DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Descriptors:

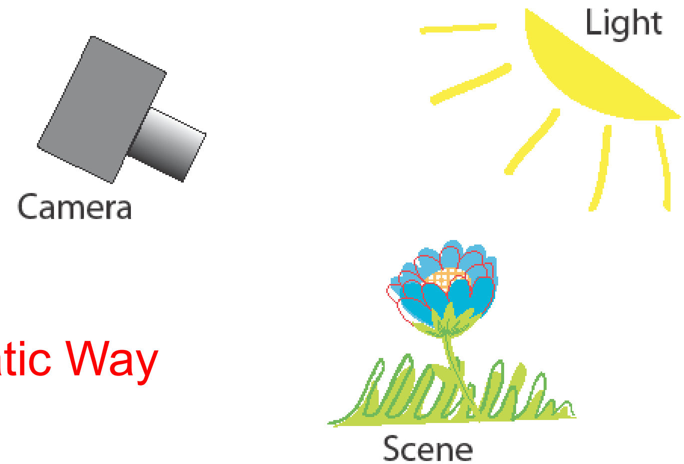
Detecting Interest Points, attributing them with descriptors and and matching them

François Lauze

Plan for today

- Detector comparison
- Descriptor construction, SIFT
- Matching
- High level descriptors

So which detectors are the best?

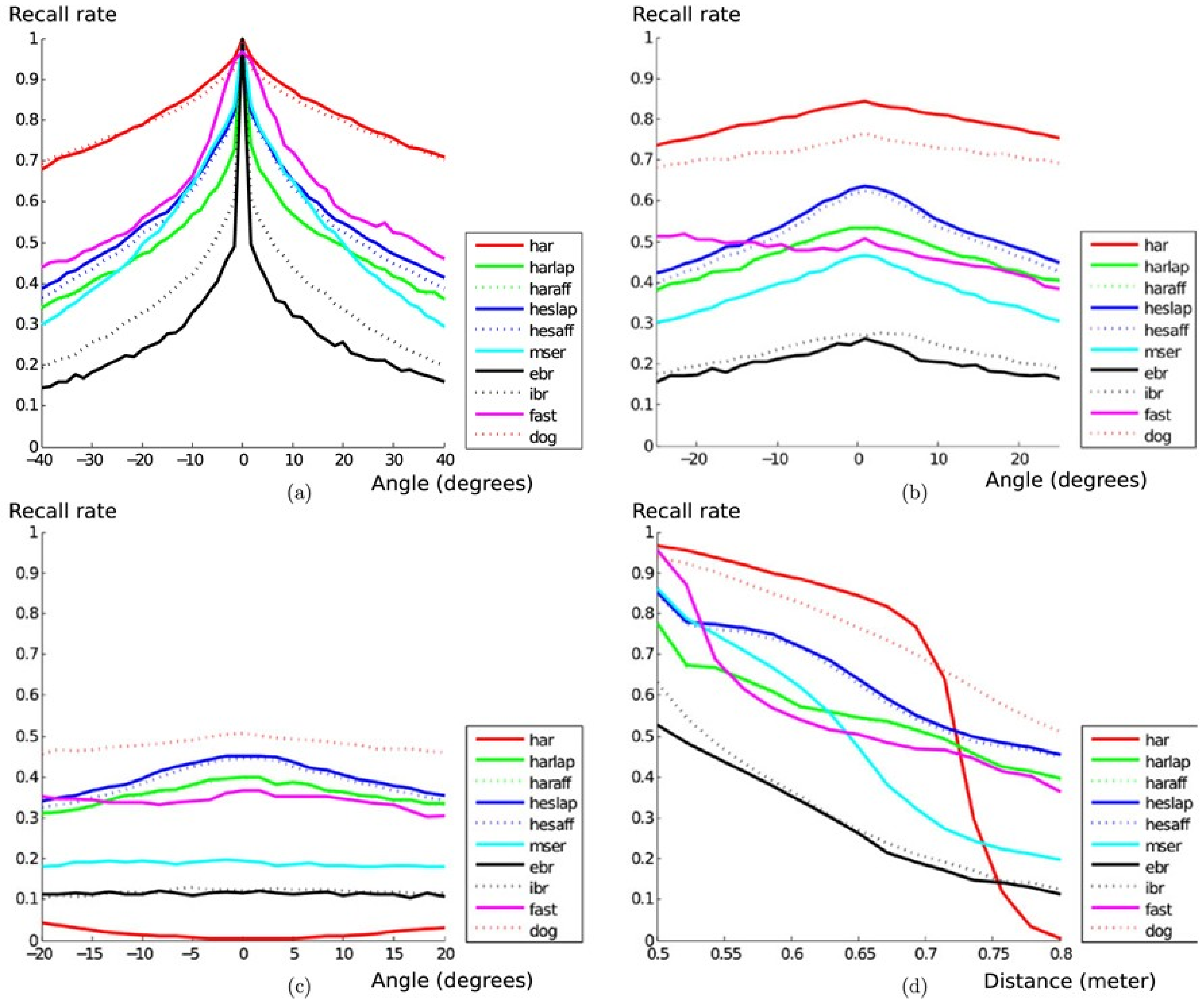


We Want to Control it All in a Systematic Way

$$\text{Recall} = \frac{\text{Potential Matches}}{\text{Total Interest Points}}$$

Kim S. Pedersen DIKU, Anders Dahl and Henrik Aanæs from DTU
Interesting Interest Points.
International Journal of Computer Vision, 97:18 – 35, 2012

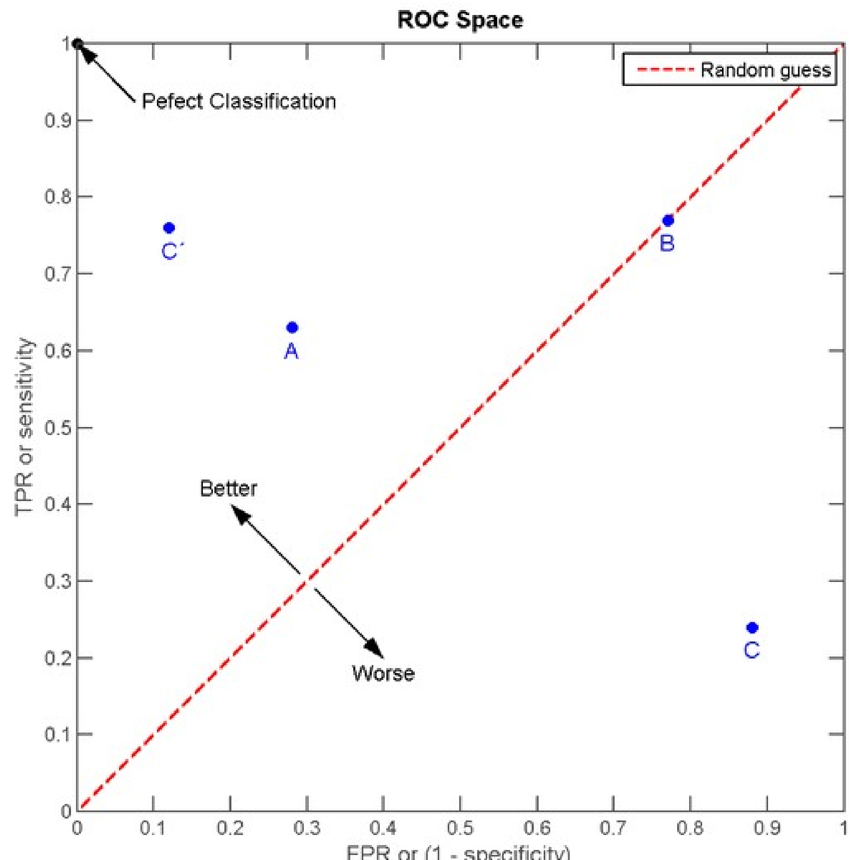
Effect of position



Aside: ROC and AUC?

