



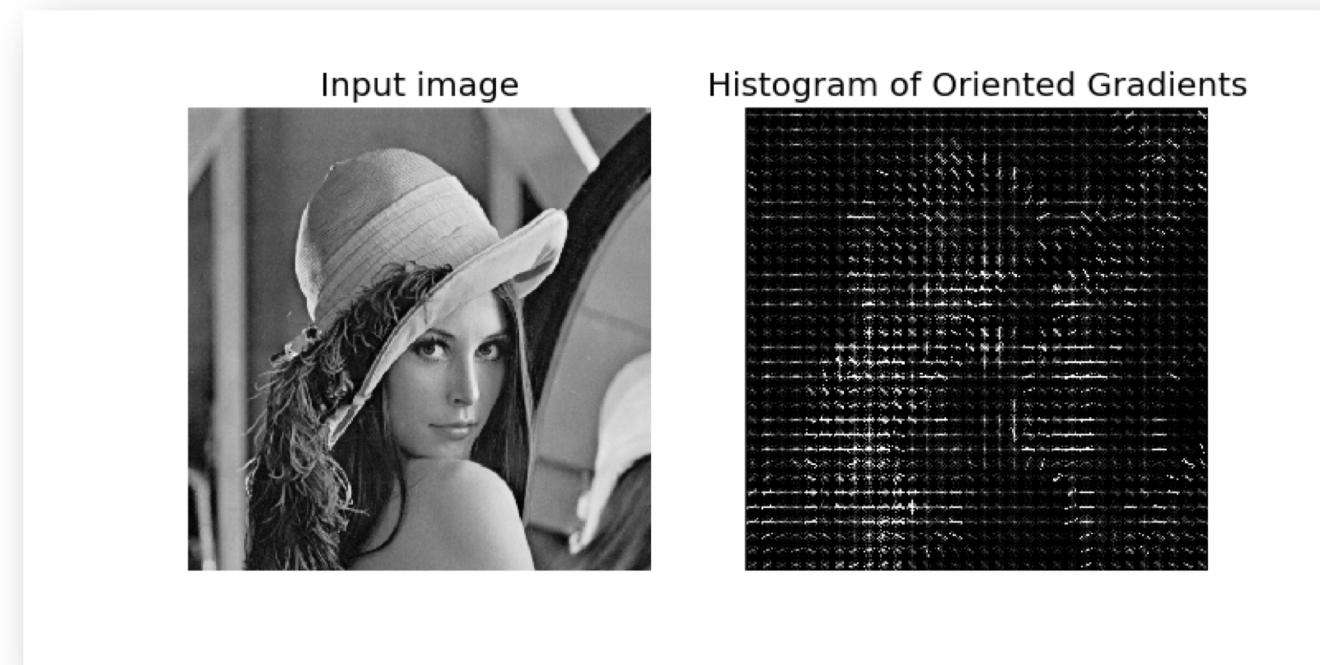
Part VI: Local Feature Extraction

Nicola Conci
nicola.conci@unitn.it

Use of gradients – HOG



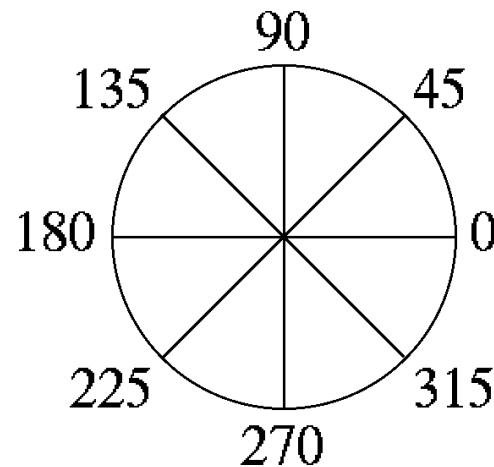
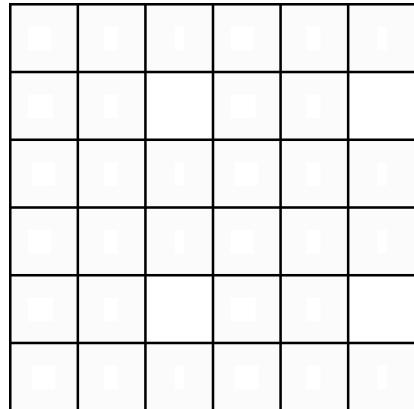
- Intensity and direction of edges is one of the most salient features that help us characterizing objects



Histogram of oriented gradients – HOG



- Divide the image into **small cells**
- Cells can be **rectangular** or **radial**
- For each pixel in the cell **compute the orientation** of edges



