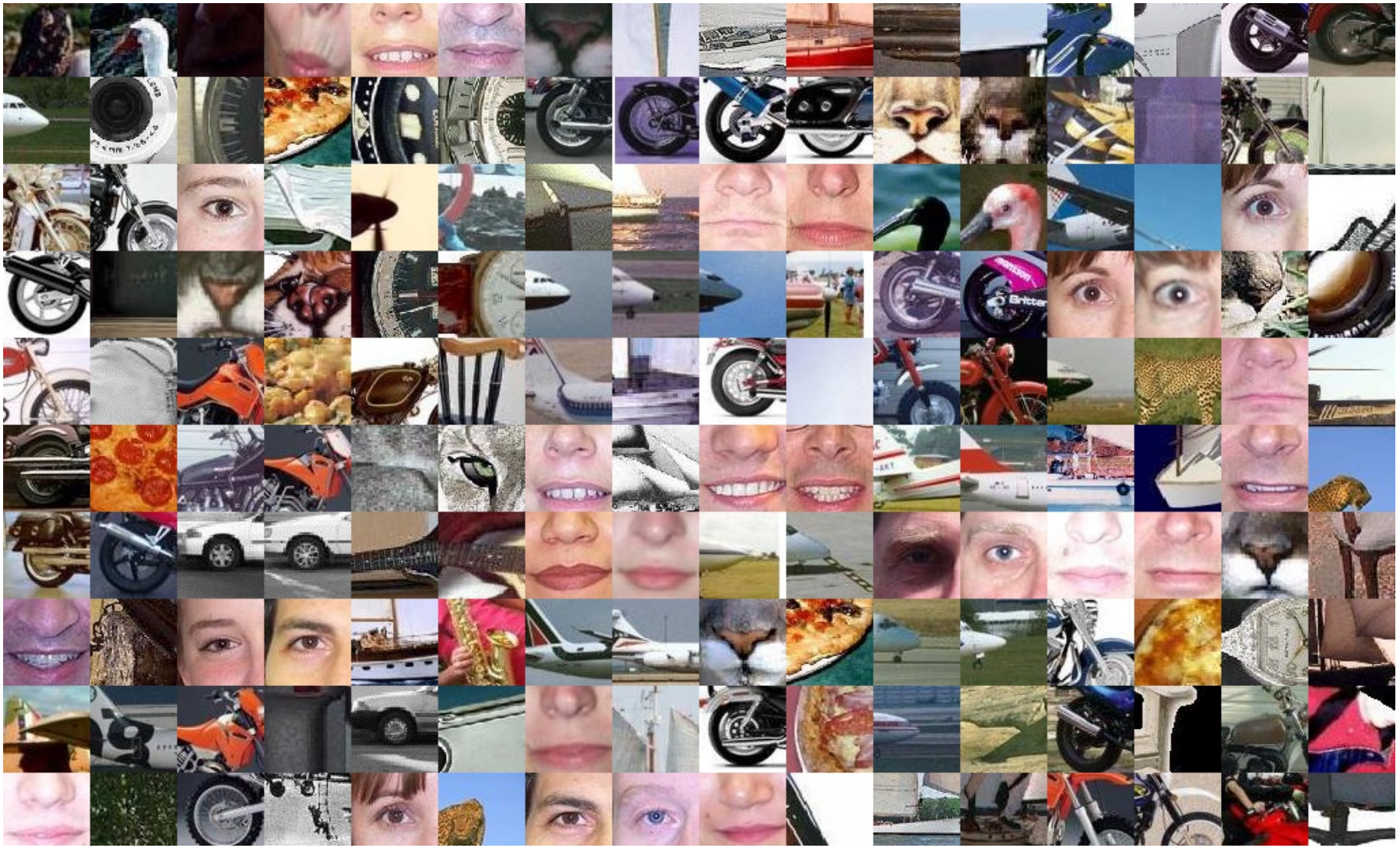
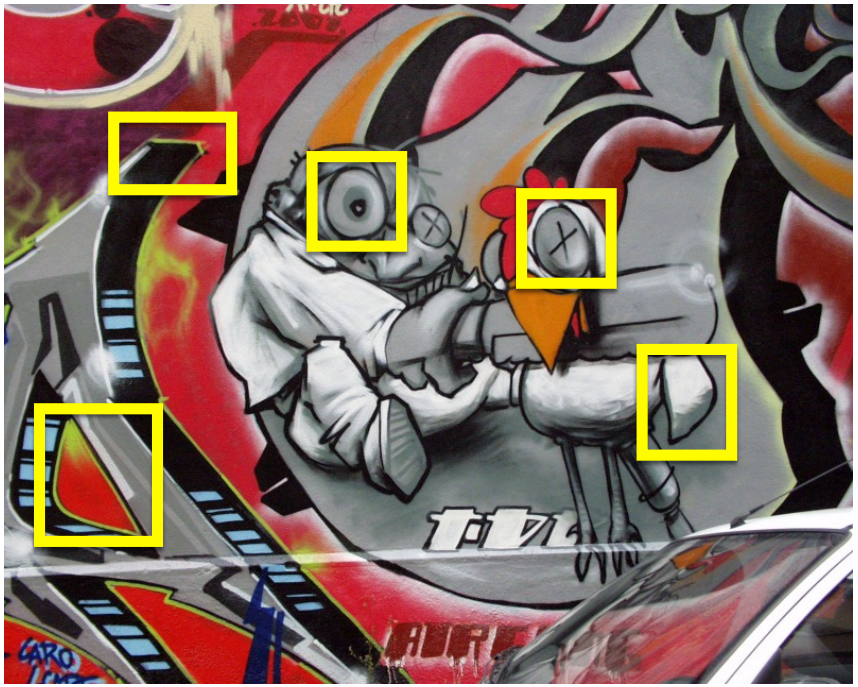
Designing descriptors

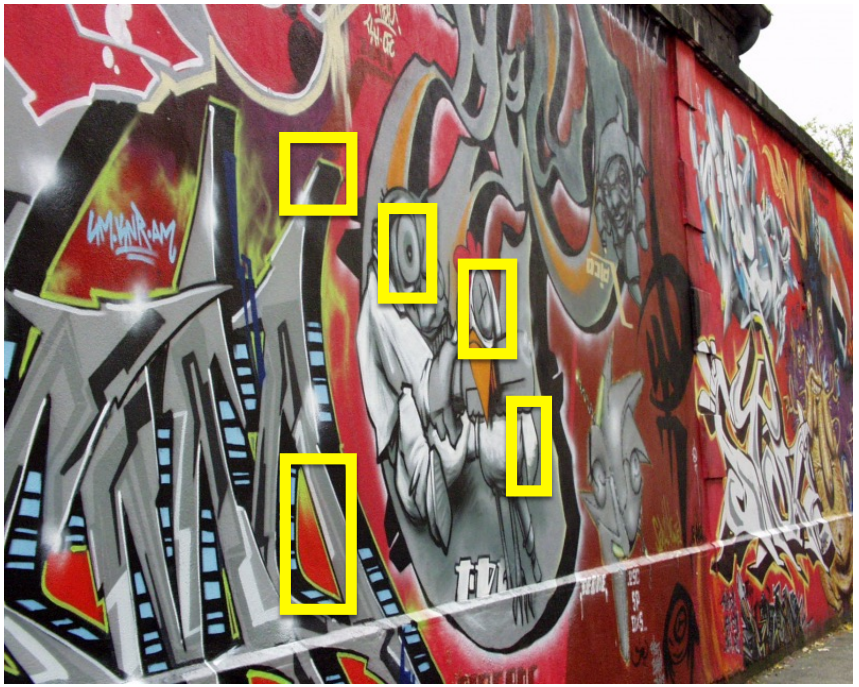


Overview

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

Why do we need feature descriptors?

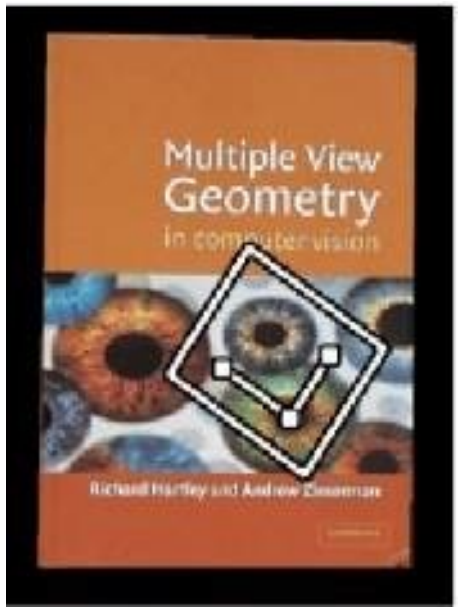




If we know <u>where</u> the "good" features are,
how do we <u>match</u> them?


How do we describe an image patch?
Patches with similar content should have similar descriptors.

Transformations



objects will appear at different
scales, translations, and rotations

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?

Image patch

Just use the pixel values of the patch

Perfectly fine if geometry and appearance is unchanged

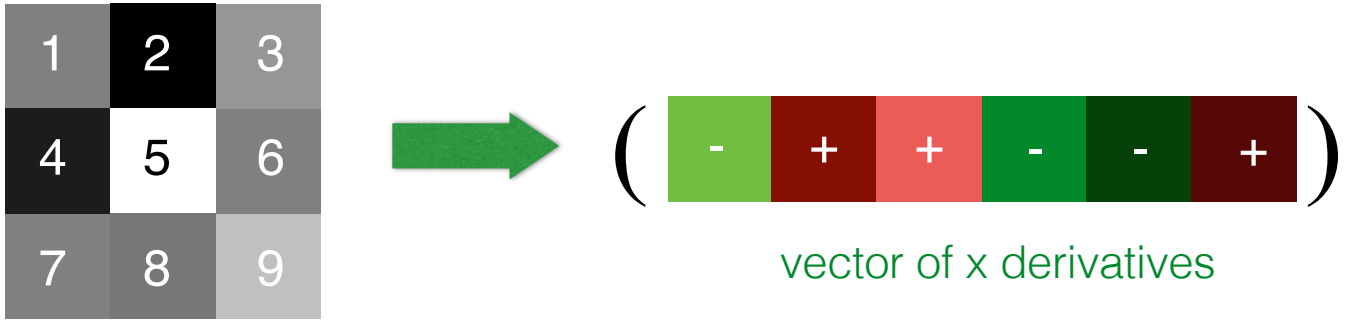
(a.k.a. template matching)

What are the problems?

How can you be less sensitive to absolute intensity values?

Image gradients

Use pixel differences

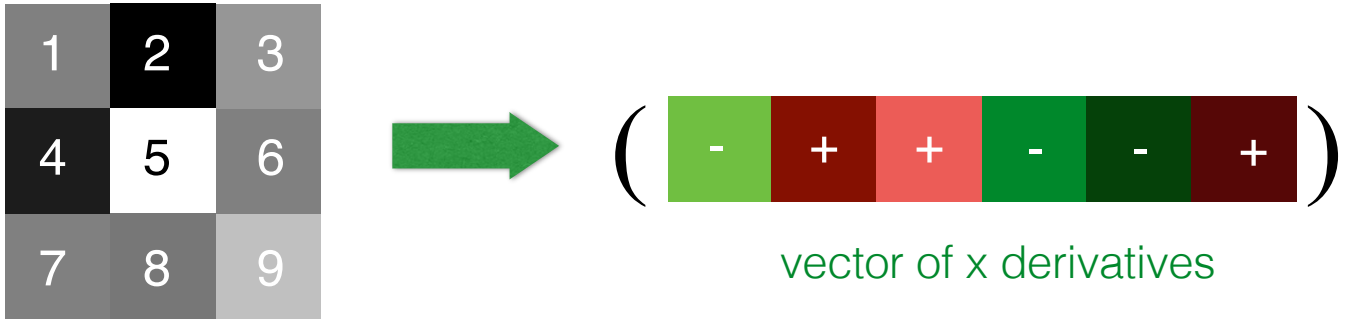


Feature is invariant to absolute intensity values

What are the problems?

