Vision and Perception

Lecture 5: feature detectors and **descriptors**

SAPIENZA
UNIVERSITÀ DI ROMA

Overview

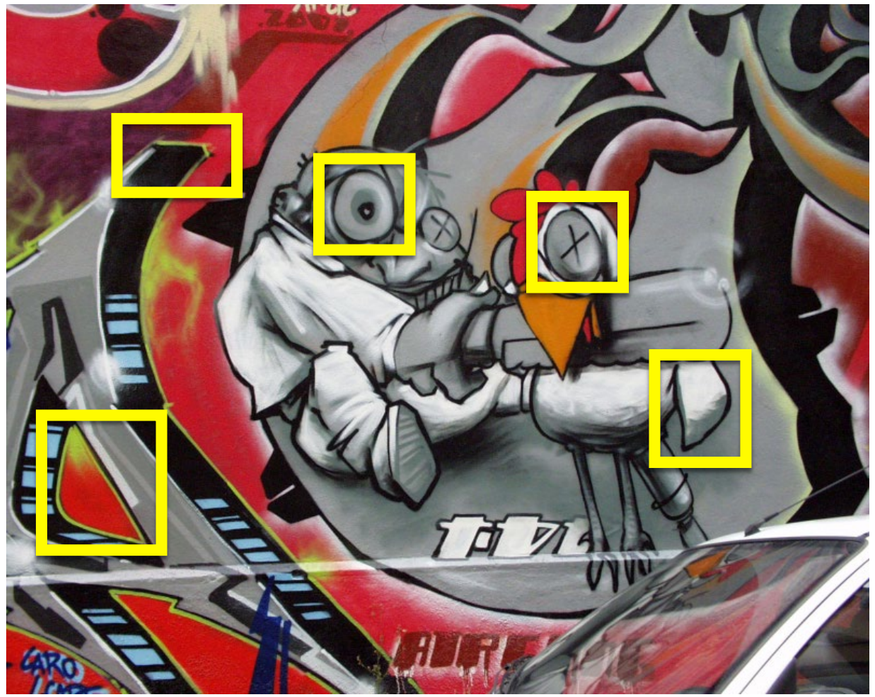
- Why do we need feature descriptors?
- Designing feature descriptors
- HOG descriptor
- SIFT

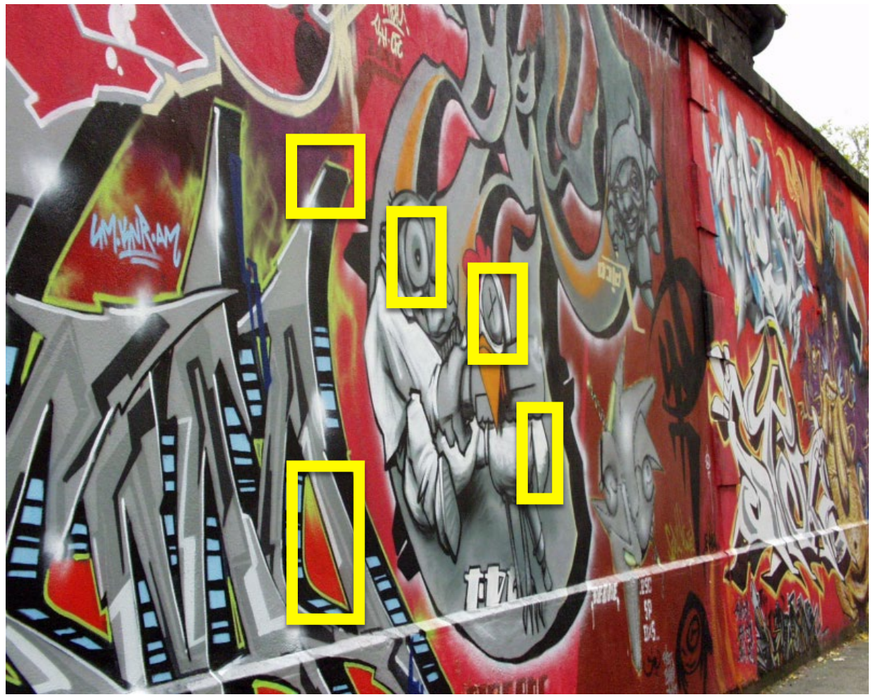
References

Basic reading:

- Szeliski textbook, Sections 7.1.2, 6.3.2, 7.1.3

Why do we need feature descriptors?