- Receiver operating characteristic (ROC):
  - TPR = TP / P (Recall)
  - FPR = FP / N
- Area under the ROC curve (AUC):
  - AUC close to 1 is good
  - The probability of a correct match

		actual value		
		p	n	total
prediction outcome	p'	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
	total	P	N	



From Wikipedia 2020

Evaluation results

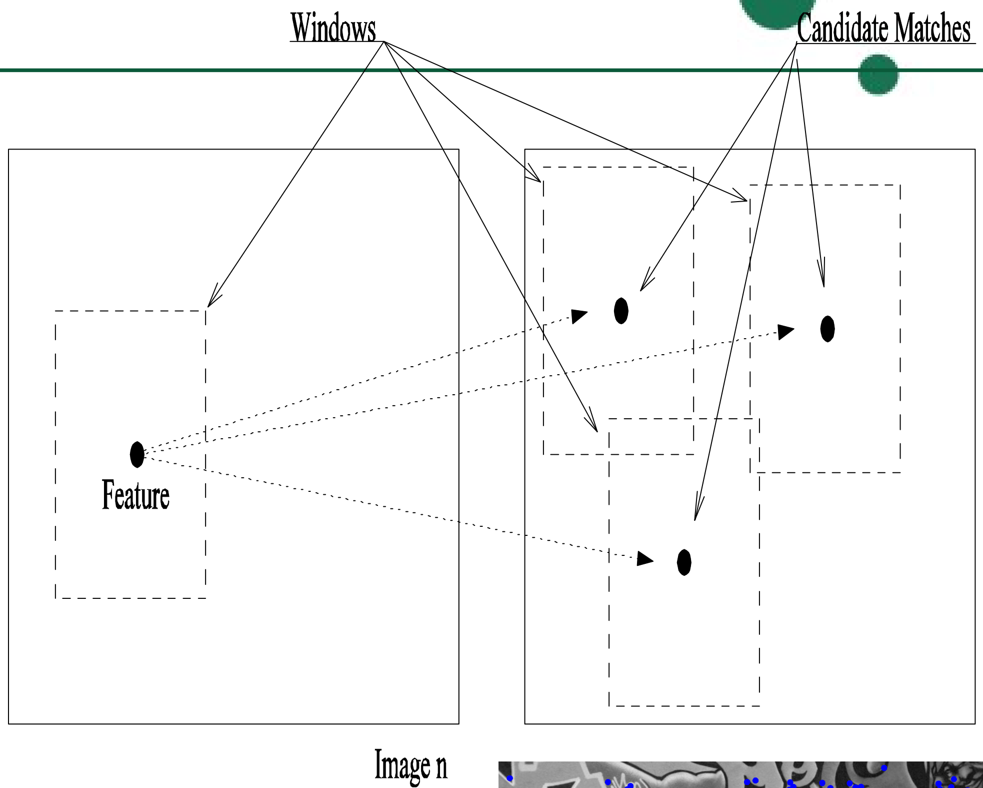
- Light is more disruptive than position (angle & scale).
- Many different evaluations on different databases
- Best Performers:
  - Harris Variants
  - DoG (SIFT Blobs)
- Less well performers:
  - MSER, FAST, Hessian Laplace Variants, etc.

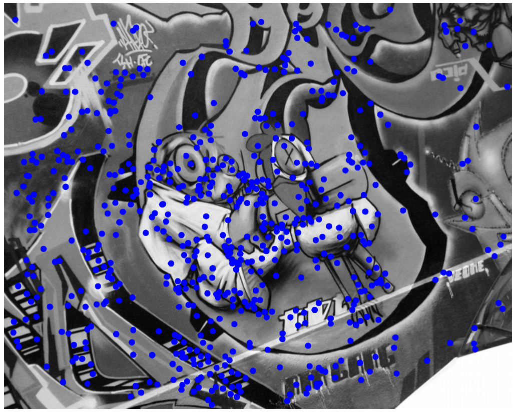
Matching Strategy Illustrated

1. Extract **interest points:** Harris corners, DoG (blobs)

2. Compute feature **descriptors**: Raw patches, , SIFT,…

3. **Match** points by pairing similar descriptors

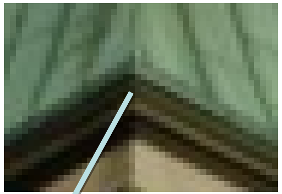
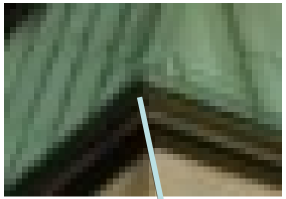


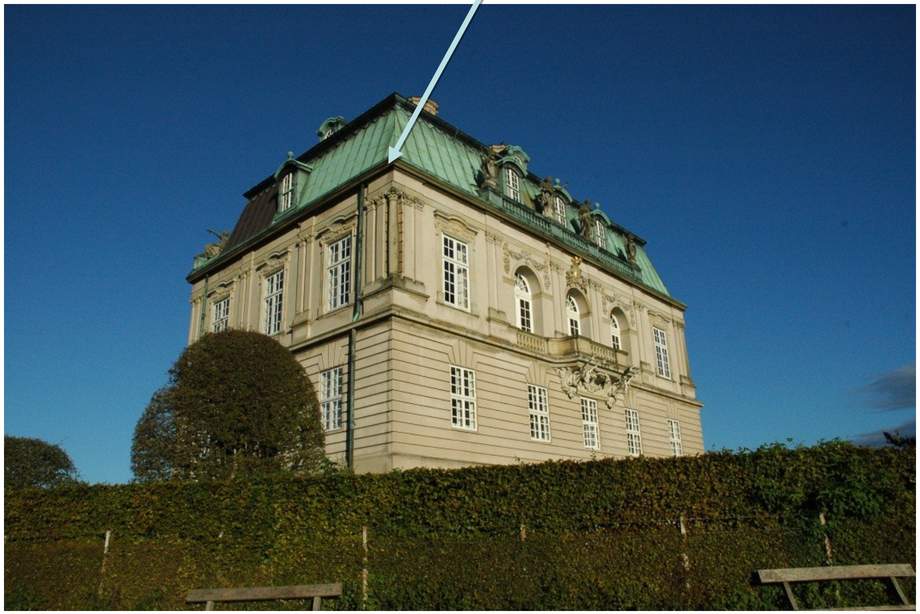


What do we wish of a descriptor

- We have to attribute each feature point with a descriptor in order to compare if the candidate match is good or not.
- Descriptors must be local, informative, insensitive to noise, luminance variations, perspective deformation including rotation, to pixel quantization etc.
- We like compact (low dimensional) descriptors but recognize that high dimensional ones may be more discriminative.