Image gradients

Use pixel differences

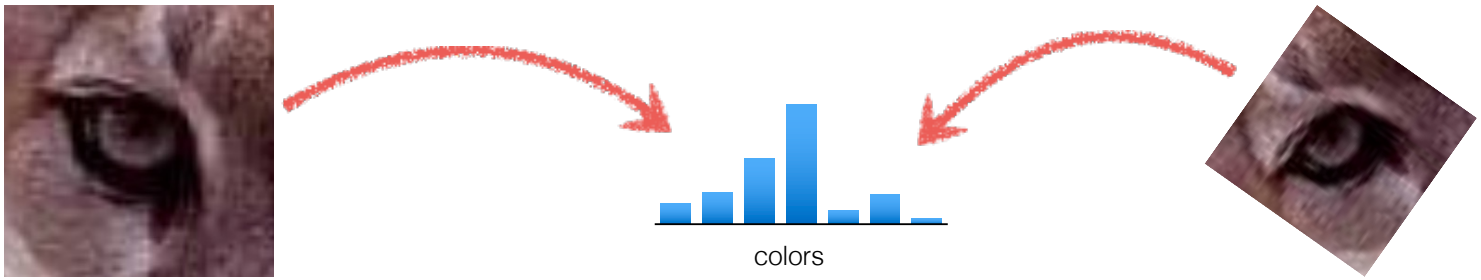


Feature is invariant to absolute intensity values

How can you be less sensitive to deformations?

Color histogram

Count the colors in the image using a histogram

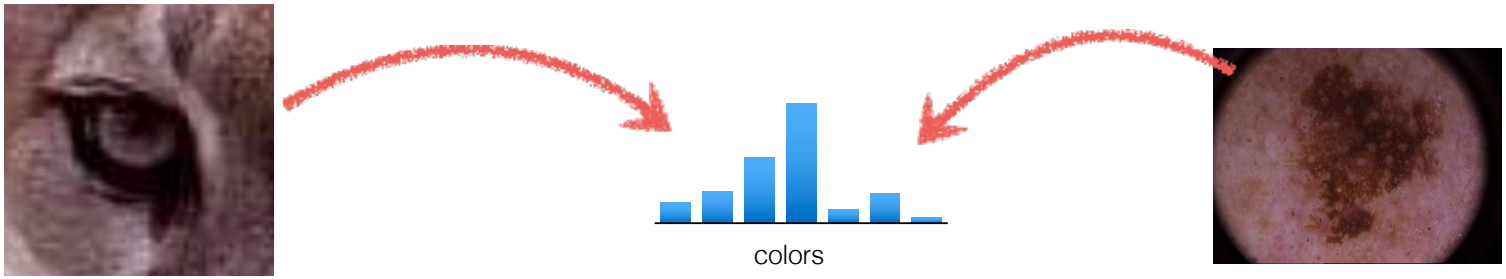


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram

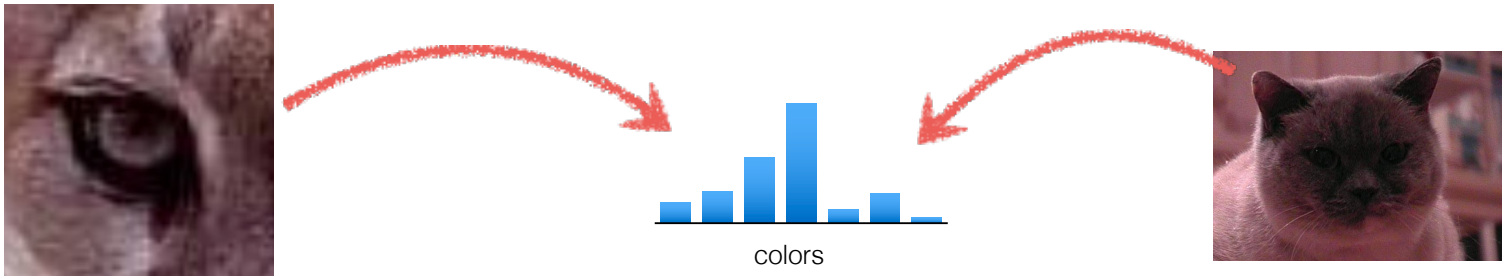


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram



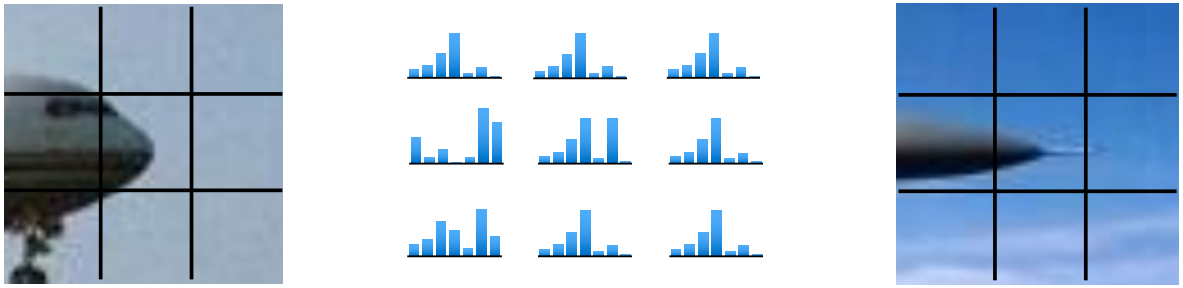
Invariant to changes in scale and rotation

What are the problems?

How can you be more sensitive to spatial layout?

Spatial histograms

Compute histograms over spatial ‘cells’



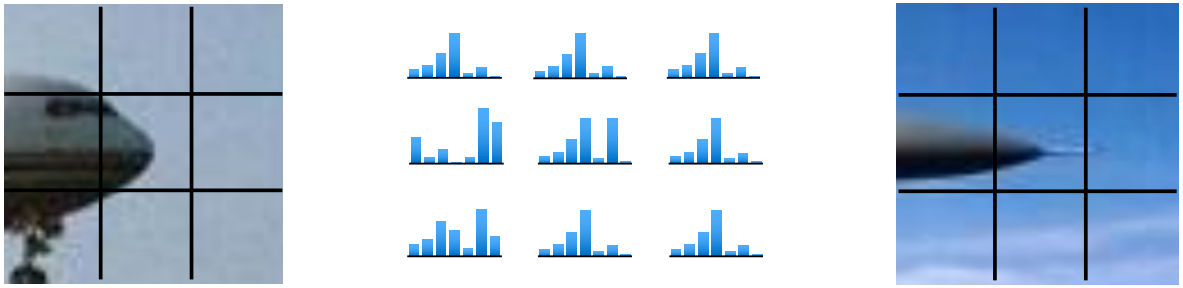
Retains rough spatial layout

Some invariance to deformations

What are the problems?

Spatial histograms

Compute histograms over spatial ‘cells’



Retains rough spatial layout

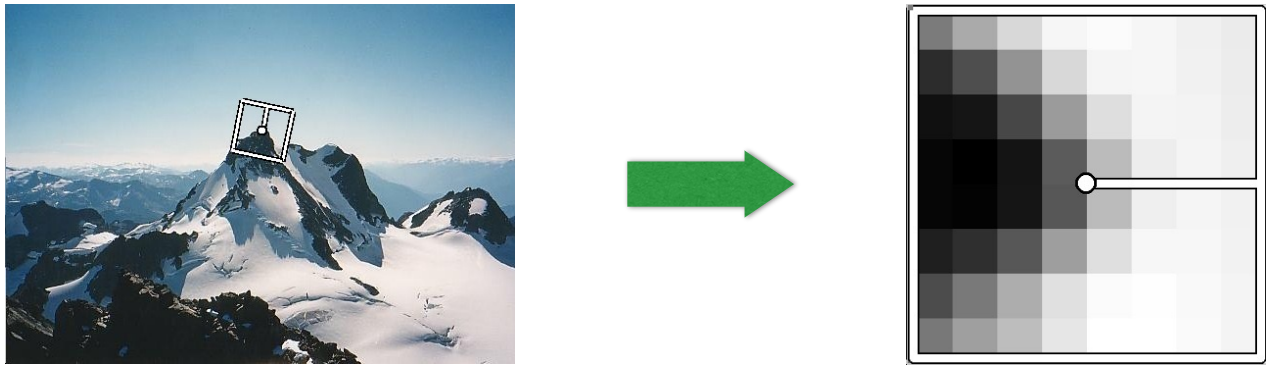
Some invariance to deformations

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



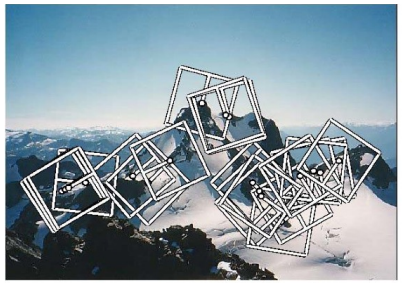
save the orientation angle θ along with (x, y, s)

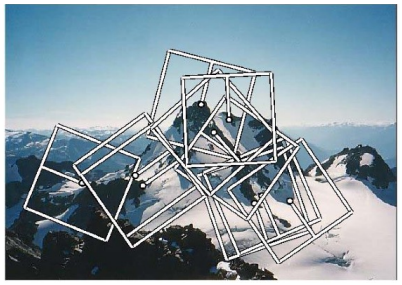
What are the problems?

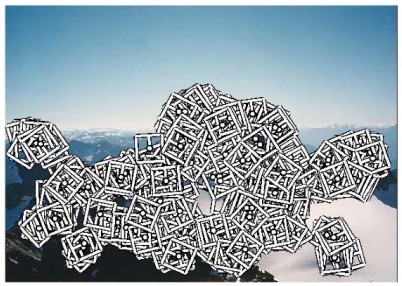
MOPS descriptor

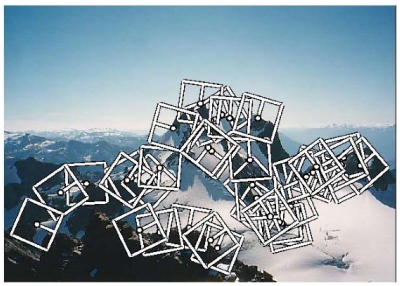
Multi-Scale Oriented Patches (MOPS)

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







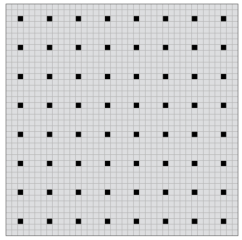


Multi-Scale Oriented Patches (MOPS)

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?*)

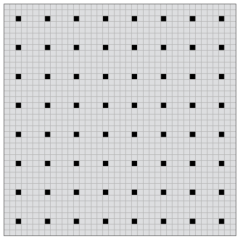
Haar Wavelet Transform
(*what's the purpose of this step?*)

Multi-Scale Oriented Patches (MOPS)

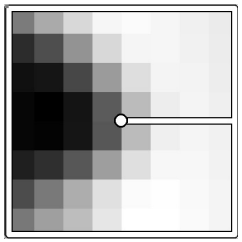
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

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?*)



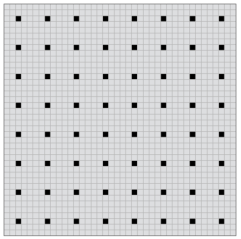
Haar Wavelet Transform
(*what's the purpose of this step?*)

Multi-Scale Oriented Patches (MOPS)

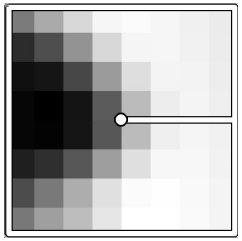
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

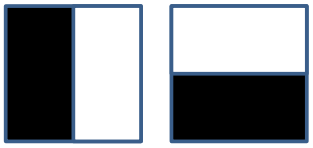
Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



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



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

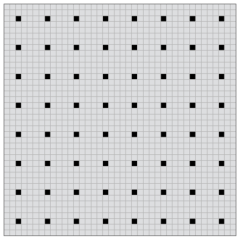


Multi-Scale Oriented Patches (MOPS)

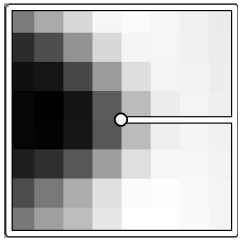
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

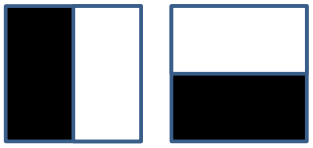
Get 40 x 40 image patch,
subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