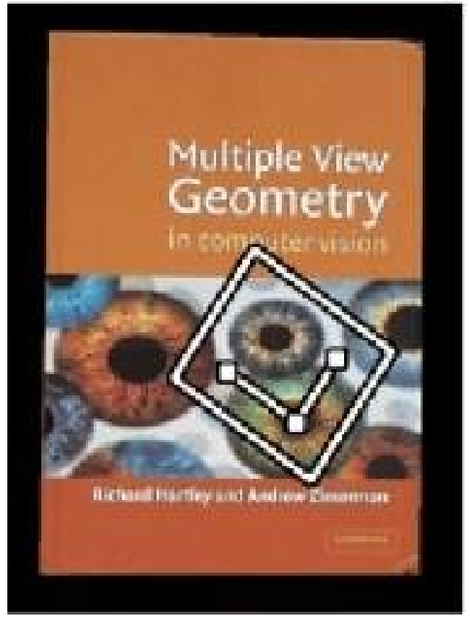
If we know where the <u>good</u> features are, how do we <u>match</u> them?

Designing feature descriptors

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

Photometric transformations

Geometric transformations

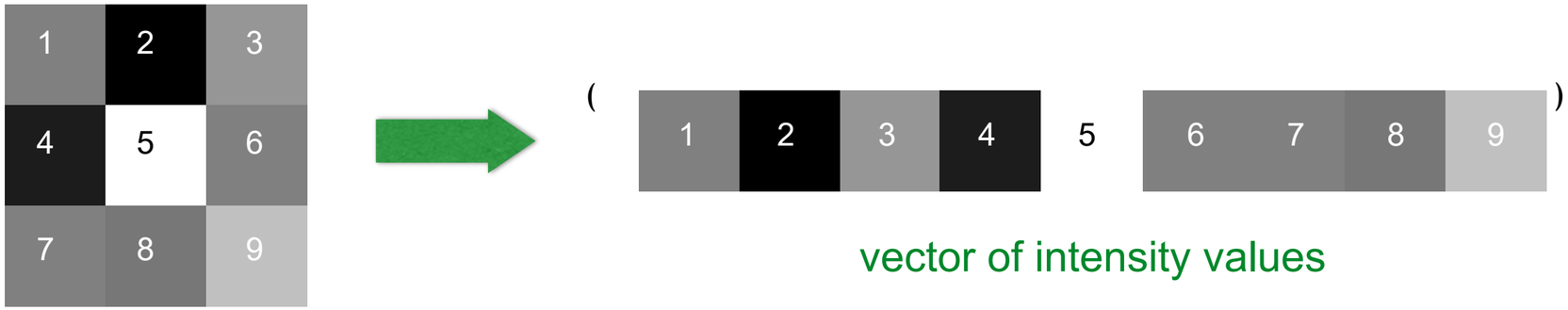


objects will appear at different scales,
translation and rotation

What is the best descriptor for an image feature?

Image patch

Just use the pixel values of the patch

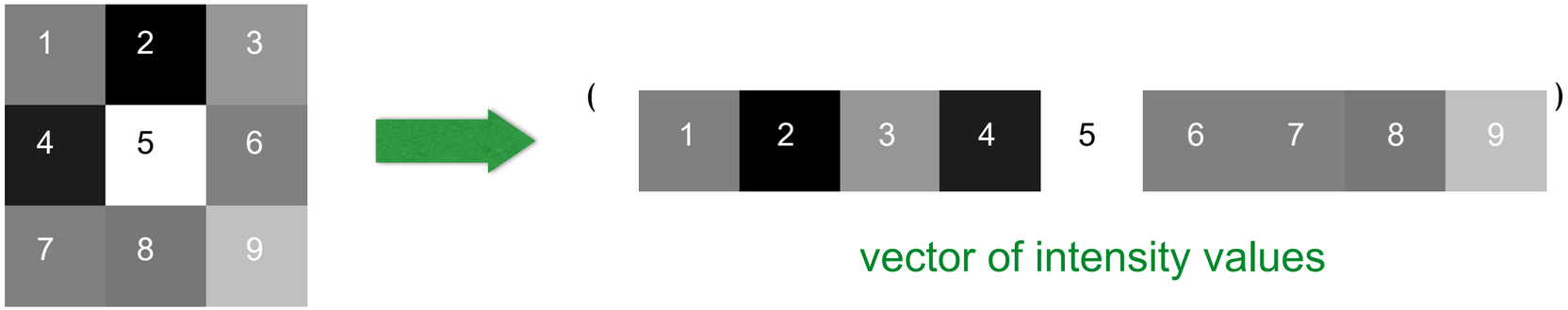


Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

What are the problems?