Matching patches

F_1  $\overset{?}{=}$  F_2



Using raw pixel patches as descriptors

- Represent a patch with the raw pixels – the patch is the descriptor.
- Compare patches pixel by pixel, e.g. by

$$d_1(x,y) = \sum_x \sum_y |F_1(x,y) - F_2(x,y)| \quad L^1\text{-norm}$$

$$d_2(x,y) = \sqrt{\sum_x \sum_y (F_1(x,y) - F_2(x,y))^2} \quad L^2\text{-norm}$$

  - (For simplicity lets assume gray scale intensities and the sums are over all pixels in the patches.)
    (Patches must be of equal size)
- This is referred to as either distances or dissimilarities

Using raw pixel patches as descriptors

- We need to compensate for changes in illumination conditions, because …
- Contrast change *a* and change in brightness level *c* (affine model):

$$F' = aF + b$$

- We want *F'* and *F* to have zero distance, but

$$d_1(F, F') \neq 0$$
$$d_2(F, F') \neq 0$$

- Normalisation: can I find a (intensity) affine invariant change?

$$F' = aF + b \implies F = \frac{F' - b}{a}$$

Matching patches: Normalized cross correlation

- Compute mean intensity and standard deviation for each patch

$$\bar{F}_1 = \frac{1}{n}\sum_{x,y} F_1(x,y), \bar{F}_2 = \frac{1}{n}\sum_{x,y} F_2(x,y)$$

$$\sigma_1^2 = \frac{1}{n}\sum_{x,y}\left(F_1(x,y) - \bar{F}_1\right)^2, \sigma_2^2 = \frac{1}{n}\sum_{x,y}\left(F_2(x,y) - \bar{F}_2\right)^2$$

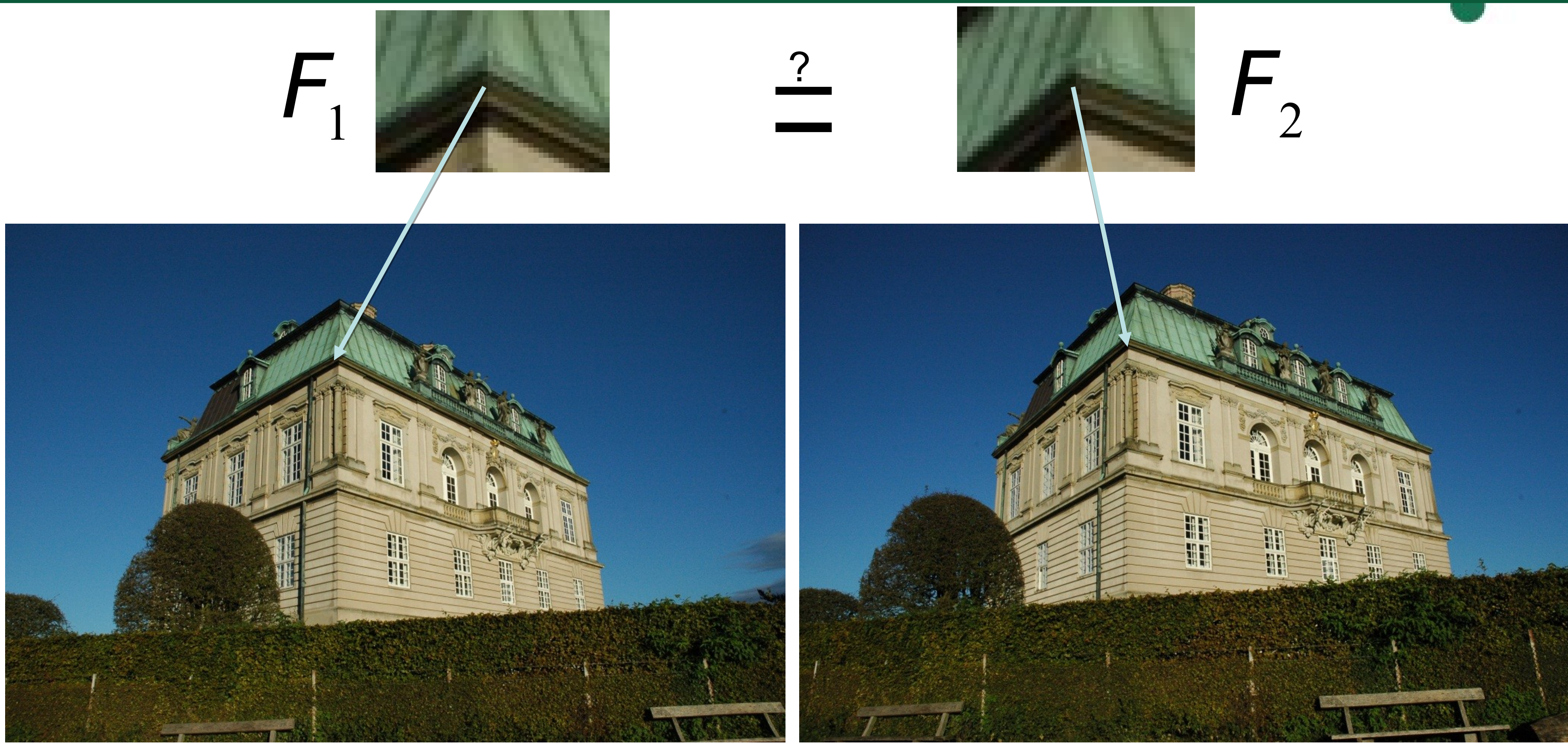
- (sum over all *n* pixels in patches)
- Measure distance with normalized cross correlation

$$\mathrm{NCCD} = 1 - \frac{1}{n}\frac{\sum_x \sum_y \left(F_1(x,y) - \bar{F}_1\right)\left(F_2(x,y) - \bar{F}_2\right)}{\sigma_1 \sigma_2}$$

- Affine intensity invariance:

$$F' = aF + b, a > 0, \implies \mathrm{NCCD}(F, F') = 0$$

Matching patches



Raw pixel descriptor: Use pixels in patch and compare with Normalized Cross Correlation

Open problems

- What patch size should we use?
  - Use the detection scale and resample so both patches have equal size in pixels
- Is this approach robust to scale changes in the scene?
  - Yes, if we do the resampling (see above)
- Is this approach robust to rotation in the scene?
  - No, this will lead to large dissimilarities
- Is it robust to perspective distortions?
  - No, this will lead to large dissimilarities