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



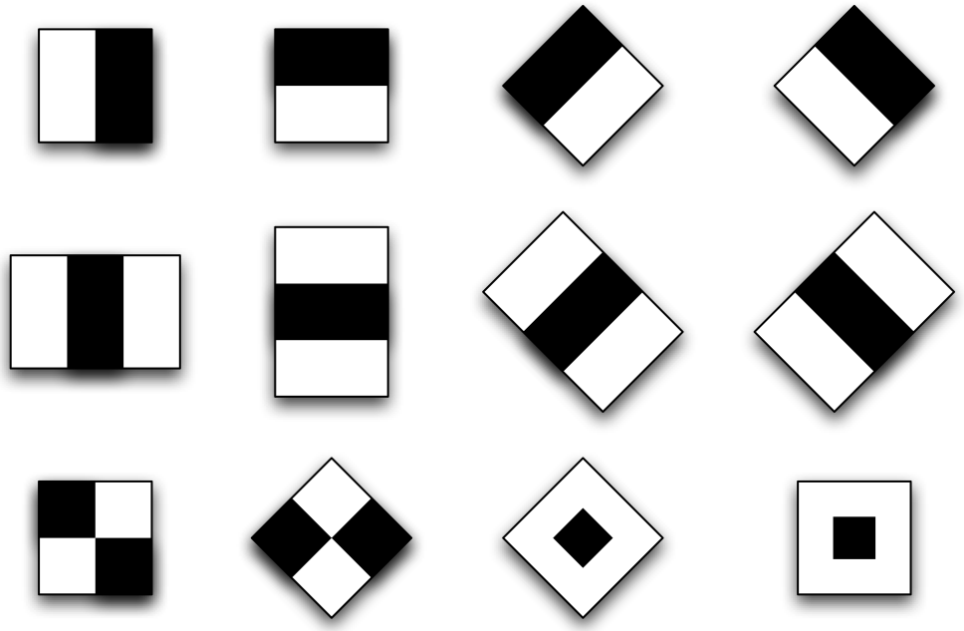
Haar Wavelet Transform
(low frequency projection)



Haar Wavelets

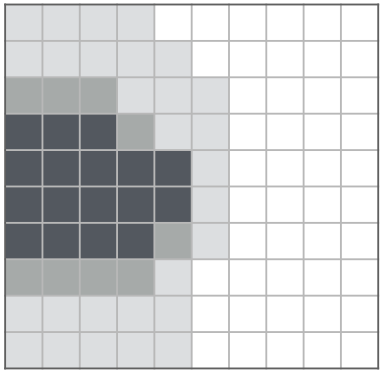
(actually, Haar-like features)

Use responses of a bank of filters as a descriptor

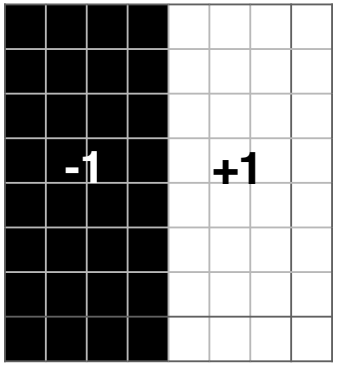


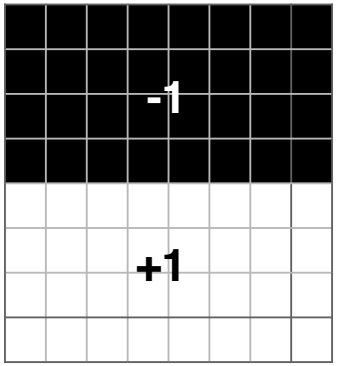
Haar wavelet responses can be computed with filtering

image patch



Haar wavelets filters



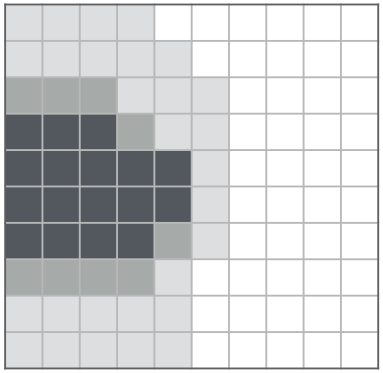


...

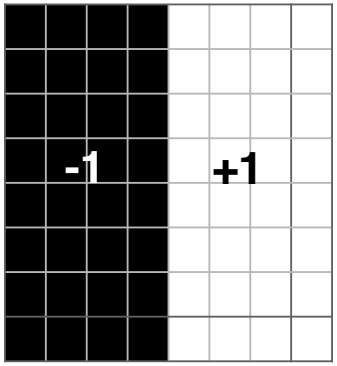
Filter application is one step of convolution!

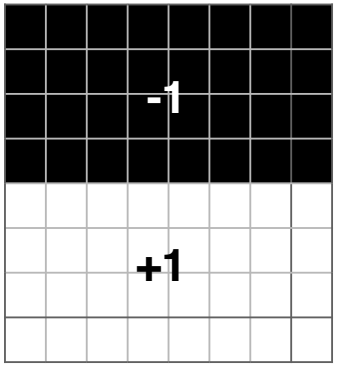
Haar wavelet responses can be computed with filtering

image patch



Haar wavelets filters





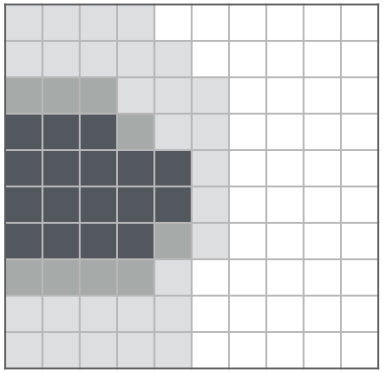
…

-45

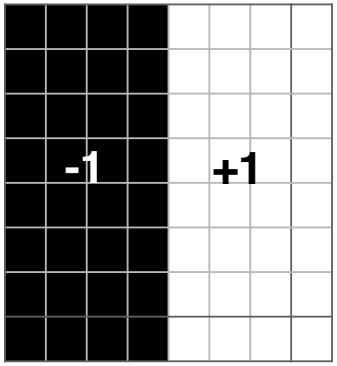
Filter application is one step of convolution!

Haar wavelet responses can be computed with filtering

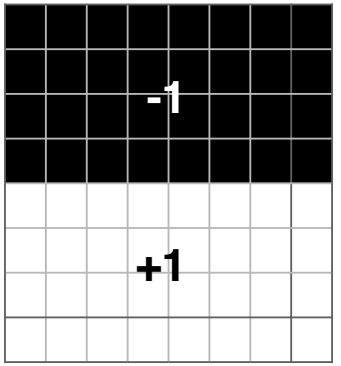
image patch



Haar wavelets filters



-45



16

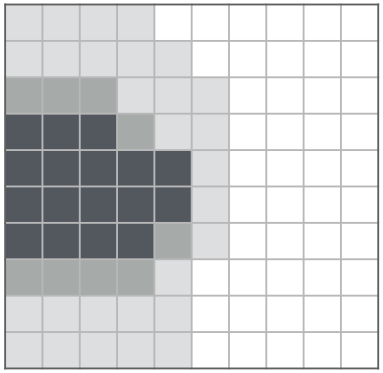
…

…

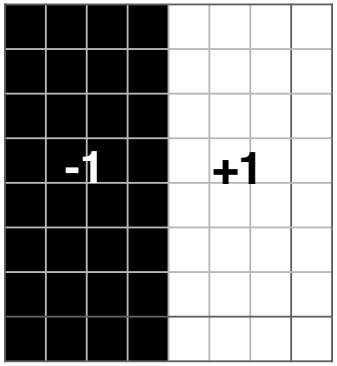
Filter application is one step of convolution!

Haar wavelet responses can be computed with filtering

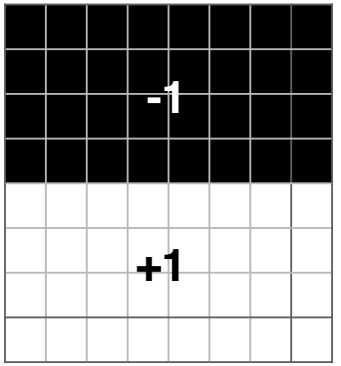
image patch



Haar wavelets filters



-45



16

…

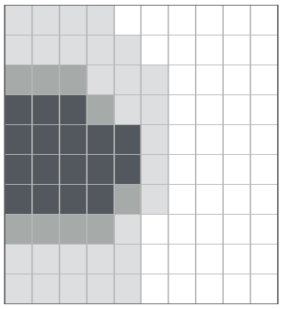
…

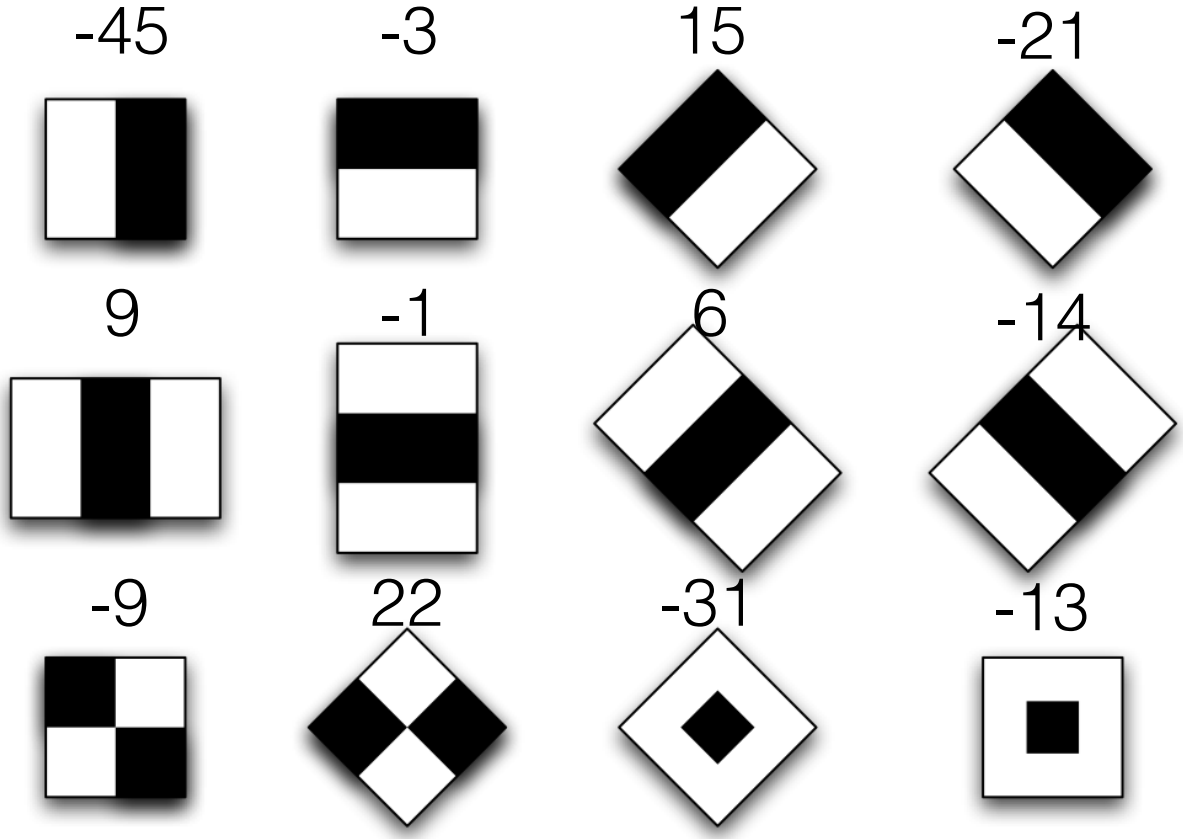
Filter application is one step of convolution!