Image patch

Just use the pixel values of the patch



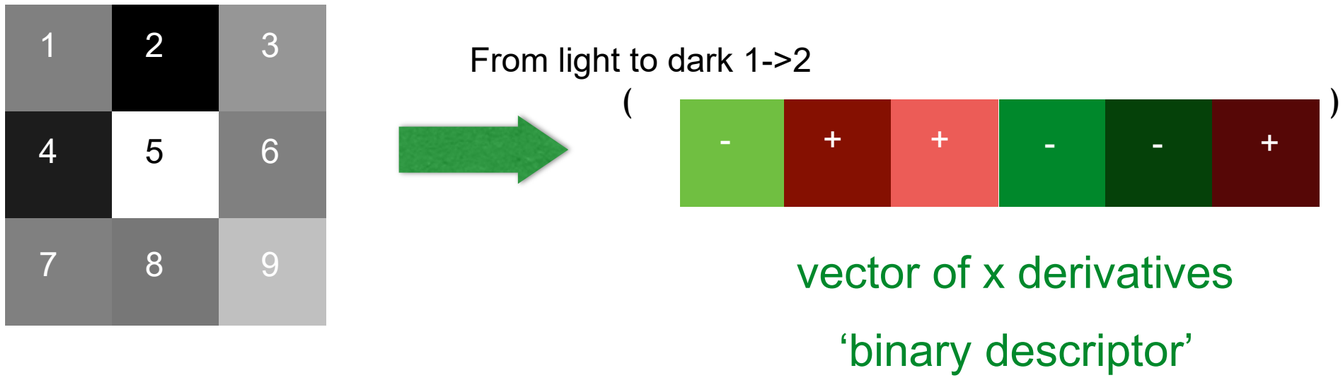
Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

What are the problems?

How can you be less sensitive to absolute intensity values?

Image gradients

Use pixel differences



vector of x derivatives

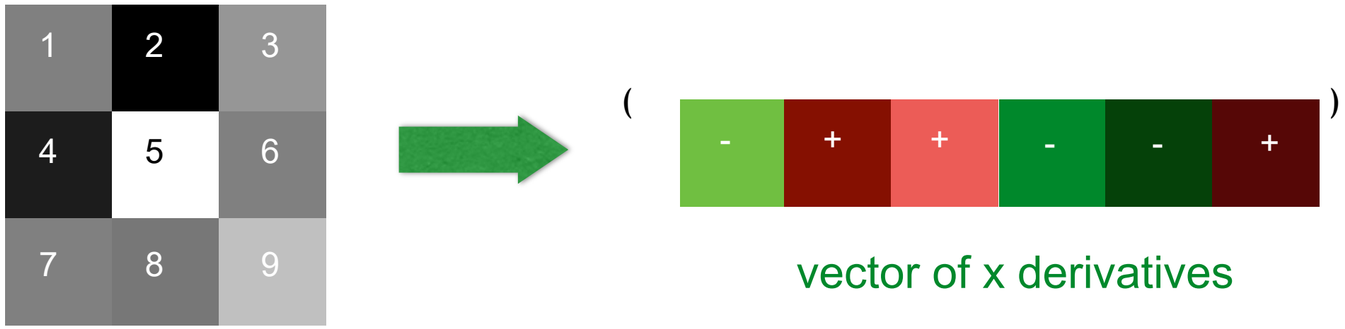
‘binary descriptor’

Intensity variation not the intensity values
Feature is invariant to absolute intensity values

What are the problems?

Image gradients

Use pixel differences



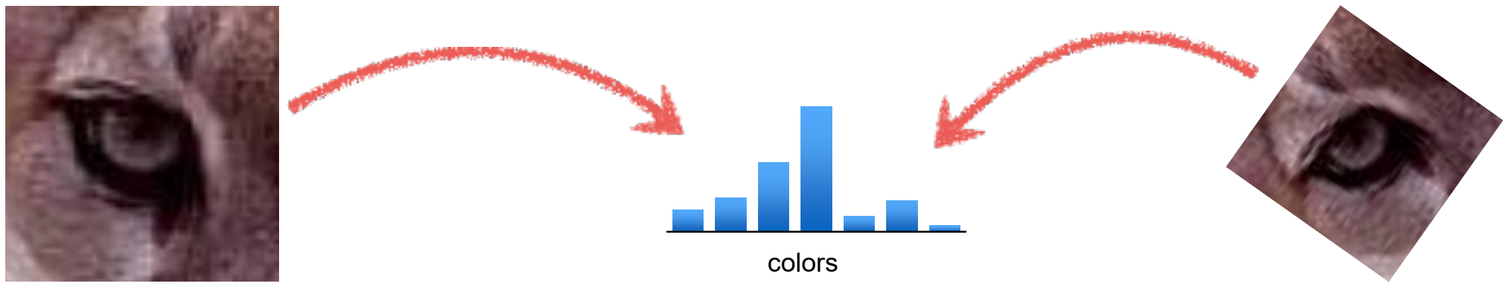
Feature is invariant to absolute intensity values

What are the problems?

How can you be less sensitive to deformations?