Scale Invariant Feature Transform (SIFT)

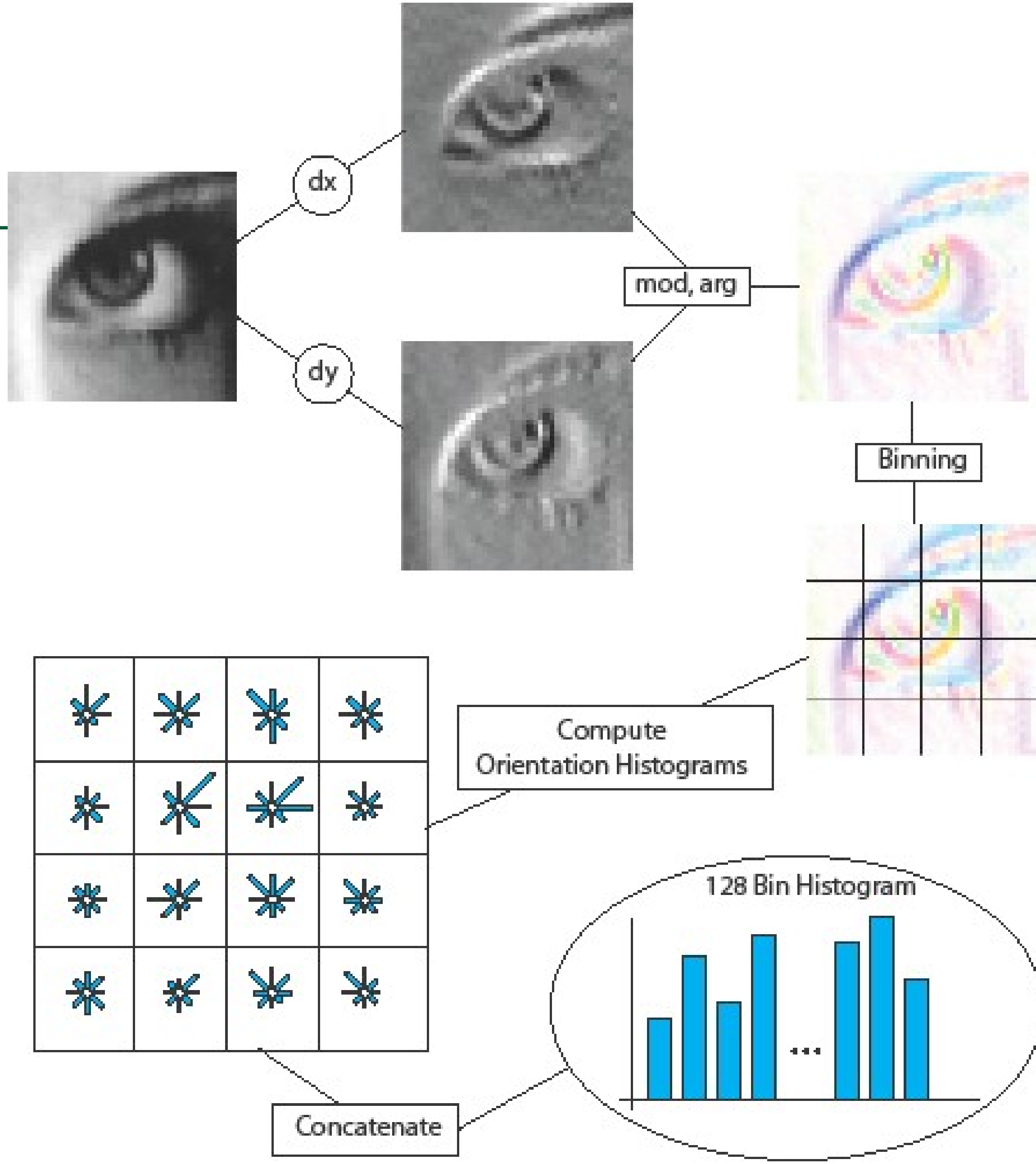
SIFT is a very popular descriptor
(Google. Scholar says 64.578 citations Dec 5. 2021 – 5000 more than a year ago...)

Scale invariance:

- This is obtained by using the DoG blob detector which is multi-scale. Descriptor build at these interest points in scale-space.
- After detection we have an interest point at $(\tilde{x}, \tilde{y}, \tilde{\sigma})$

- Rotational invariance:
  - Estimate an orientation of the interest point and build the descriptor relative to this.
- Translational invariance:
  - To some extend by construction of the descriptor (more on this)
- Illumination invariance:
  - By construction of the descriptor (more on this)

SIFT



dx
dy
mod, arg
Binning
Compute Orientation Histograms
128 Bin Histogram
...
Concatenate

SIFT: Rotational invariance by orientation assignment

- At detection scale compute image gradients

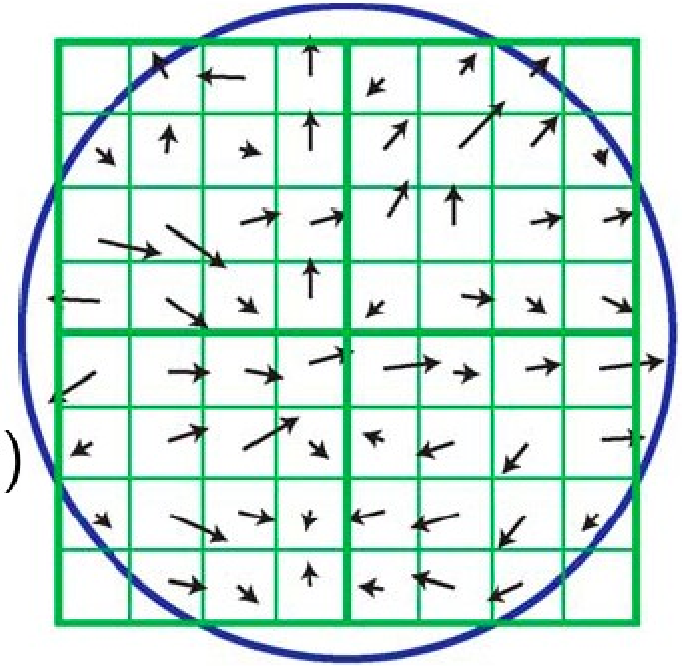
  $\nabla L(x, y, \tilde{\sigma}) = (L_x, L_y)^T$ for all points in the image