Constant time w/ integral images (explained soon)

Descriptor = 12 dim vector

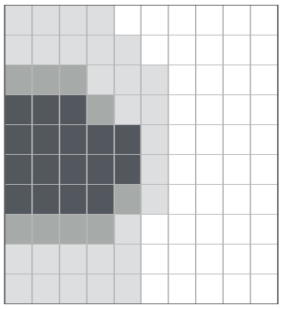
image patch

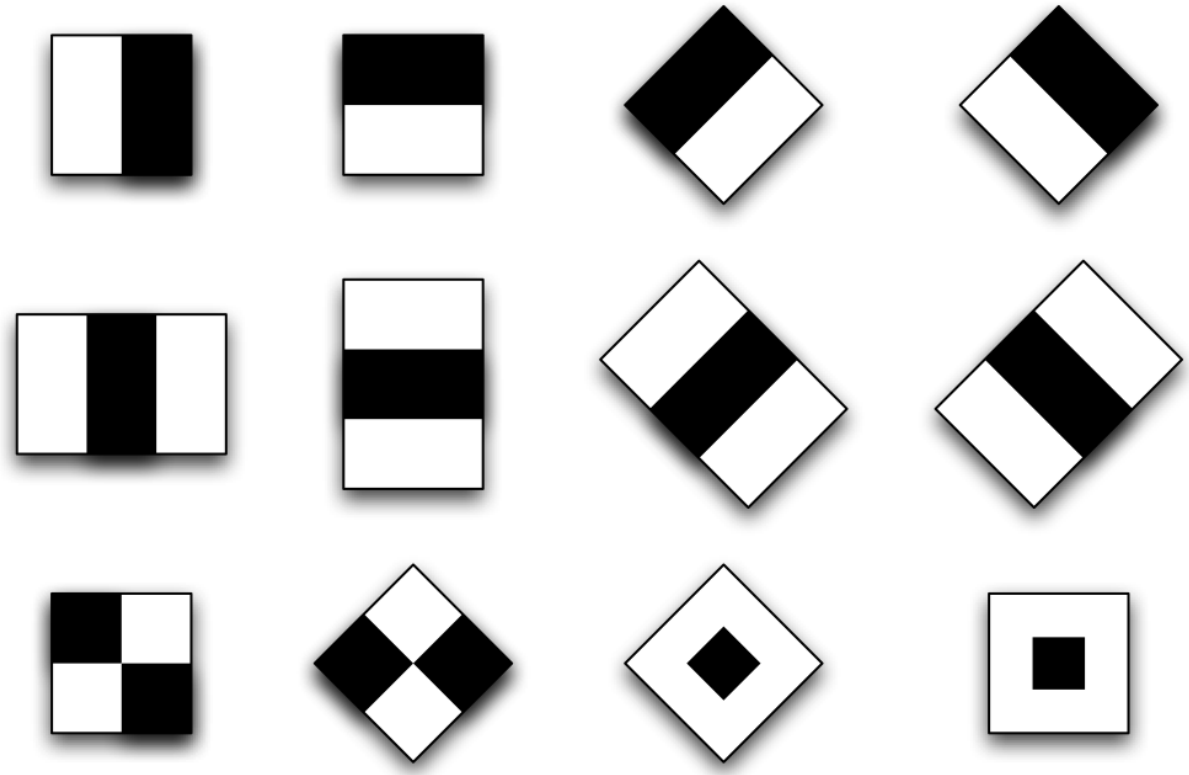




Descriptor = 12 dim vector

image patch





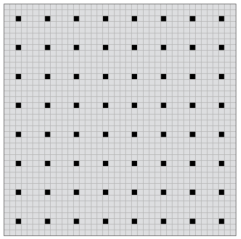
[-45 -3 15 -21 9 -1 6 -14 -9 22 -31 -13]

Multi-Scale Oriented Patches (MOPS)

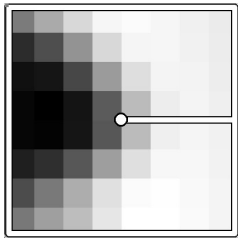
Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

Given a feature (x, y, s, θ)

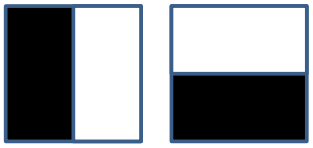
Get 40 x 40 image patch,
subsample every 5th pixel
(low frequency filtering, absorbs localization errors)



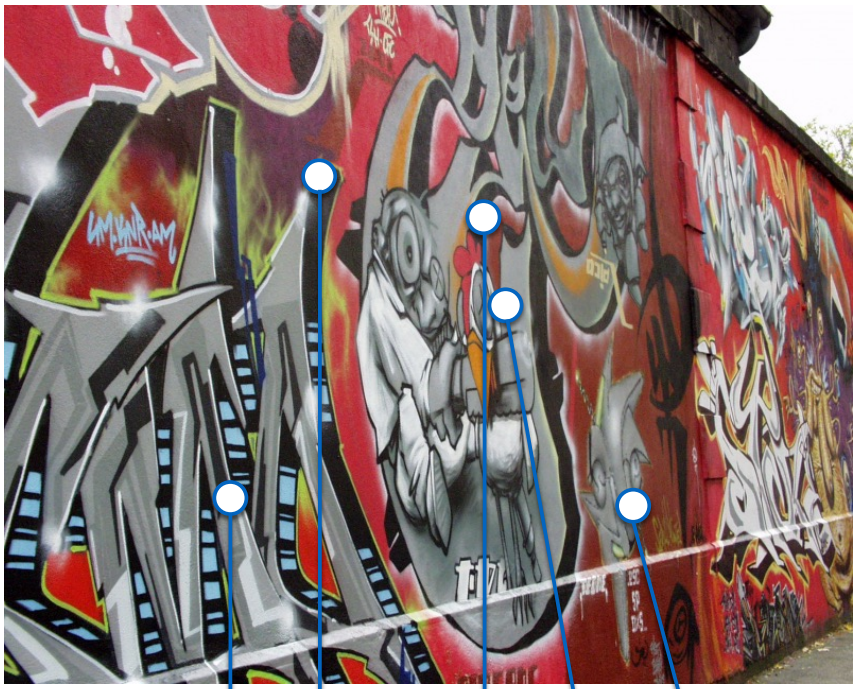
Subtract the mean, divide by
standard deviation
(removes bias and gain)



Haar Wavelet Transform
(low frequency projection)



(---) (---) (---) (---) (---) (---)

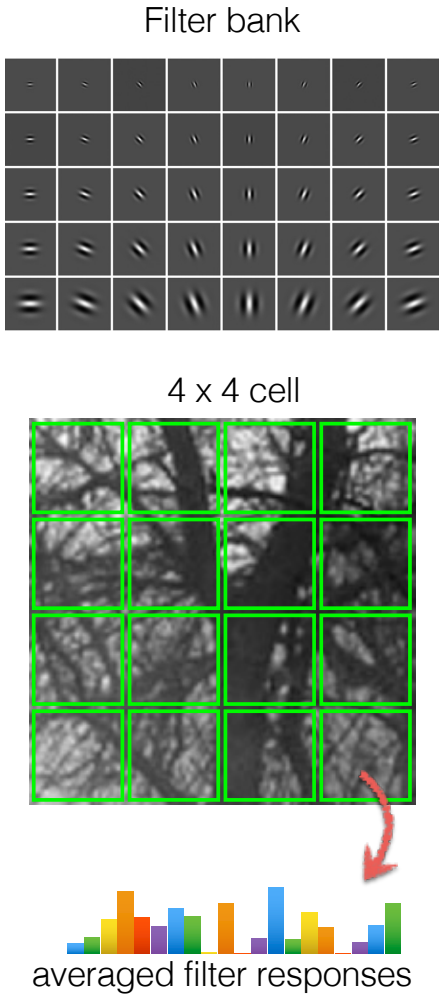


(---) (---) (---) (---) (---)

GIST

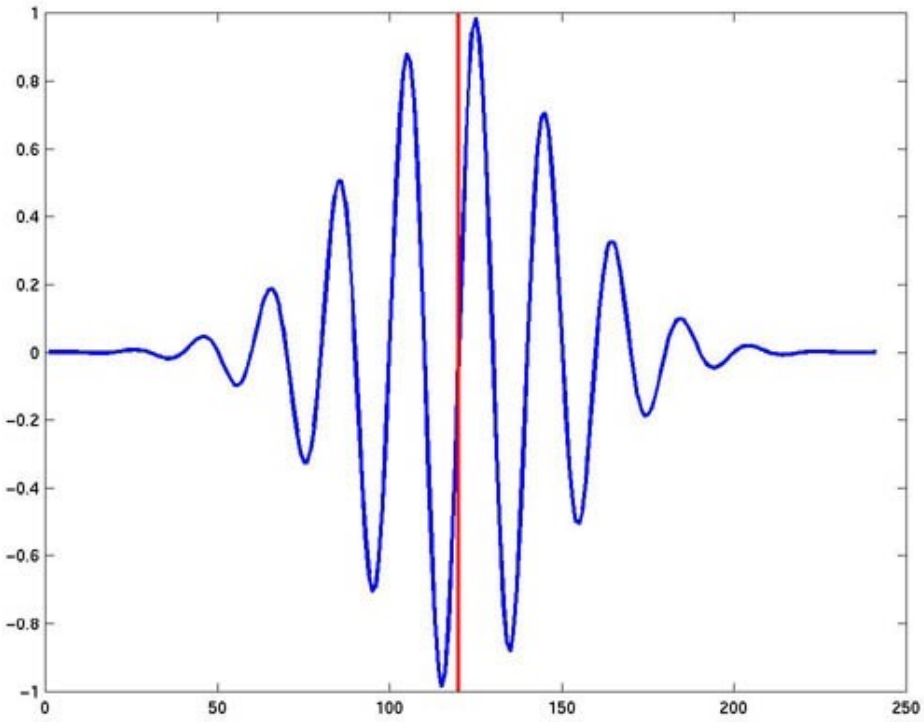
GIST

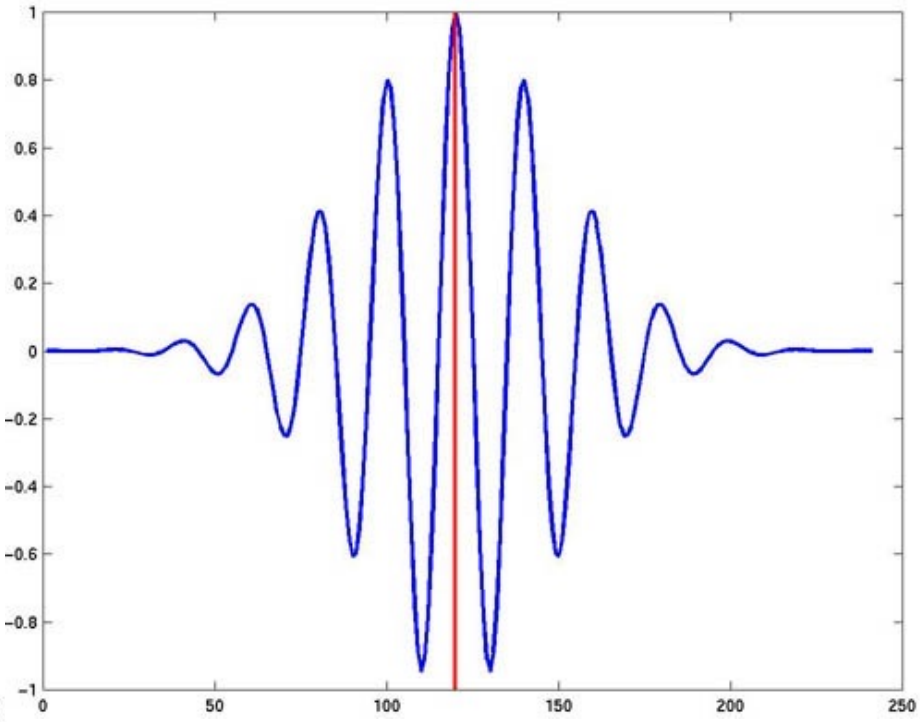
1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank



Gabor Filters

(1D examples)