Color histogram

Count the colors in the image using a histogram

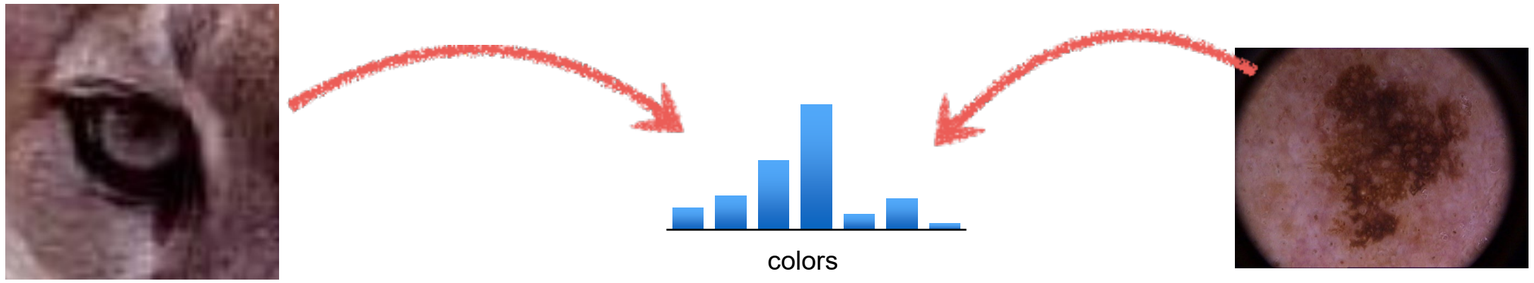


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram

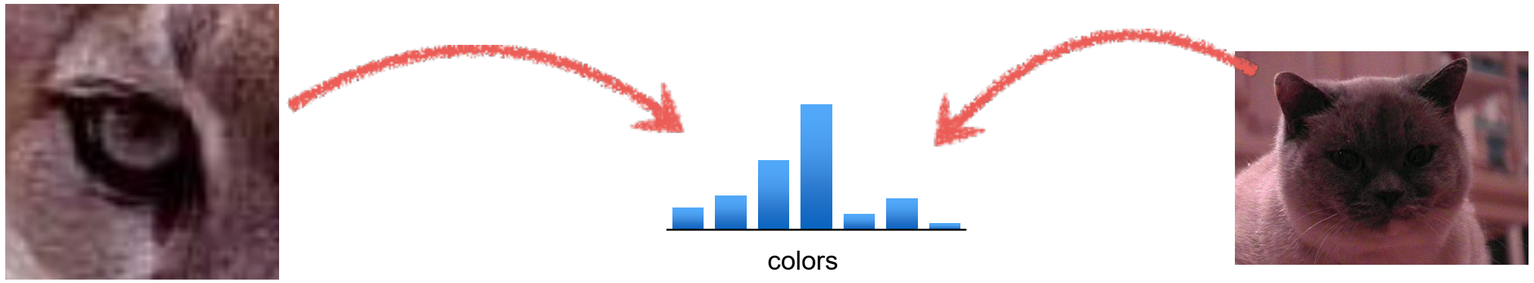


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram



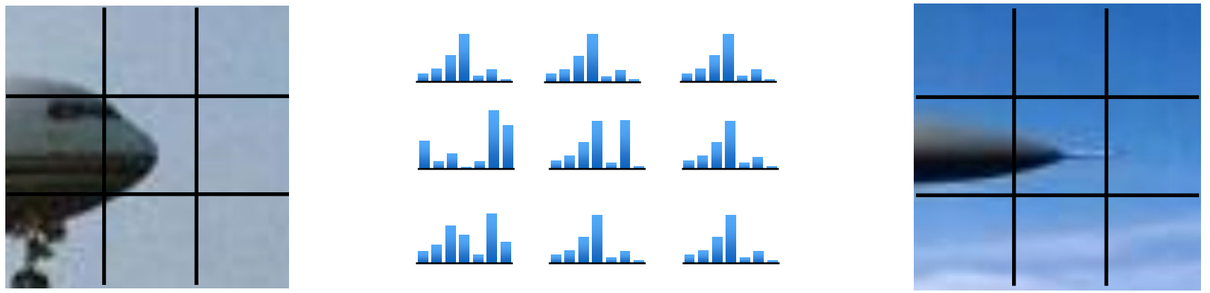
Invariant to changes in scale and rotation

What are the problems?

How can you be more sensitive to spatial layout?

Spatial histograms

Compute histograms over spatial ‘cells’



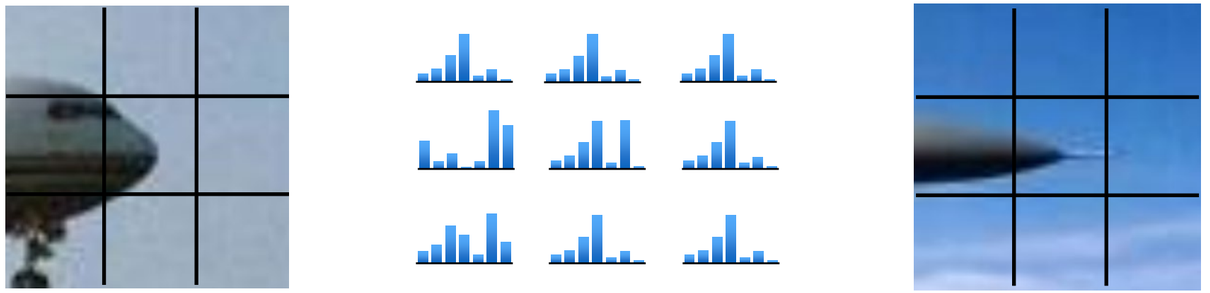
Retains rough spatial layout

Some invariance to deformations

What are the problems?