HOG



- Each pixel within the cell casts a weighted vote for an **orientation-based histogram** based on the values found in the gradient computation
- In other words, each cell is represented through a **1D array of gradient directions**
- The vote is based on the **gradient magnitude**
- **Intensity is locally normalized**, to account for illumination changes and shadowing especially when using larger areas (blocks), consisting of more cells
- Normalization of the cell energy is performed in the RGB or LAB color space

Normalization



- Blocks can be of rectangular (R-HOG) or circular (C-HOG) shape
- Normalization can be computed using different metrics, such as L1 and L2 norm

L1-Norm

$$f = \frac{v}{(\|v\|_1 + e)}$$

L2-Norm

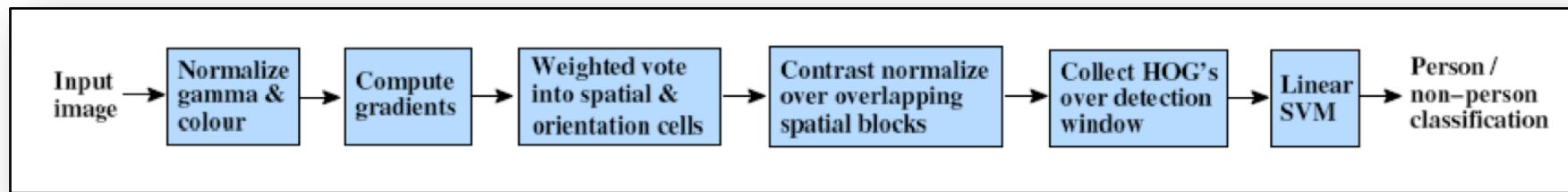
$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

- v is the non-normalized vector of the histograms for a given block
- e is a small constant

Classification



- The HOG representation is the feature-set used for learning
- General application in object detection
- Good results in human detection
 - Binary classification (human vs non-human)



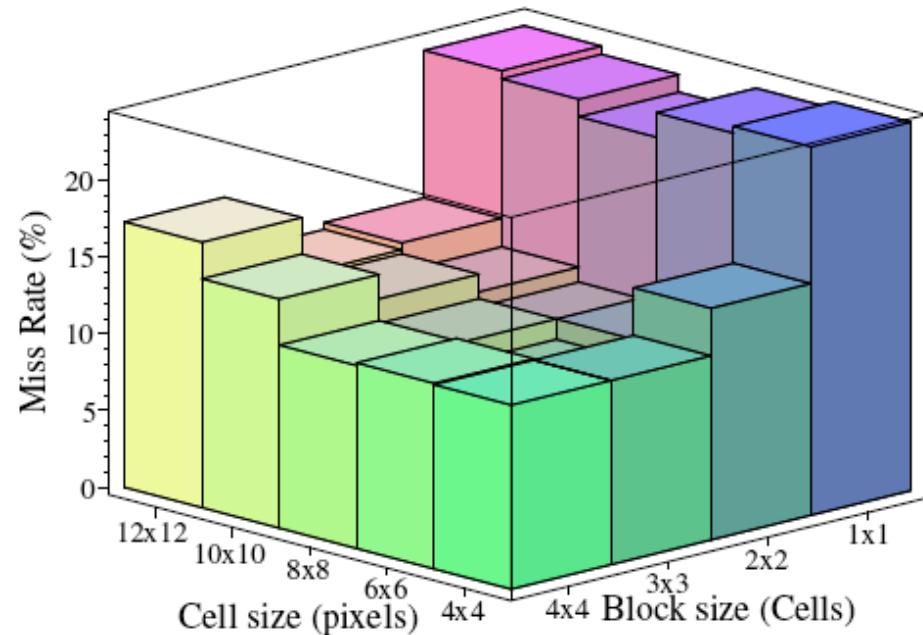
Configuration



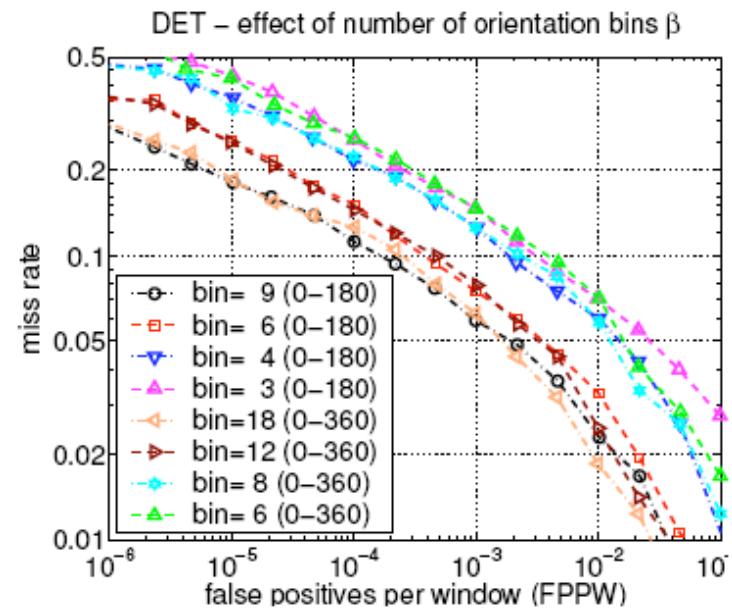
- Dalal and Triggs (2005):
 - Input data are arranged in a 64x128 window
 - Cells of 8x8 pixels
 - Blocks of 2x2 cells
 - Each cell is a 9-bin histogram
 - Each block is represented by the concatenated feature vector (36D)
 - Blocks overlap
 - 7x15 blocks in a 64x128 window