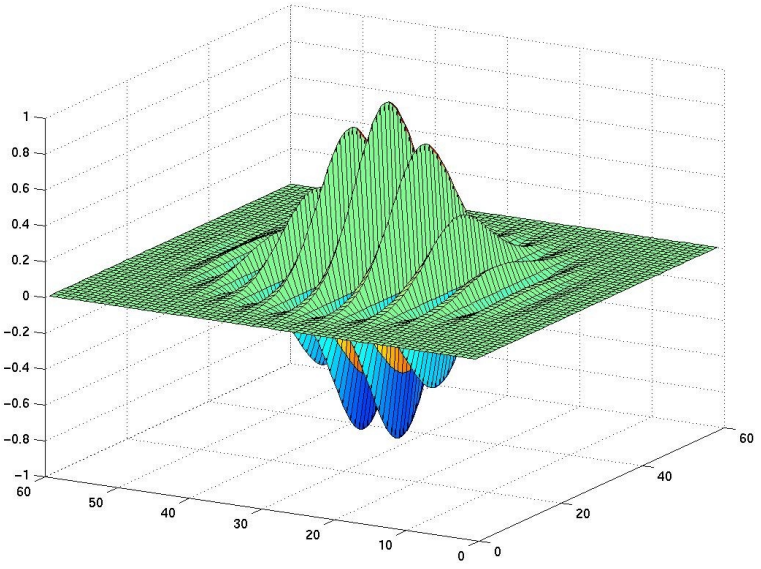


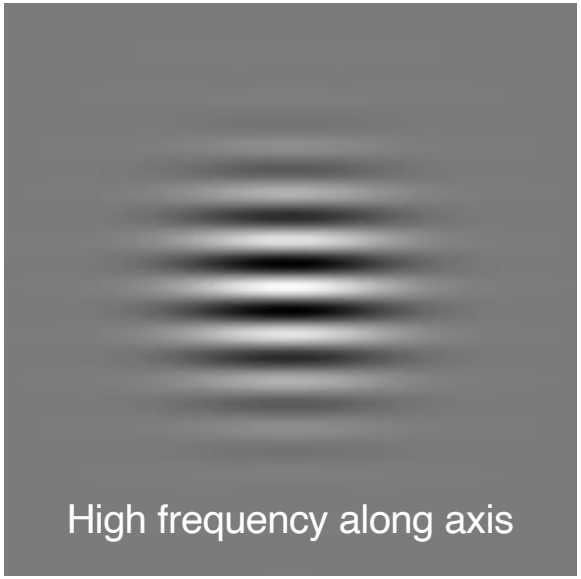


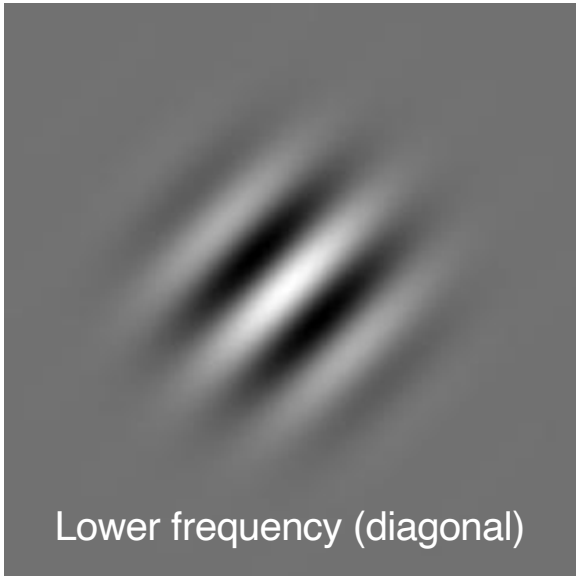
$$e^{-\frac{x^2}{2\sigma^2}} \sin(2\pi\omega x) \qquad e^{-\frac{x^2}{2\sigma^2}} \cos(2\pi\omega x)$$

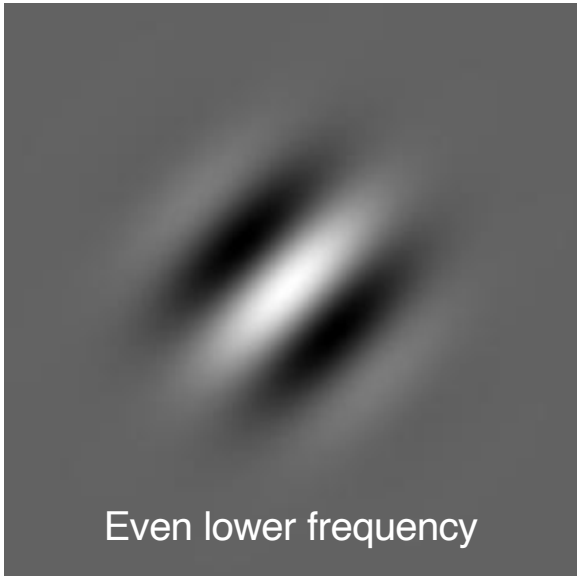
2D Gabor Filters

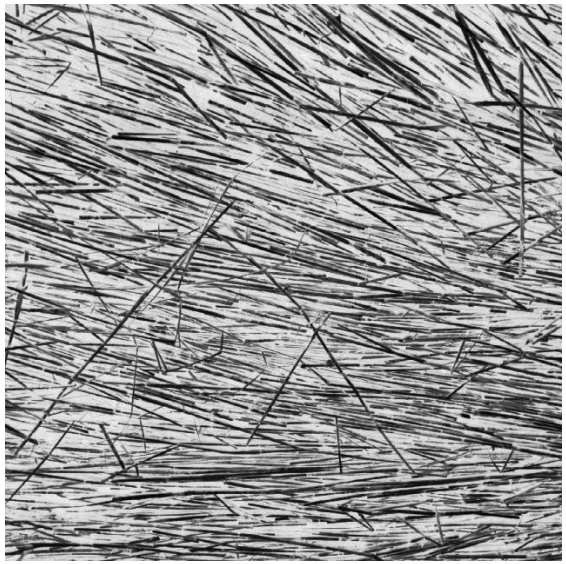
$$e^{-\frac{x^2+y^2}{2\sigma^2}} \cos\left(2\pi\left(k_x x + k_y y\right)\right)$$