Spatial histograms

Compute histograms over spatial ‘cells’



Retains rough spatial layout

Some invariance to deformations

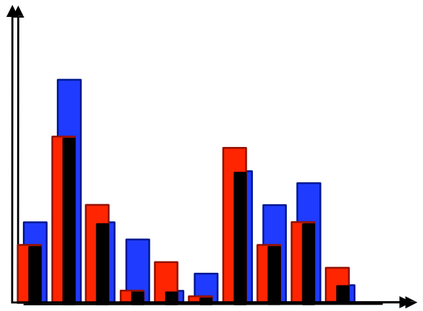
What are the problems?

How can you be completely invariant to rotation?

Histogram Comparison

- Comparison measures
  - Intersection

$$\cap(Q,V) = \sum_i \min(q_i, v_i)$$



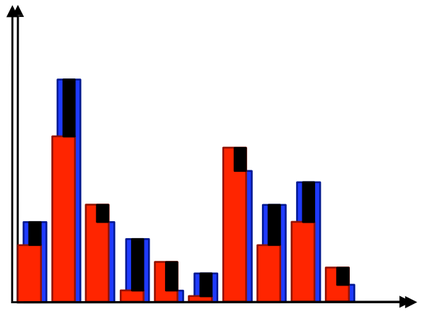
- Motivation
  - Measures the common part of both histograms
  - Range: [0,1]
  - For unnormalized histograms, use the following formula

$$\cap(Q,V) = \frac{1}{2}\left(\frac{\sum_i \min(q_i, v_i)}{\sum_i q_i} + \frac{\sum_i \min(q_i, v_i)}{\sum_i v_i}\right)$$

Histogram Comparison

- Comparison Measures
  - Euclidean Distance

$$d(Q,V) = \sum_i (q_i - v_i)^2$$

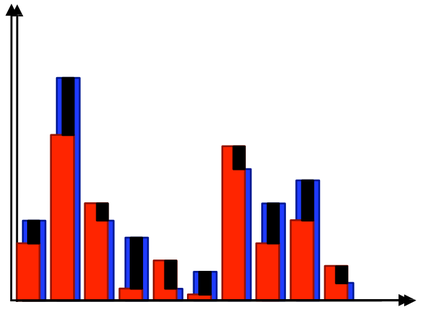


- Motivation
  - Focuses on the differences between the histograms
  - Range: [0,∞]
  - All cells are weighted equally. Not very discriminant

Histogram Comparison

- Comparison Measures
  - Chi-square

$$\chi^2(Q,V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$



- Motivation
  - Statistical background:
    - Test if two distributions are different
    - Possible to compute a significance score
  - Range: [0,∞]
  - Cells are not weighted equally!
    - therefore more discriminant
    - may have problems with outliers (therefore assume that each cell contains at least a minimum of samples)

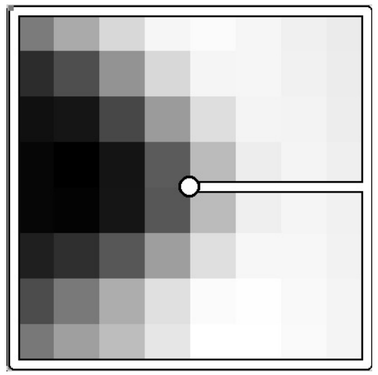
Histogram Comparison

- Which measure is best?
  - Depends on the application…
  - Both Intersection and χ^2 give good performance.
    - Intersection is a bit more robust.
    - χ^2 is a bit more discriminative.
  - Euclidean distance is not robust enough.

  There exist many other measures
    - e.g. statistical tests: Kolmogorov-Smirnov
    - e.g. information theoretic: Kullback-Leiber divergence, Jeffrey divergence, ...

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?