- Gradient orientation and magnitude images

  $$\theta(x, y, \tilde{\sigma}) = \arctan\left(\frac{L_y}{L_x}\right)$$

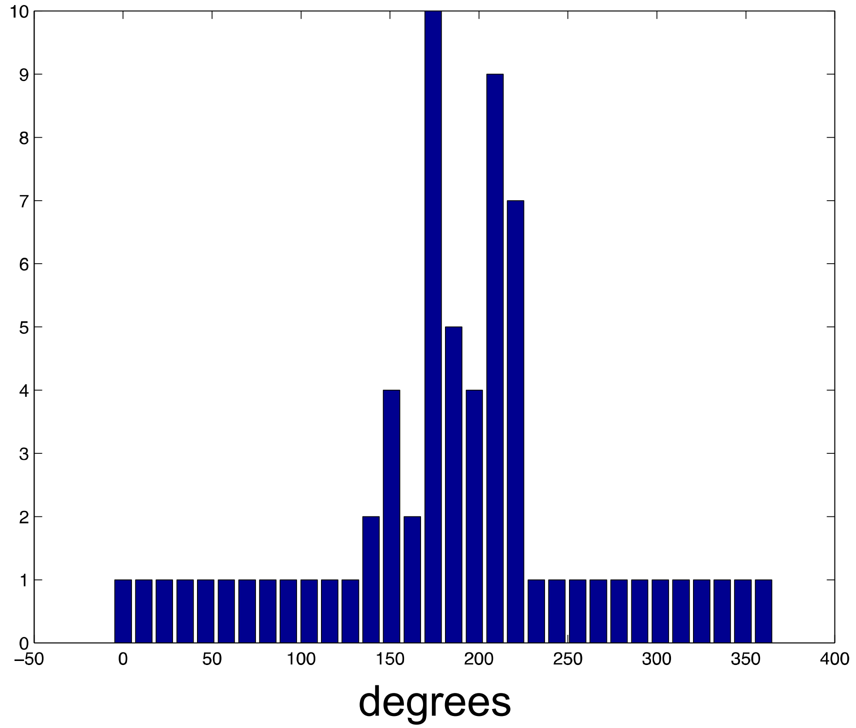
  $$m(x, y, \tilde{\sigma}) = \sqrt{L_x^2 + L_y^2}$$

- Build a 32 bin orientation histogram for neighborhood around $(\tilde{x}, \tilde{y}, \tilde{\sigma})$
  - Every point weighted with *m* and a Gaussian window $G(x - \tilde{x}, y - \tilde{y}, 1.5\tilde{\sigma})$

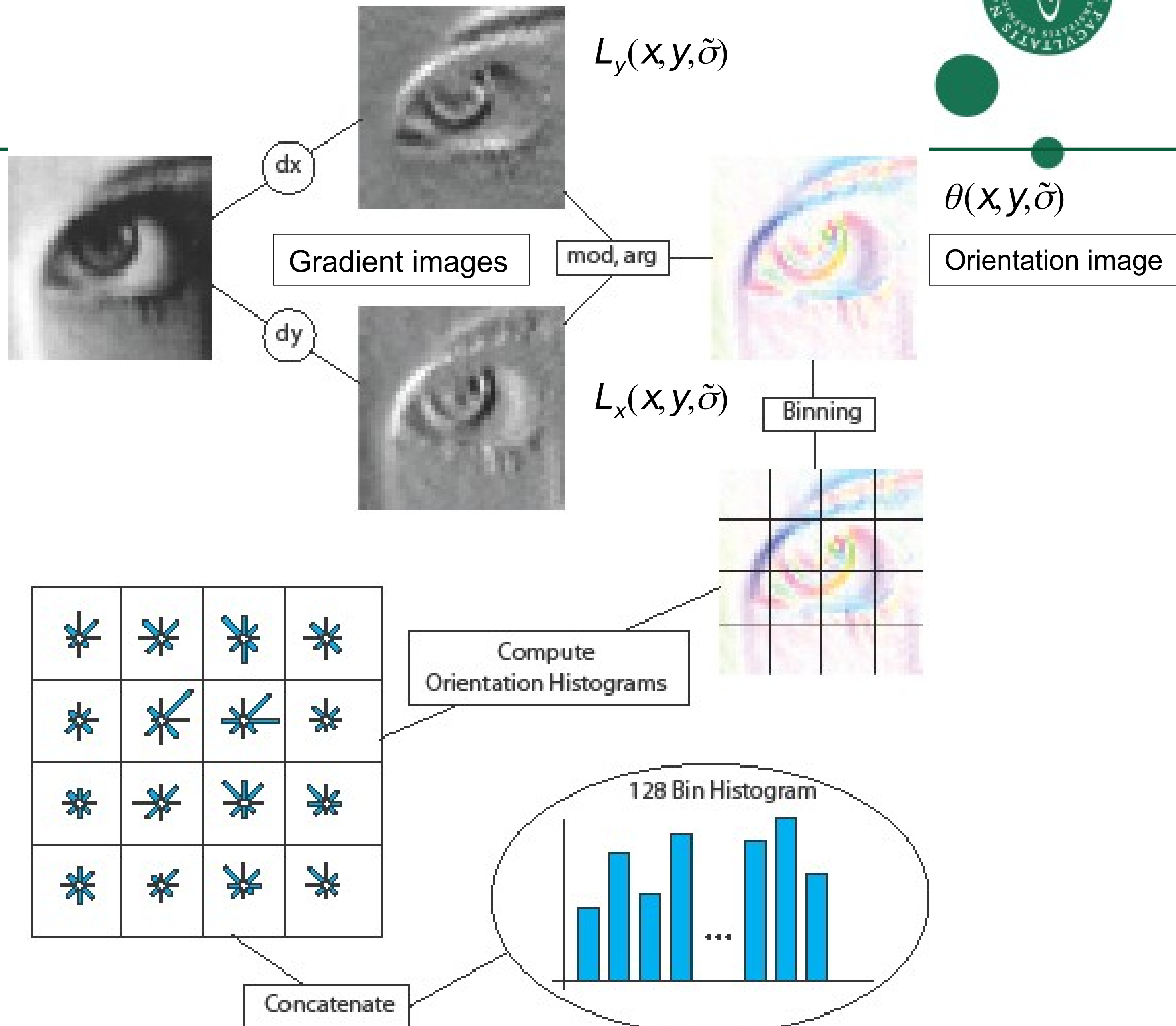


SIFT: Rotational invariance by orientation assignment

- Find highest peak and its orientation
- Create an interest point descriptor using this orientation
- For all other peaks larger than 80% of the highest peak – also create a descriptor using theses orientations.



SIFT descriptor



8-bin histograms

The SIFT descriptor: Details

- 8-bin orientation histograms:
  - When adding to bin, every data point is weighted with *m* and a Gaussian aperture window $G(x-\tilde{x}, y-\tilde{y}, 1.5\tilde{\sigma})$
  - Adding a data point to a bin also add a little to the neighboring bins (linear interpolation)
  - Pixels on the other side of a histogram grid border contributes a little to the histogram (linear interpolation)

- Feature vector:
  - Concatenate 8-bin histograms from the 4x4 grid into one vector with dimensionality 4x4x8 = 128.