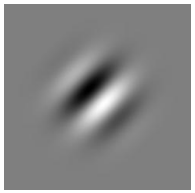


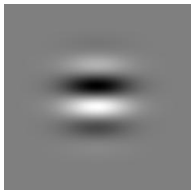


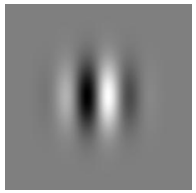


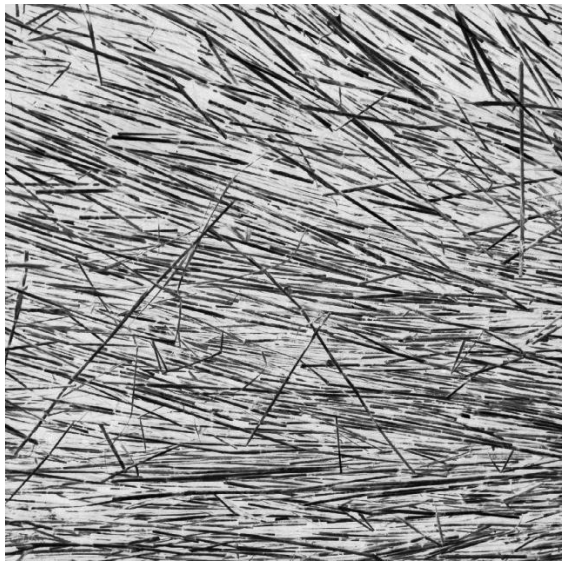


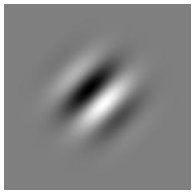


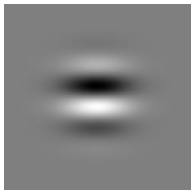


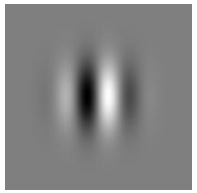


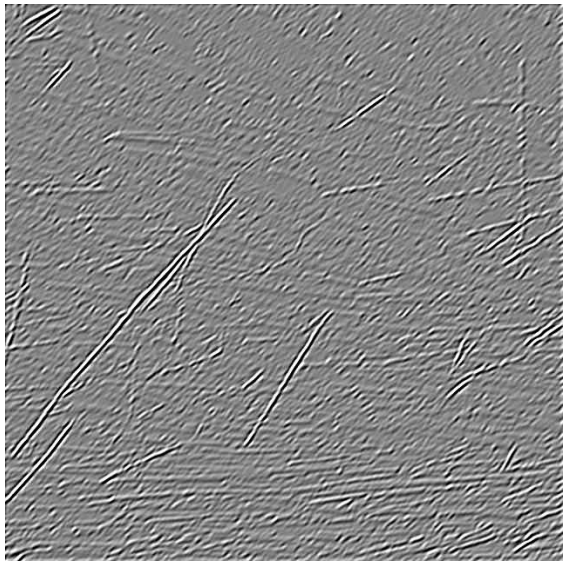


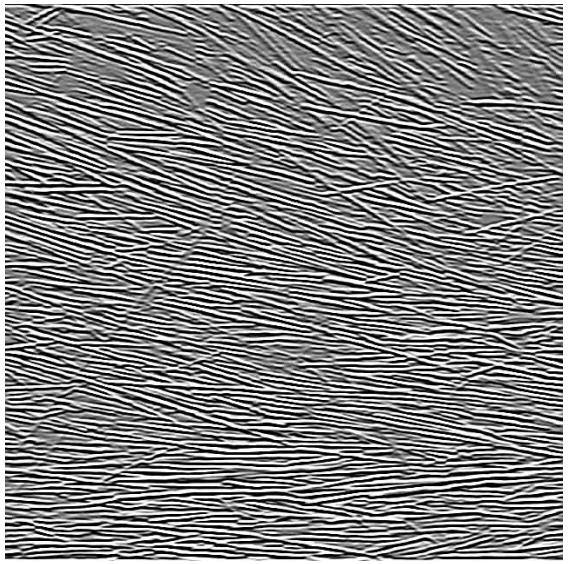


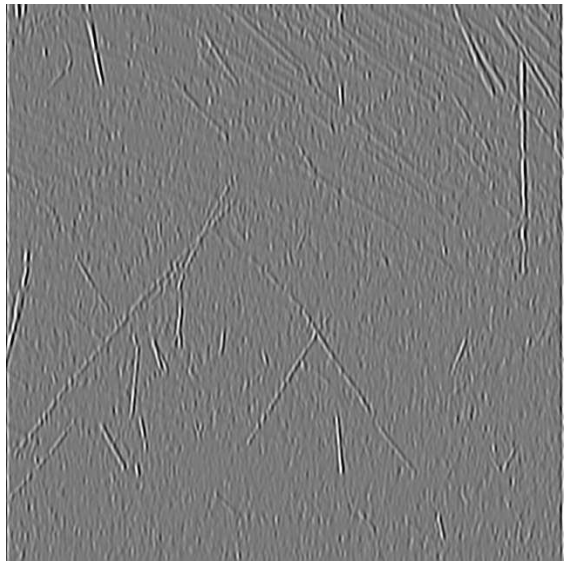


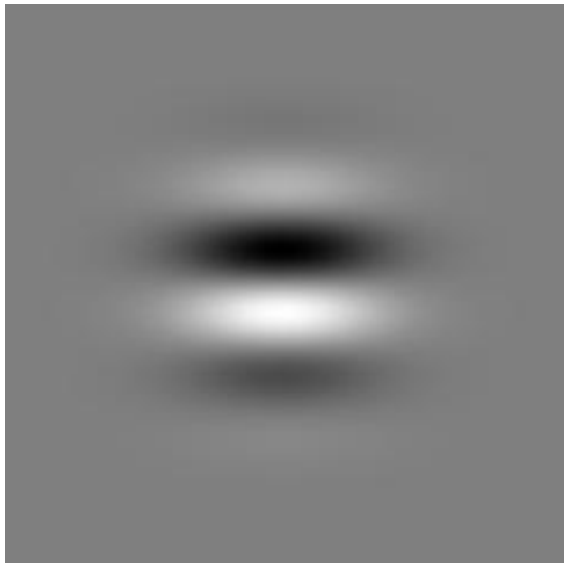




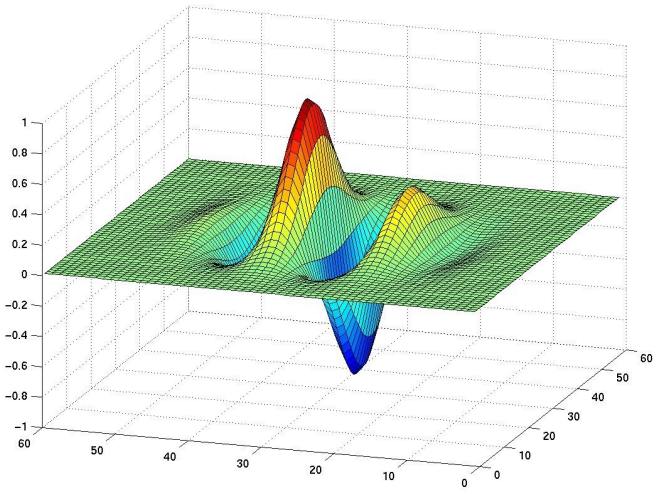




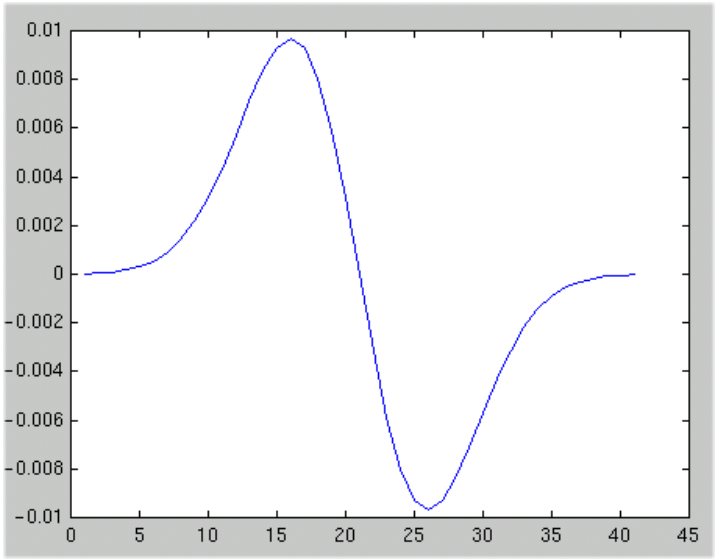




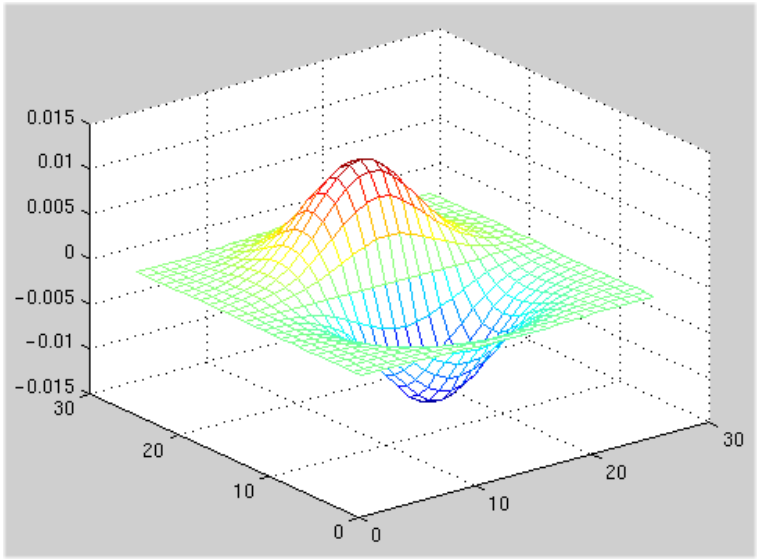
Odd
Gabor
filter

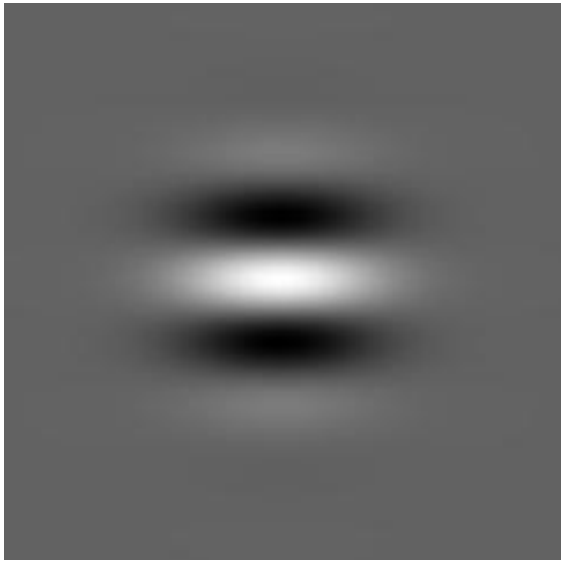


… looks a lot like…

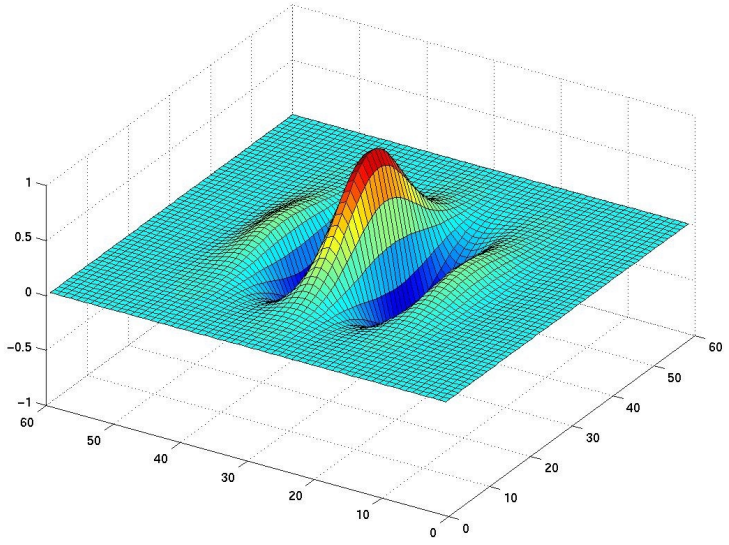


Gaussian
Derivative

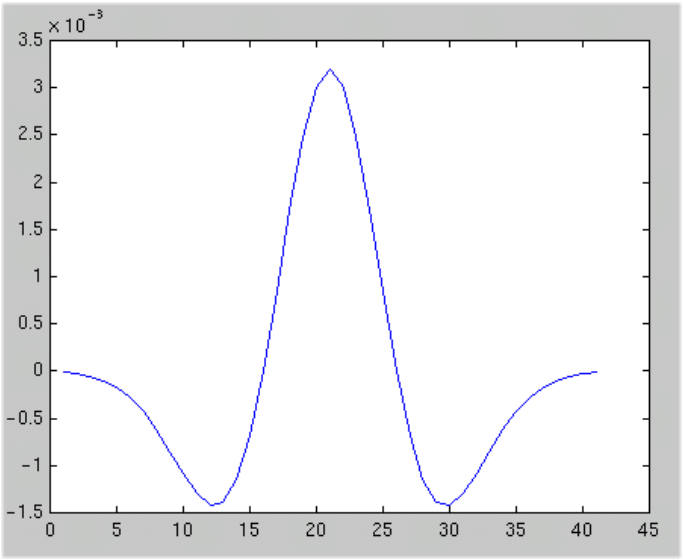




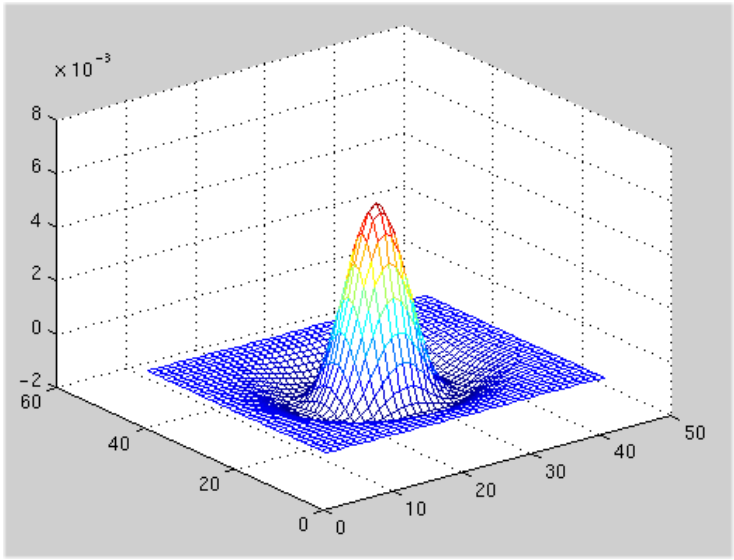
Even
Gabor
filter



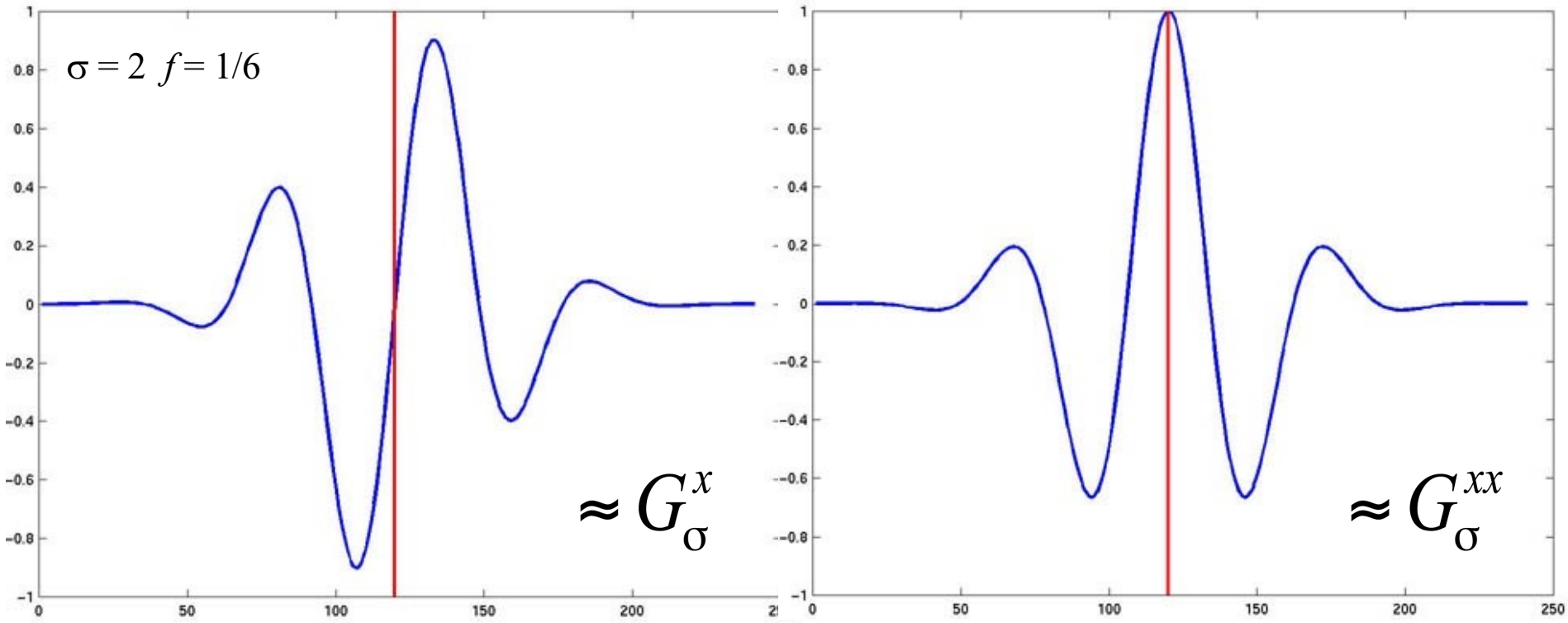
… looks a lot like…



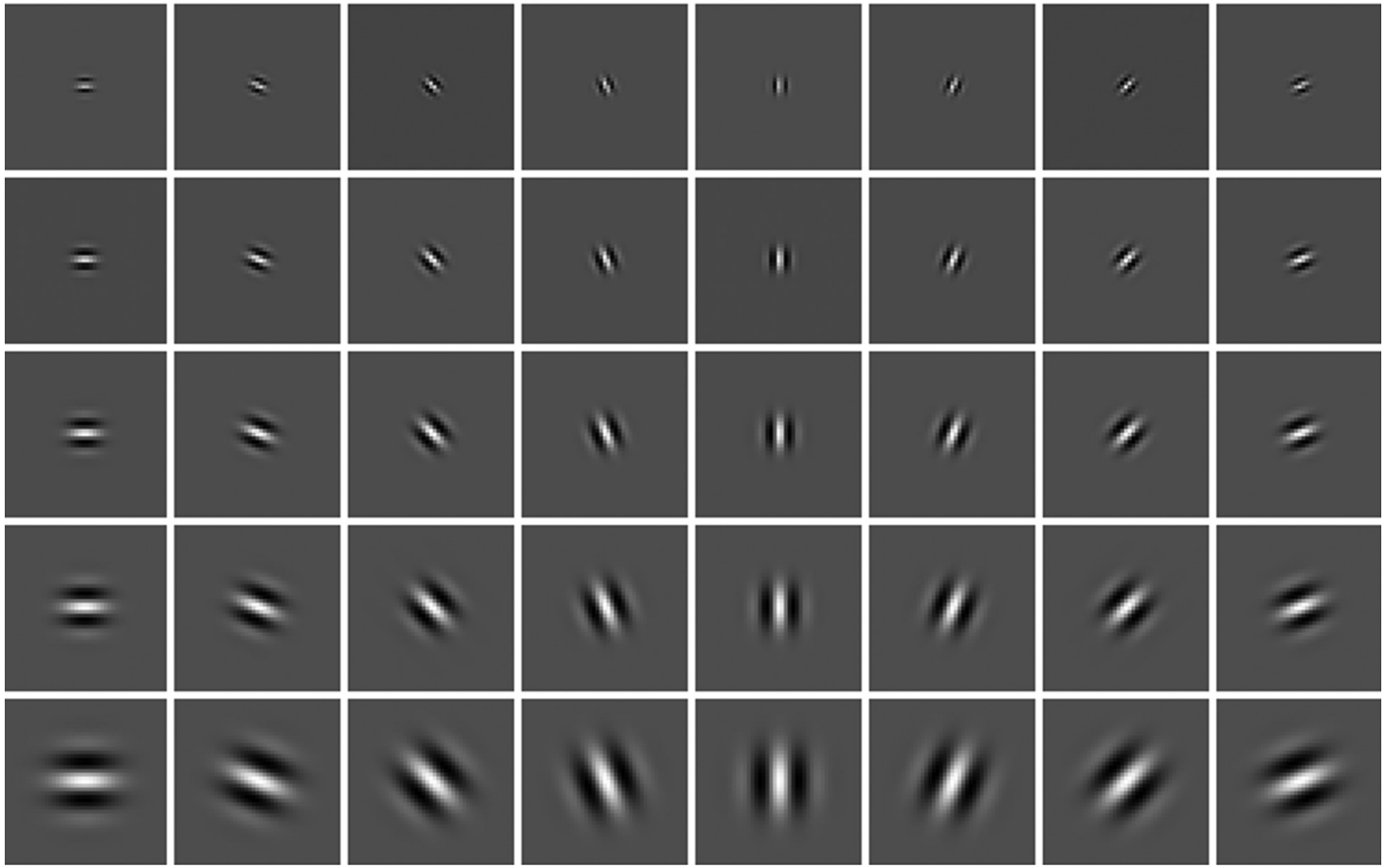
Laplacian



For small scales, the Gabor filters become derivative operators