The pixel colors change with the illumination (“color constancy problem“)

- Intensity
- Spectral composition (illumination color)

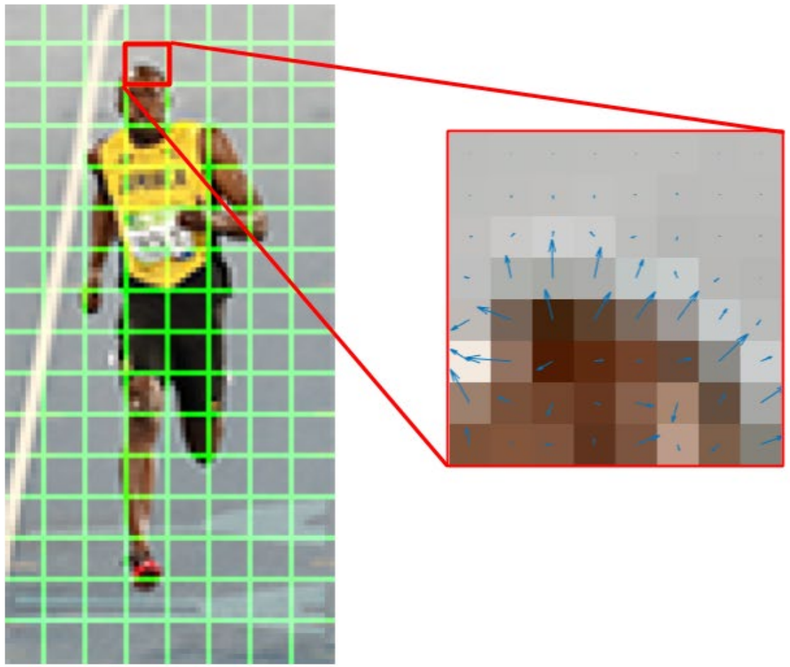
Not all objects can be identified by their color distribution.

HOG descriptor

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

HOG

- Preprocessing: patch aspect ratio of 1:2; this patch is cropped out of an image and resized to 64×128
- First calculate the horizontal and vertical gradients
- Next, we can find the magnitude and direction of gradient
- The image is divided into 8×8 cells and a histogram of gradients is calculated for each 8×8 cells.



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

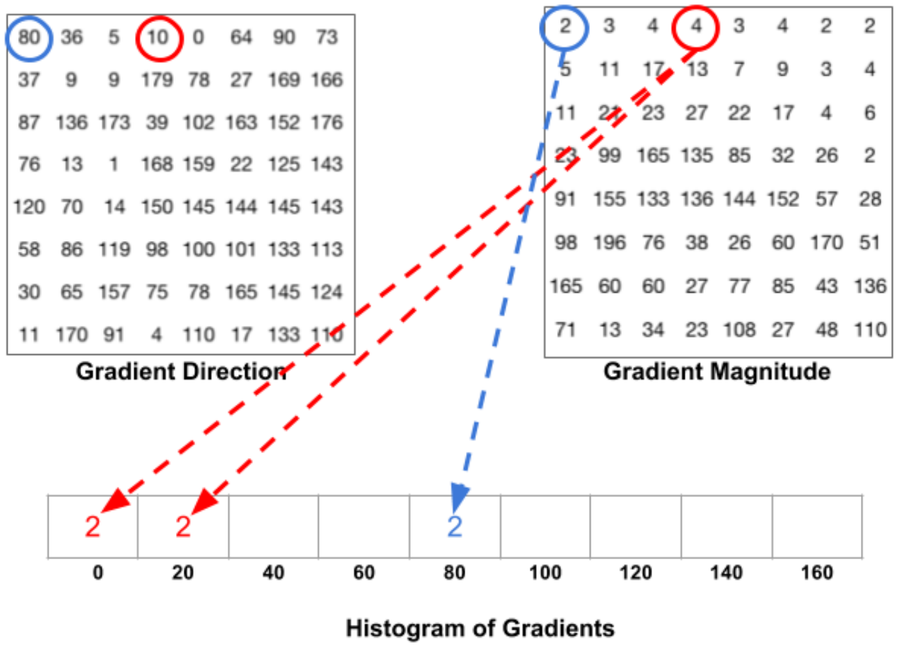
Gradient Direction

The angles are between 0 and 180 degrees instead of 0 to 360 degrees.
These are called “unsigned” gradients because a gradient and it’s negative are represented by the same numbers.

HOG

The next step is to create a histogram of gradients in these 8×8 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40 … 160.

- A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude.

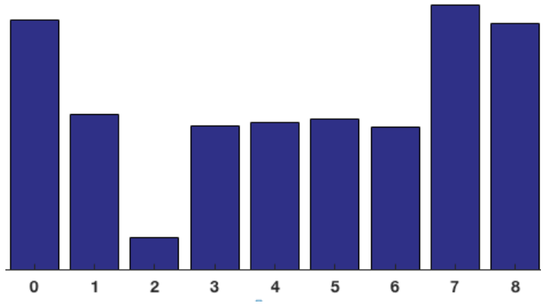


The gradient at the pixel encircled in red has an angle of 10 degrees and magnitude of 4.

Since 10 degrees is half way between 0 and 20, the vote by the pixel is splitted evenly into the two bins (the contribution is proportional).

HOG

The contributions of all the pixels in the 8×8 cells are added up to create the 9-bin histogram.

