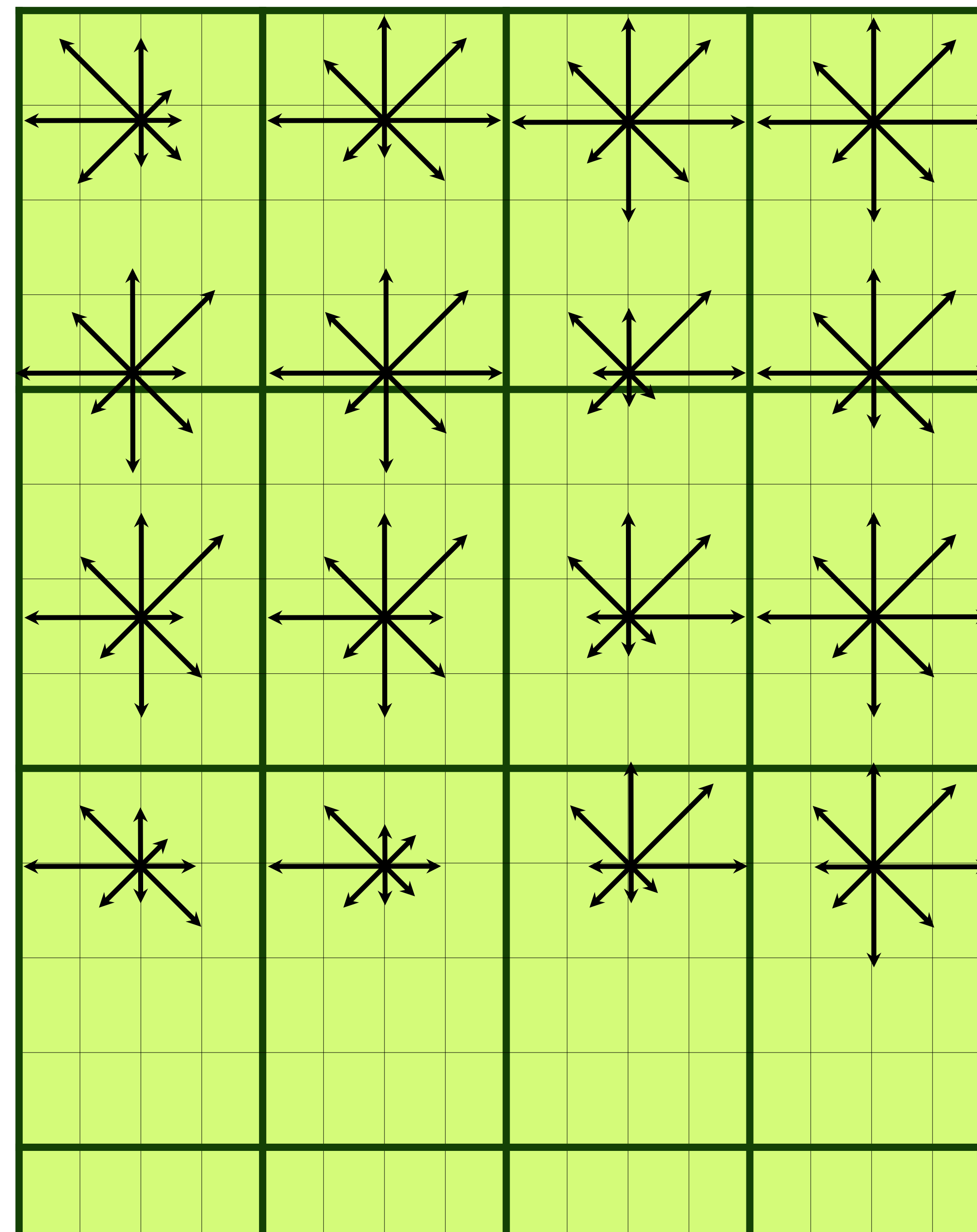
Image Gradients

(4 x 4 pixel per cell, 4 x 4 cells)