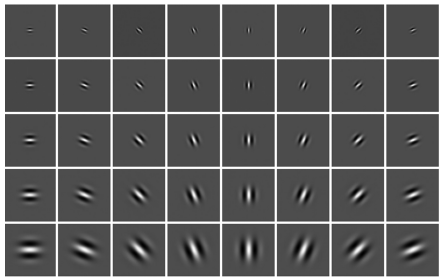
Directional edge detectors



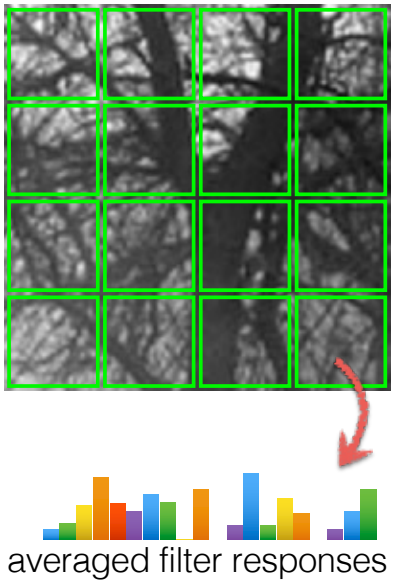
GIST

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

Filter bank



4 x 4 cell



averaged filter responses

What is the GIST descriptor encoding?

Rough spatial distribution of image gradients

SURF

SURF

(‘Speeded’ Up Robust Features)

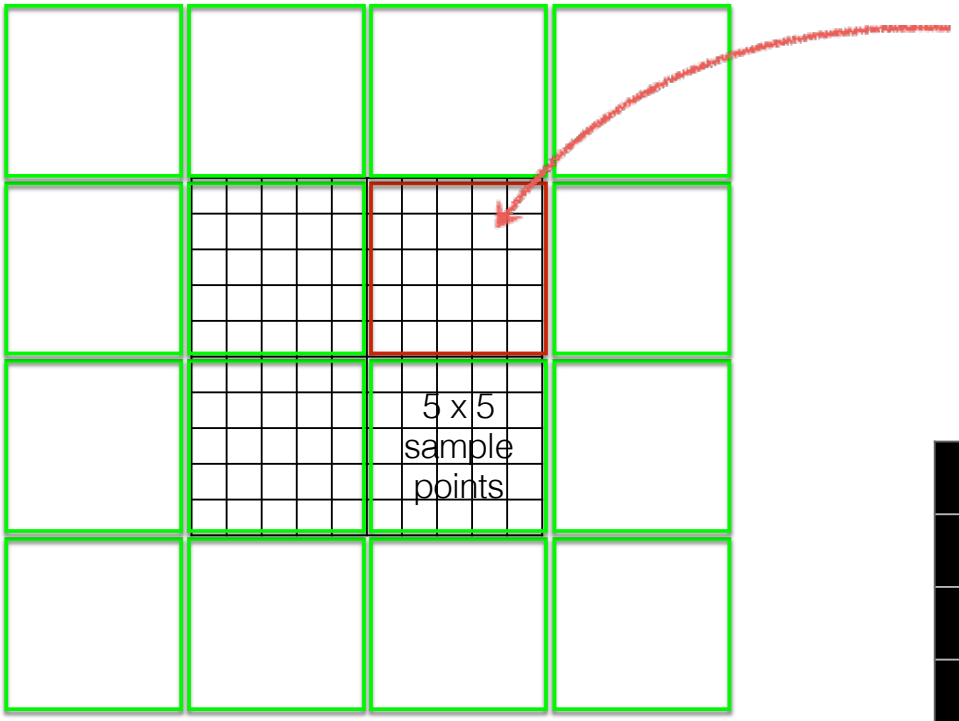
Compute Haar wavelet response at each pixel in patch

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool ECCV 2006

SURF

(‘Speeded’ Up Robust Features)

4 x 4 cell grid

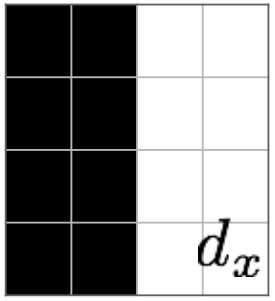


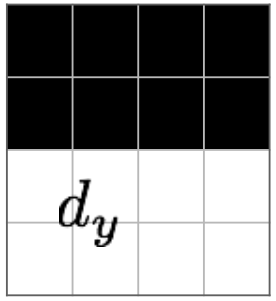
Each cell is represented by 4 values:

$$\left[\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right]$$

Haar wavelets filters

(Gaussian weighted from center)

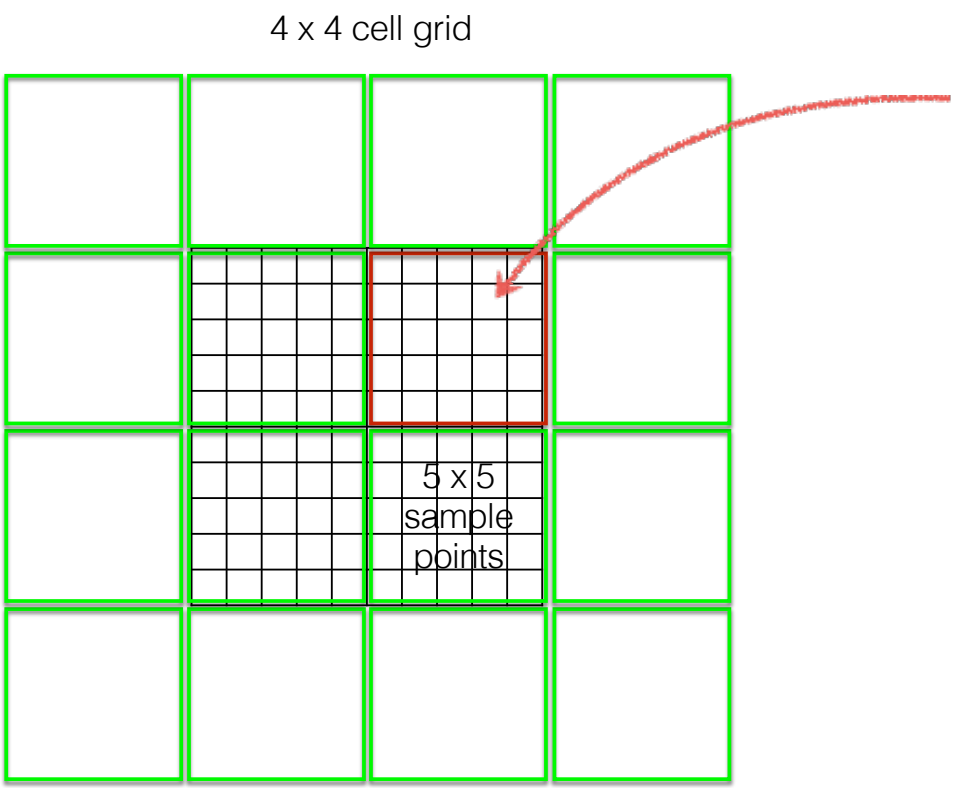




How big is the SURF descriptor?

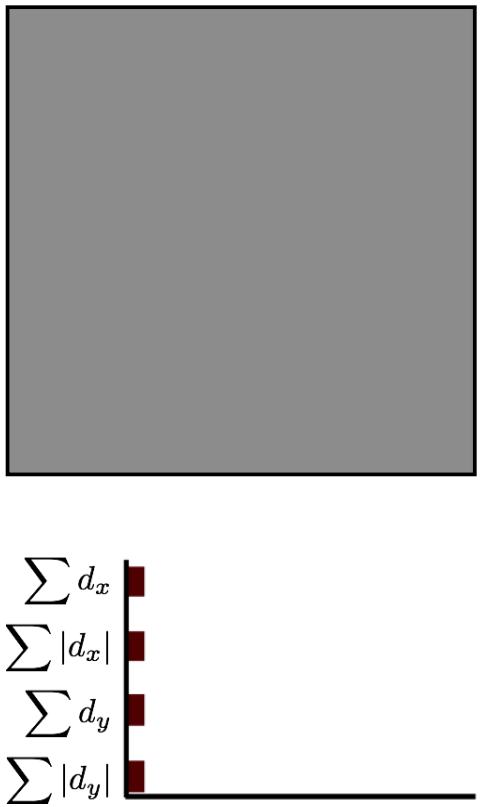
SURF

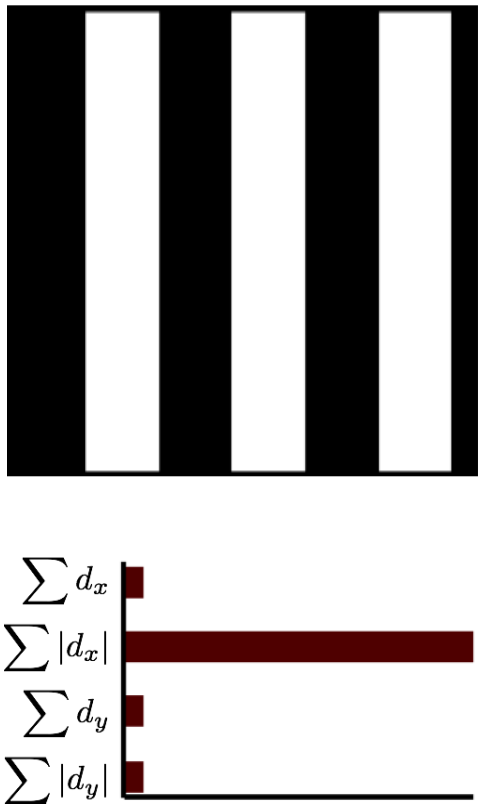
('Speeded' Up Robust Features)