The SIFT descriptor: Details

- Normalization of feature vector:
  - Normalize the feature vector: $\tilde{F} = F / \|F\| \Rightarrow \|\tilde{F}\| = 1$
  - Non-linear illumination changes (e.g. shadows) may cause high gradient magnitudes locally.
  - Therefore reduce all bin values larger than 0.2 down to 0.2.
  - Renormalize (normalize again): $\tilde{F} = F / \|F\| \Rightarrow \|\tilde{F}\| = 1$

SIFT matching:

- SIFT features are compared with Euclidean distance (L2-norm):

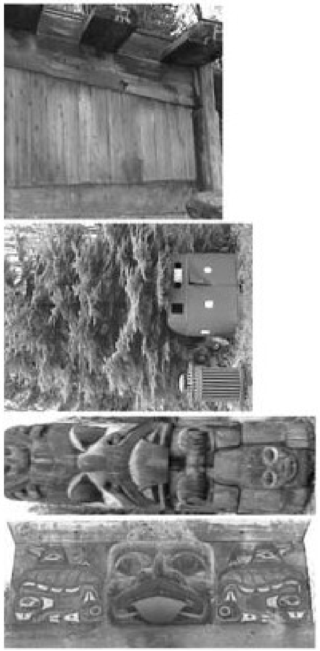
$$d_2^s(F_1, F_2) = \sqrt{\sum_{i=1}^{128} (F_1(i) - F_2(i)^2)}$$

- Matching SIFT features:
  - A match is accepted if

$$\frac{\text{Best}}{\text{2nd Best}} \leq 0.8$$

  - Best refers to the distance for the pair of features with smallest distance.
  - 2nd Best refers to the distance for the pair of features with second smallest distance.
  -

SIFT Results



The SIFT descriptor invariance’s

- Scale invariance:
  - From the (DoG) detector and further processing done at detection scale
- Rotational invariance:
  - From the orientation assignment procedure
- Approximate translational invariance:
  - From the grid of histograms. Can handle translations up to 4 pixels (within a grid cell).
- Affine illumination invariance:
  - From the choice of gradients additive brightness invariance is obtained. From normalization we obtain invariance to multiplicative contrast change.
  - Reduction of peaks give some robustness to nonlinear changes such as shadows

Open problems for SIFT descriptors

- Perspective distortion, but may be extended to become affine invariant

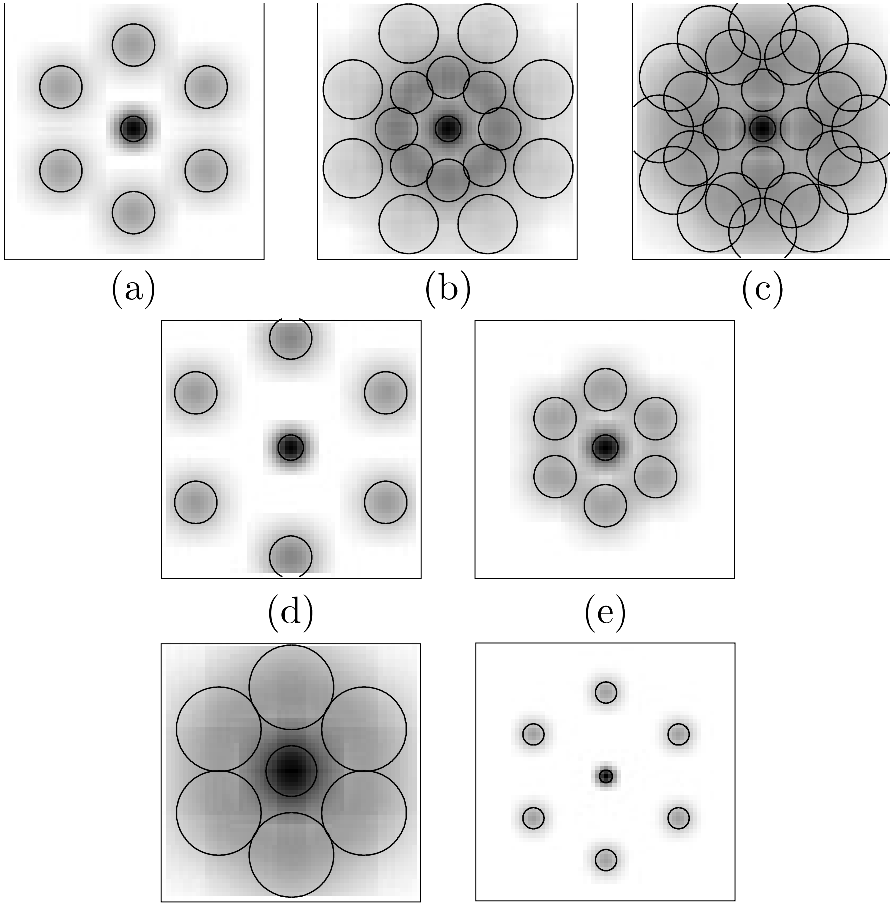
- Non-linear illumination such as cast shadows, changes to light color, and material reflection properties

- It is fairly high dimensional (128 dim.) and redundant

Variations on SIFT

- There are many variations of the SIFT descriptor:
  - GLOH
  - SURF
  - DAISY
  - PCA SIFT
  - Opponent SIFT
  - Gaussian opponent SIFT
  - CSIFT
  - ORB
  - BRISK
  - FREAK
  - …

DAISY – a common SIFT variation: Locations and spread of histograms



(a) (b) (c) (d) (e) (f) (g)

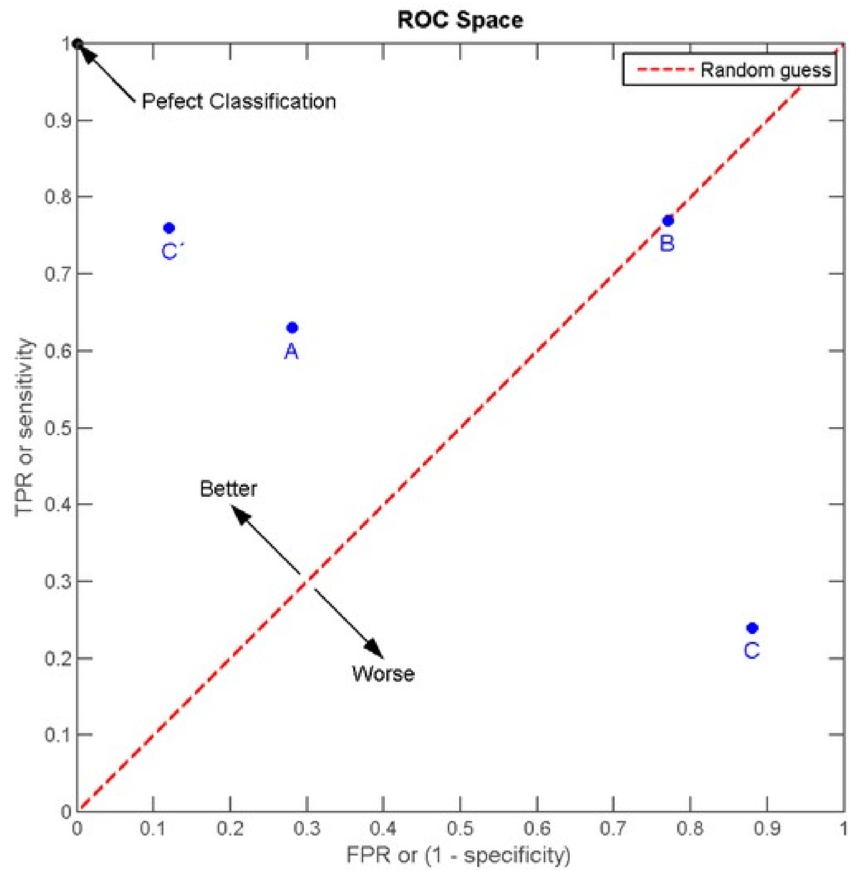
What is the best descriptor ?