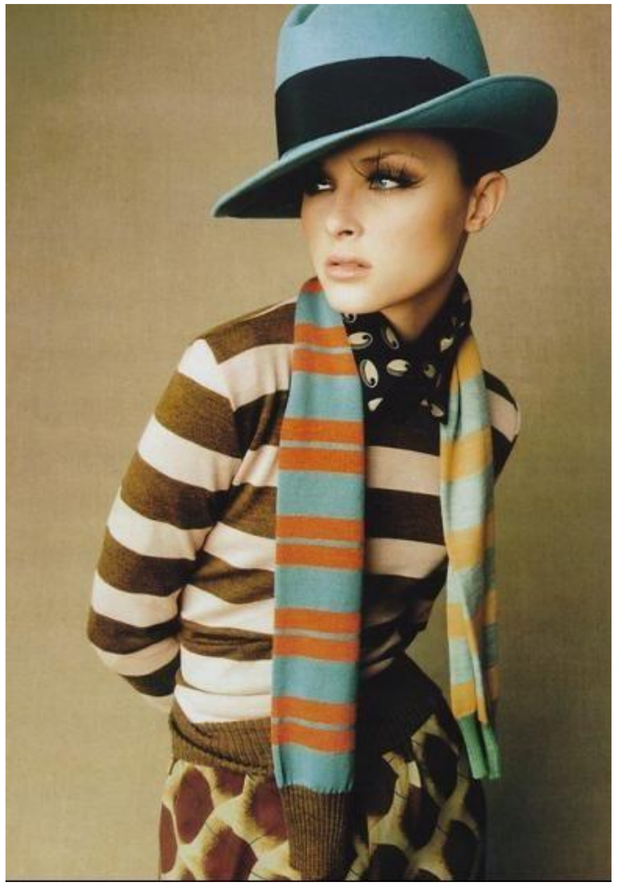


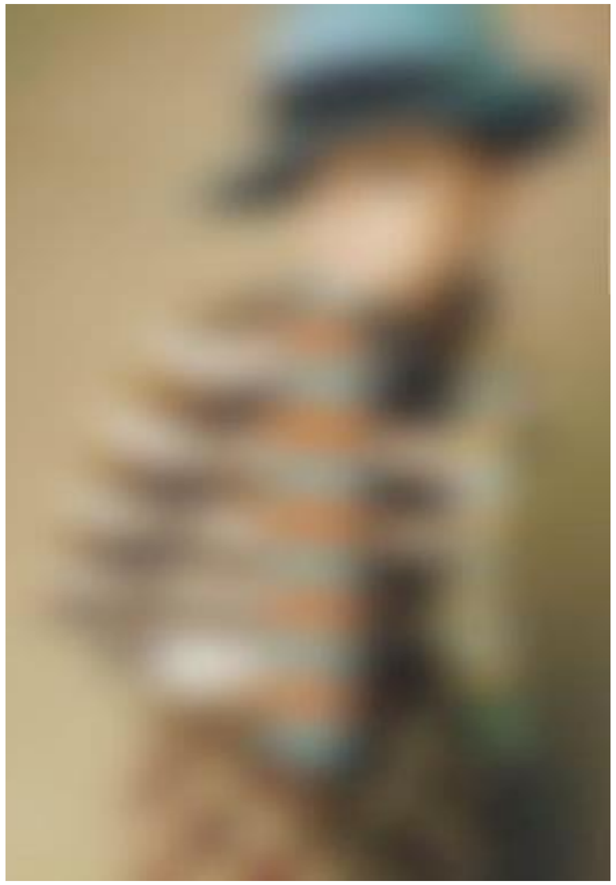
SIFT descriptor

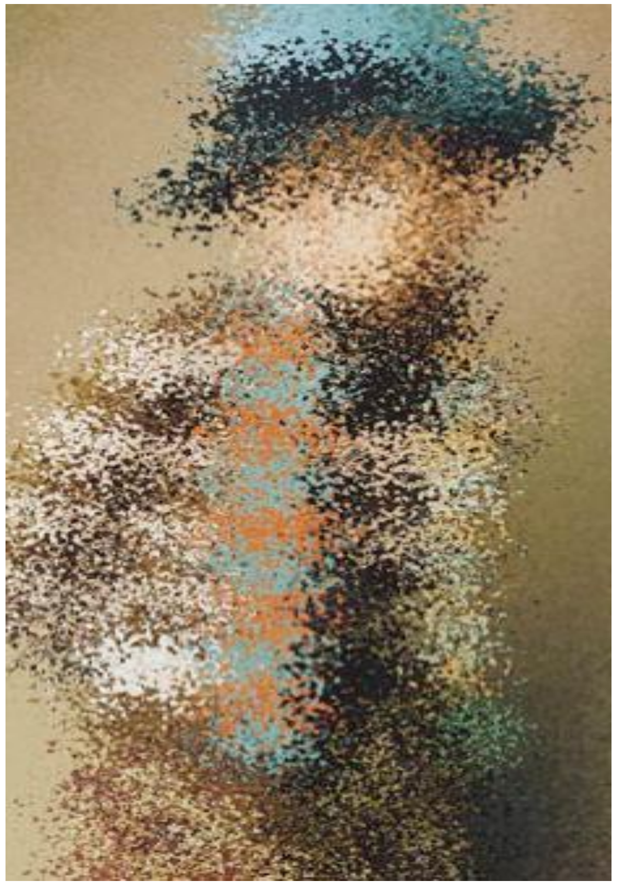
(16 cells x 8 directions = 128 dims)



Discriminative power


Raw pixels


Sampled


Locally orderless


Global histogram

Generalization power

References

Basic reading:

- Szeliski textbook, Sections 4.1.2, 14.1.2.