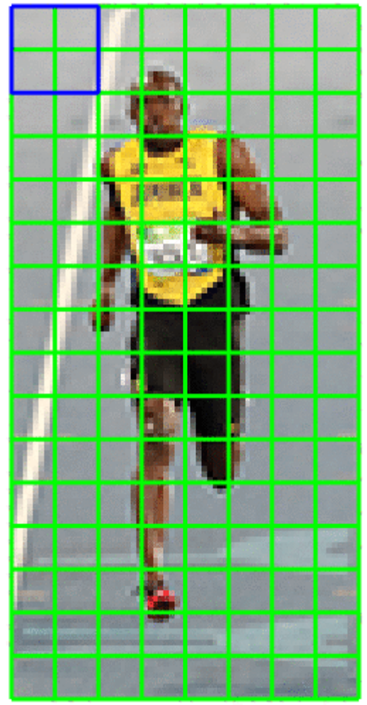


- 16×16 Block Normalization is used to normalize the histogram with L2 norm. In this way it is not affected by lighting variations.
- A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector.

- The window is then moved by 8 pixels and a normalized 36×1 vector is calculated over this window and the process is repeated for all the

On the n-dimensional Euclidean space $\mathbb{R}^n$, the intuitive notion of length of the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is captured by the formula

$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_n^2}.$$ [7]

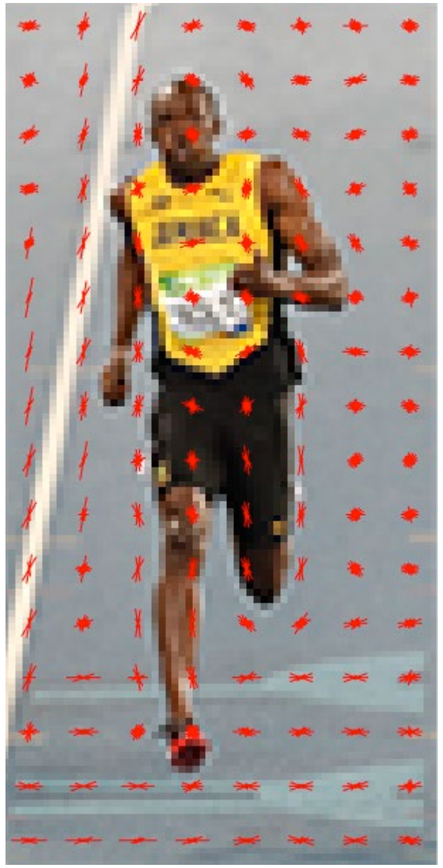


HOG

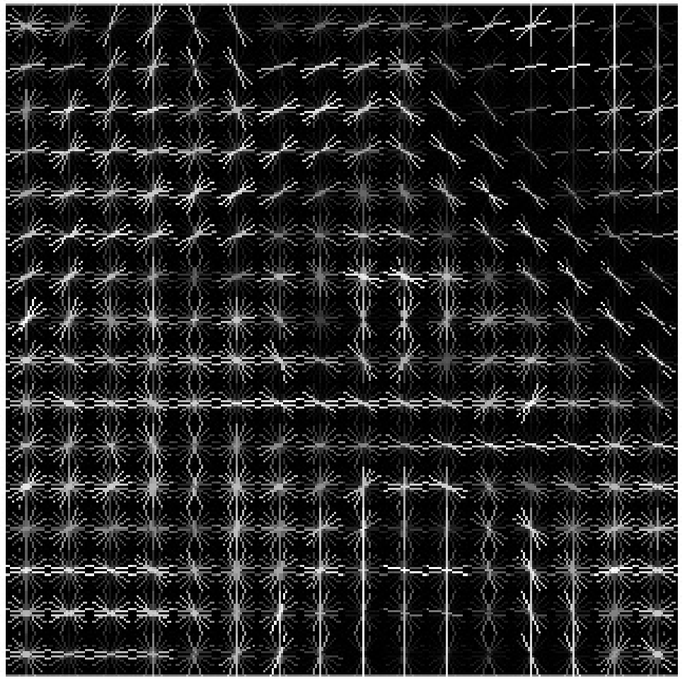
Calculate the HOG feature vector: 3780x1 dimensional vector.

The HOG descriptor of an image patch is usually visualized by plotting the 9×1 normalized histograms in the 8×8 cells.

Notice that dominant direction of the histogram captures the shape of the person, especially around the torso and legs.