- Detector and Descriptor performance is difficult to separate
- Much work has been done
- Results are not conclusive
- SIFT often best, but is very slow, and hard to implement correct. If made affine invariant even slower
- SURF sometimes ok, but performance may vary. Fast
- ORB, BRISC and other ”binary descriptors” often perform inferior

Aside: ROC and AUC?

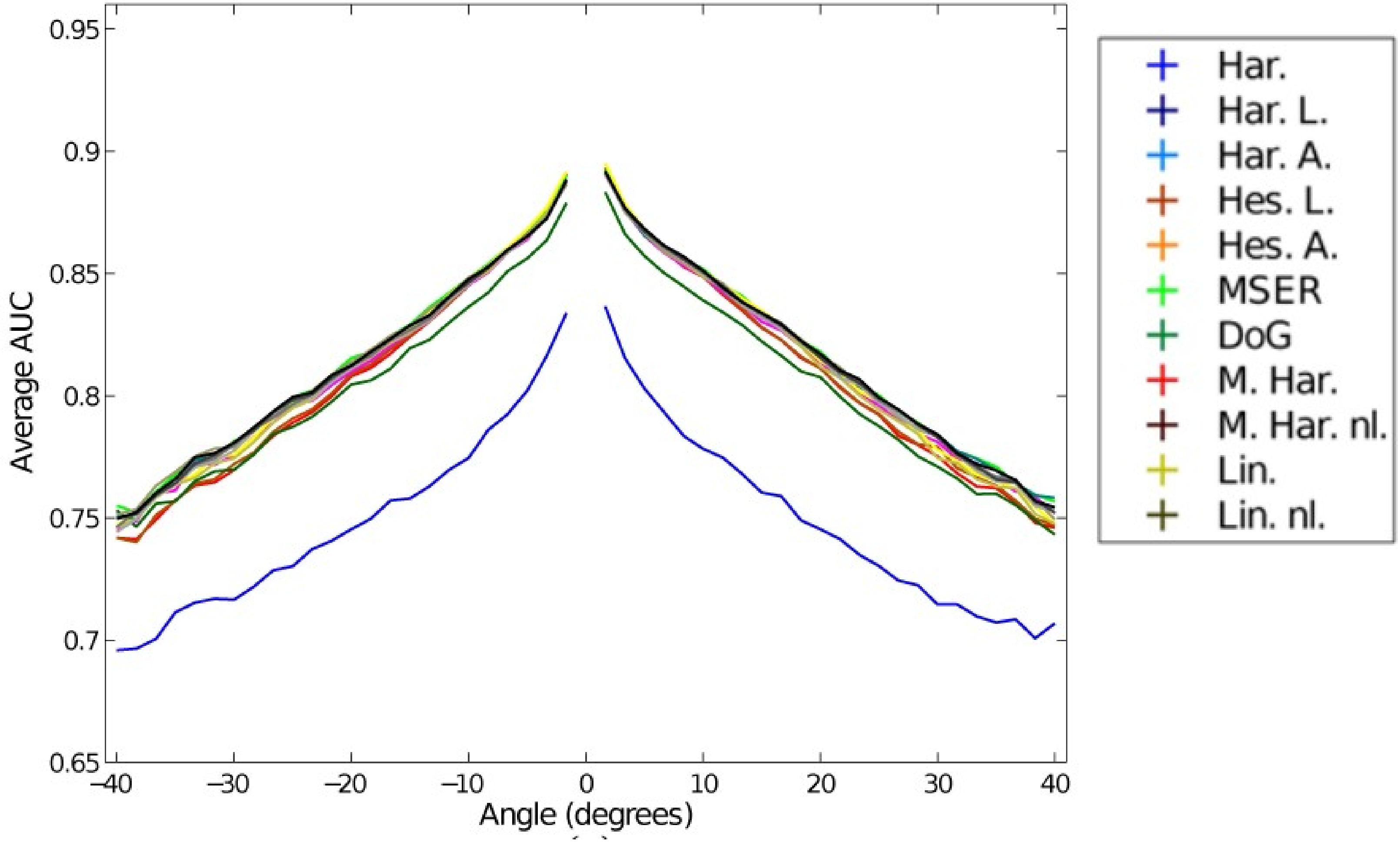
- Receiver operating characteristic (ROC):
  - TPR = TP / P (Recall)
  - FPR = FP / N
- Area under the ROC curve (AUC):
  - AUC close to 1 is good
  - The probability of a correct match

		actual value: p	actual value: n	total
prediction outcome	p'	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
total		P	N	



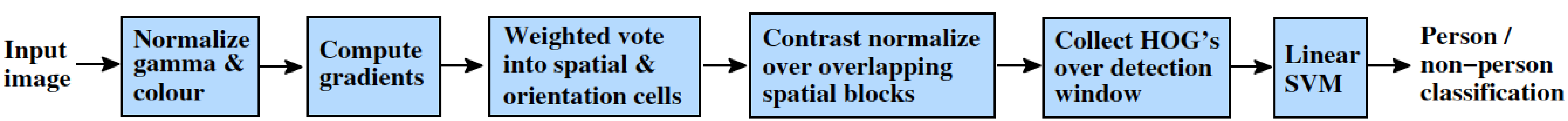
From Wikipedia 2020

Results at a Glance, Descriptors



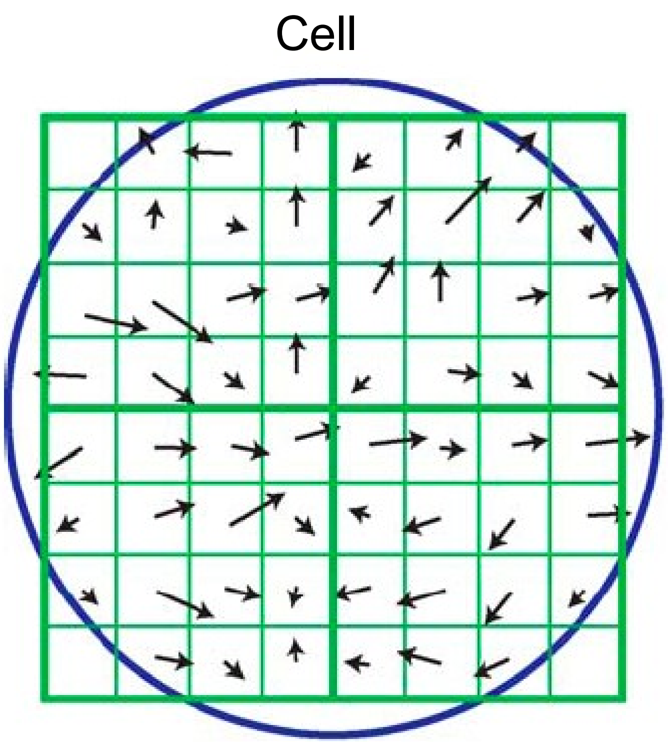
Object detection using Histograms of Oriented Gradients (HoG) features (Dalal & Triggs'05)