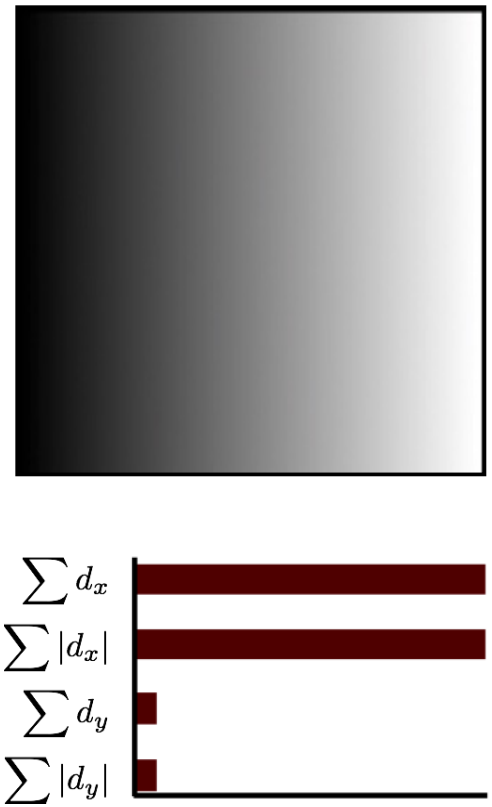


Each cell is represented by 4 values:

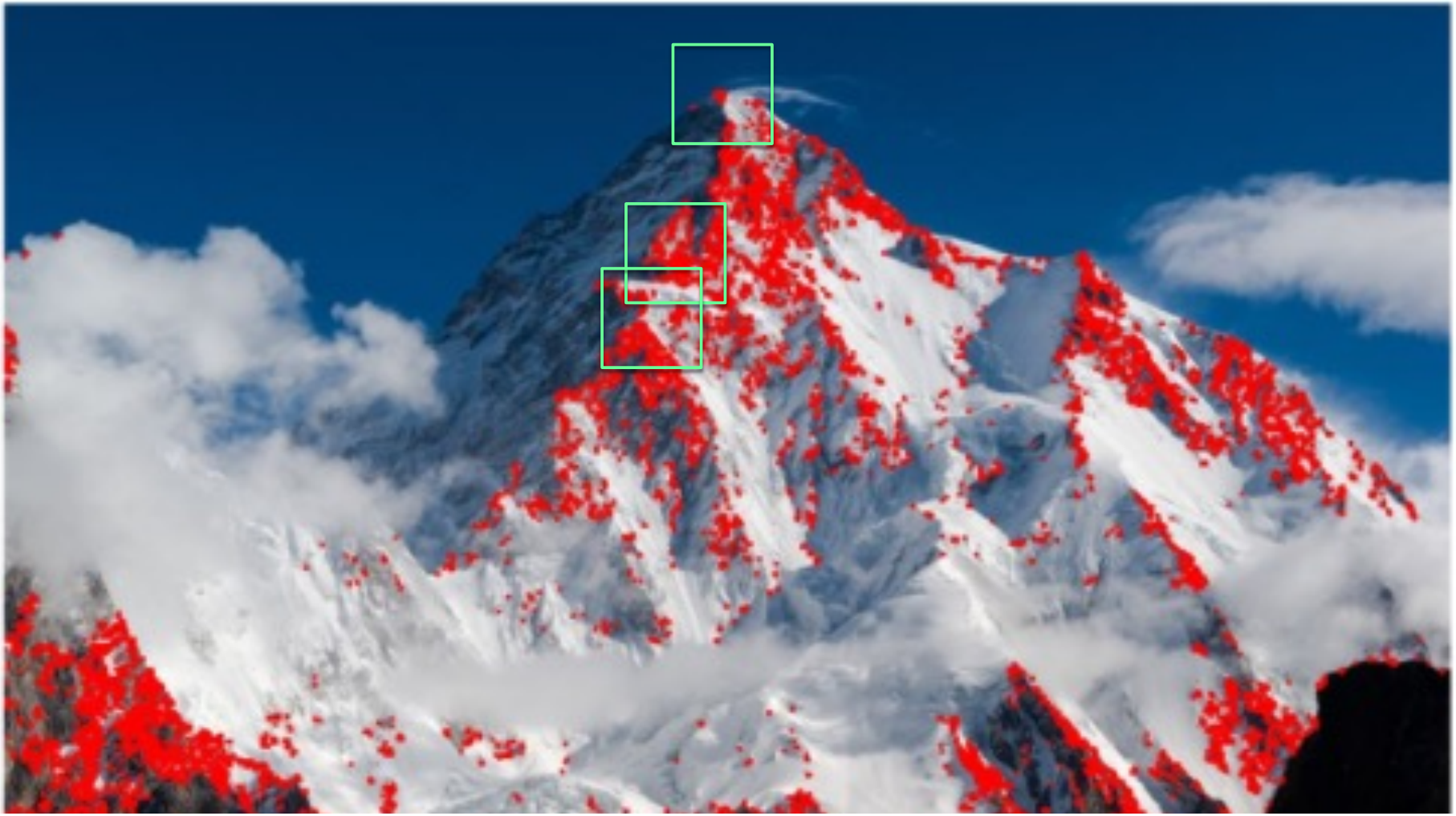
$$\left[\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right]$$


$\sum d_x$
$\sum |d_x|$
$\sum d_y$
$\sum |d_y|$


$\sum d_x$
$\sum |d_x|$
$\sum d_y$
$\sum |d_y|$


$\sum d_x$
$\sum |d_x|$
$\sum d_y$
$\sum |d_y|$

Amortize Computation



Integral Image

original image

$I(x, y)$

1	5	2
2	4	1
2	1	1

$A(x, y)$

1	6	8
3	12	15
5	15	19

integral image

$$A(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y')$$

Integral Image

original image

$I(x, y)$

1	5	2
2	4	1
2	1	1

$A(x, y)$

1	6	8
3	12	15
5	15	19

integral image

$$A(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y')$$

Can find the **sum** of any block using **3** operations

$$A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1)$$

What is the sum of the bottom right 2x2 square?

$$A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1)$$

$I(x, y)$

1	5	2
2	4	1
2	1	1

image

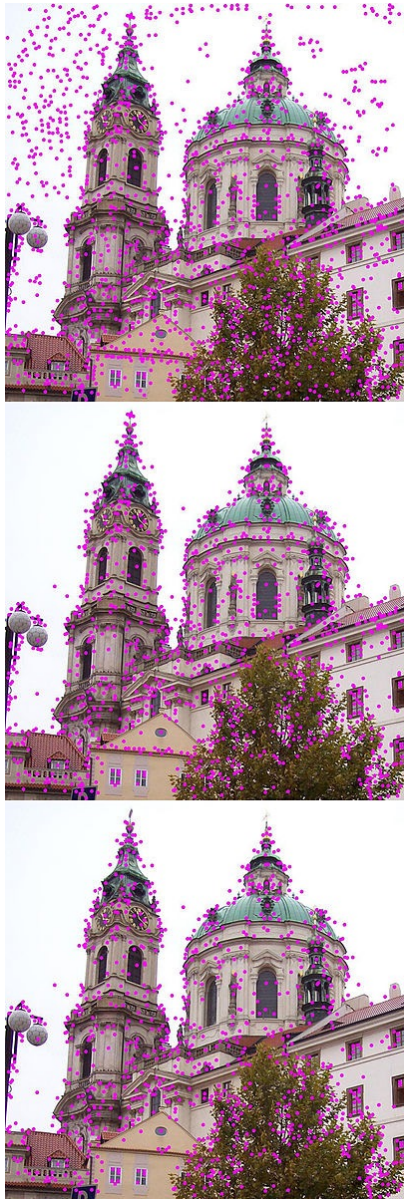
$A(x, y)$

1	6	**8**
3	12	15
5	15	**19**

integral image

$$\begin{aligned} A(1,1,3,3) &= A(3,3) - A(1,3) - A(3,1) + A(1,1) \\ &= 19 - 8 - 5 + 1 \\ &= 7 \end{aligned}$$

SIFT



SIFT

(Scale Invariant Feature Transform)

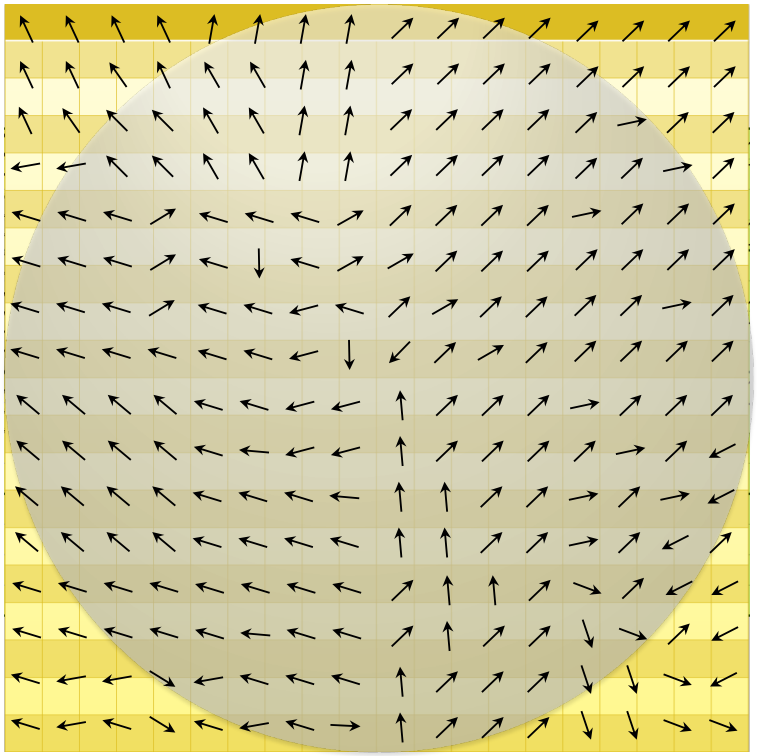
SIFT describes both a **detector** and **descriptor**

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

4. Keypoint descriptor

Image Gradients

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

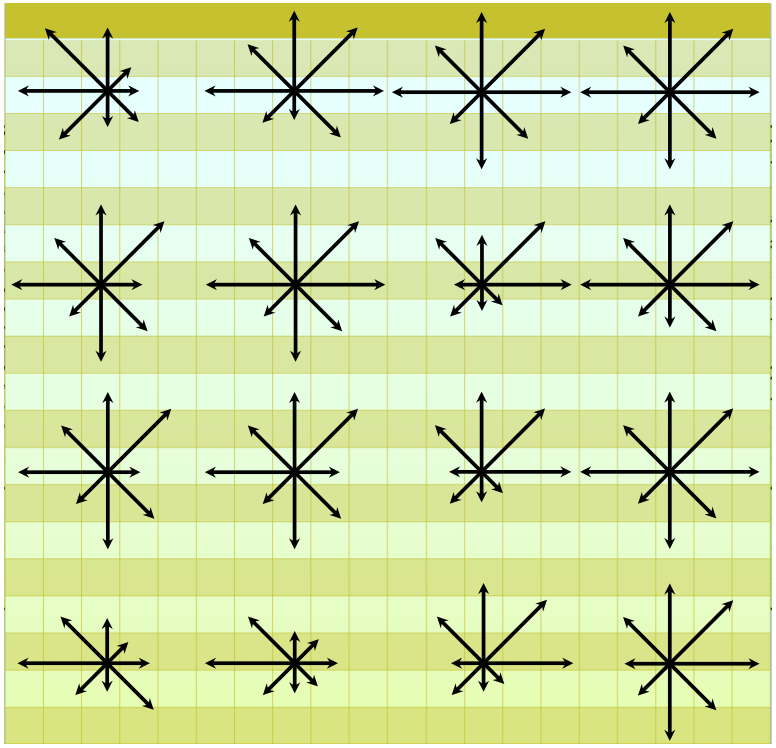


Gaussian weighting

(sigma = half width)

SIFT descriptor

(16 cells x 8 directions = 128 dims)



References

Basic reading:

- Szeliski textbook, Sections 4.1.2, 14.1.2.