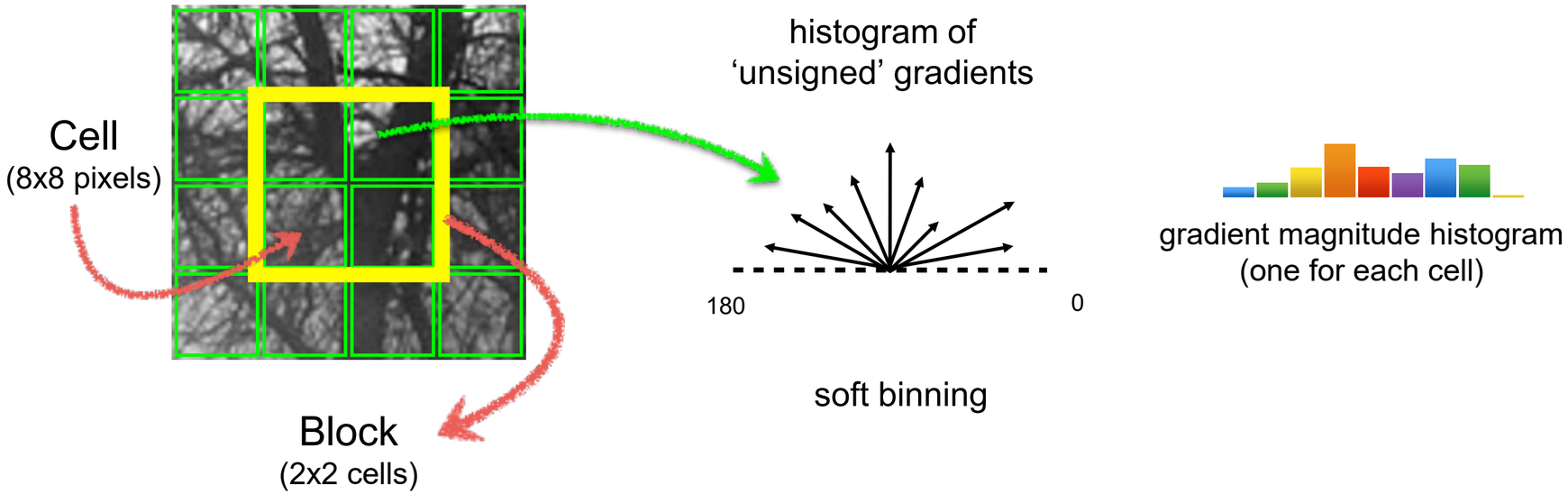
Example



Standard HOG features with a cell size of eight pixels.

HOG

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



Concatenate and L-2 normalization

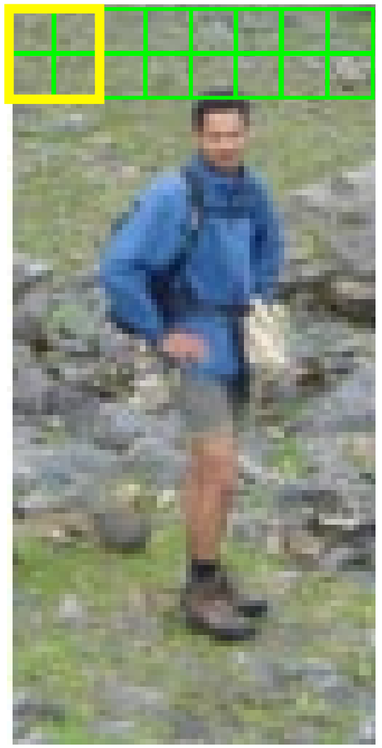
Single scale, no dominant orientation

Pedestrian detection

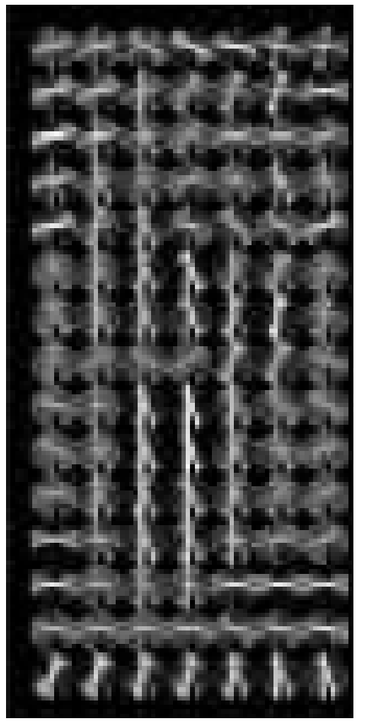
1 cell step size

visualization

128 pixels
16 cells
15 blocks



15 x 7 x 36 (9x4)
= 3780



64 pixels
8 cells
7 blocks

Redundant representation due to overlapping blocks

How many times is each inner cell encoded?

The corner cells appear once, the other edge cells appear twice each, and the interior cells appear four times each

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

Pedestrian detection

- Generate the feature vector for a given image
- After training the SVM over thousands of such HOG feature sets, we are ready to detect pedestrians
- After obtaining a feature vector for a given image, a prediction is produced in order to detect whether this feature vector is of a pedestrian or not. A simple 1/0 binary classification will suffice.

N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA, 2005, pp. 886–893 vol. 1

Acknowledgement: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei, Justin Johnson, Serena Yeung, R. Szelisky, Fabio Galasso, Ioannis Gkioulekas, Satya Mallick