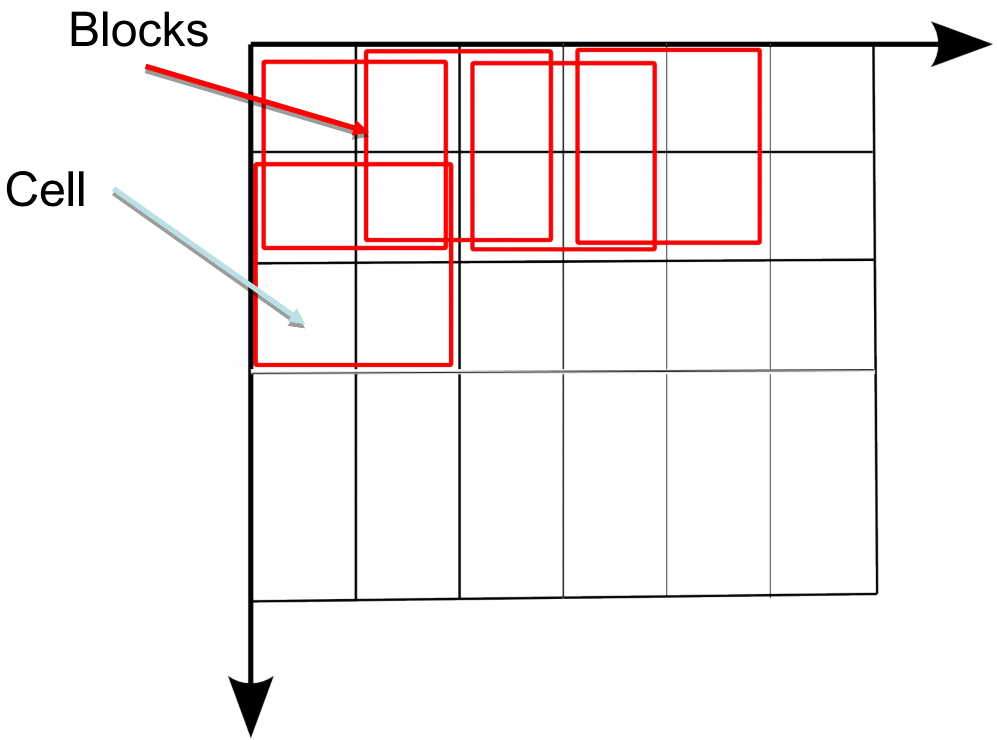
- Using the sliding detection window approach $(64 \times 128 \text{ pixels})$.
- Normalization of image prior to detection:
  - Use all RGB channels
- Compute gradient image multi-scale pyramid:
  - For each color channel, compute intensity gradients
  - For each pixel, pick the gradient from the color channel with largest gradient magnitude (simple color gradient)



Histograms of Oriented Gradients (HoG) feature
(Applied to complete detection window)

- Divide the detection window into 8 x 8 pixels non-overlapping cells.
- Divide the detection window into 16 x 16 pixels overlapping (8 pixel stride) blocks covering 2 x 2 cells





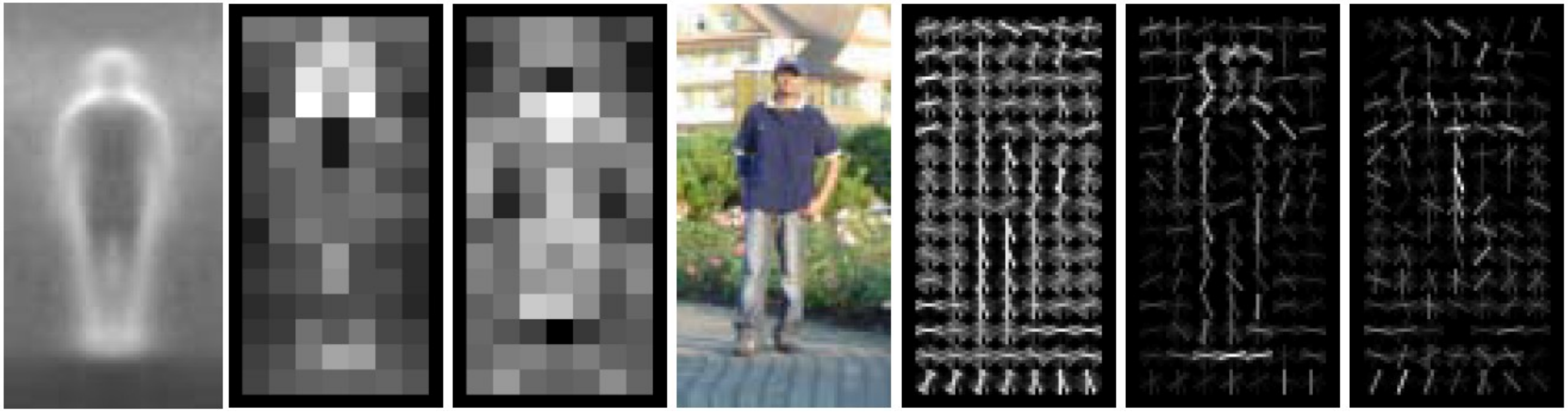
Histograms of Oriented Gradients (HoG) feature
(Applied to a detection window)

- For each block:
  - Concatenate cell histogram vectors to a feature vector $\mathbf{F}$
  - Normalize feature vector:
    - Euclidean: $\mathbf{F} = \mathbf{F} \Big/ \sqrt{\|\mathbf{F}\|^2 + \varepsilon^2}$
    - Peak clipping followed by renormalization (just as in SIFT)

- For detection window:
  - Concatenate block feature vectors to form a joint feature vector for the detection window
  - Dimensionality for 64 x 128 = 8192 pixels detection window:
    9 bins x 4 cells x (7 x 15) blocks = 3780 dimensions
  - Apply a classifier to the joint feature vector – the detection window feature (Dalal & Triggs uses a linear Support Vector Machine (SVM))

HoG features visualized

HOG has shown successful in detecting (upright) people, but can be difficult to adapt to other cases.



- **David Lowe's paper (SIFT-1.pdf in Absalon)**

- **Next time:**
  - **Imaged formation,**
  - **light models,**
  - **photometric stereo**
    (much more my cup of tea...)