Dalal, Navneet, and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." *International Conference on Computer Vision & Pattern Recognition (CVPR'05)*

Performance



Miss rate as function of the cell and block size



Miss rate as function of the number of HOG bins

Comments



- For good performance use:
 - fine scale derivatives (no smoothing)
 - many orientation bins
 - moderately sized, strongly normalized, overlapping descriptor blocks.

The problem of scale



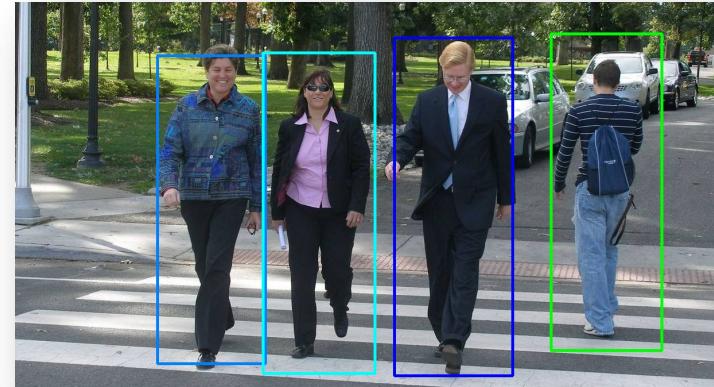
- The algorithm by Dalal and Triggs is very efficient at single scale
- In videos, human can be detected at different size
- Analysis at multi-scale → complexity increases considerably
- Use Integral Histogram
 - Within the 64x128 multiple windows are considered
 - Different size, location, and aspect ratio

Feature compression

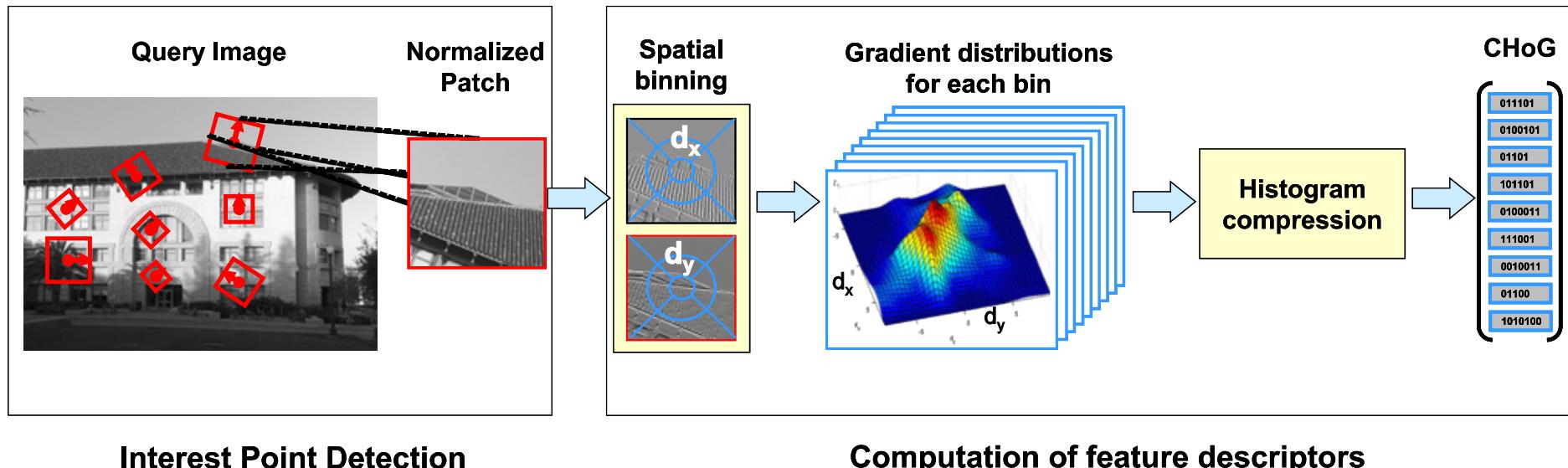


- The extracted features can be
 - processed online
 - sent over the Internet and matched real time

- Applications:
 - Surveillance and monitoring
 - Augmented reality
 - Media retrieval

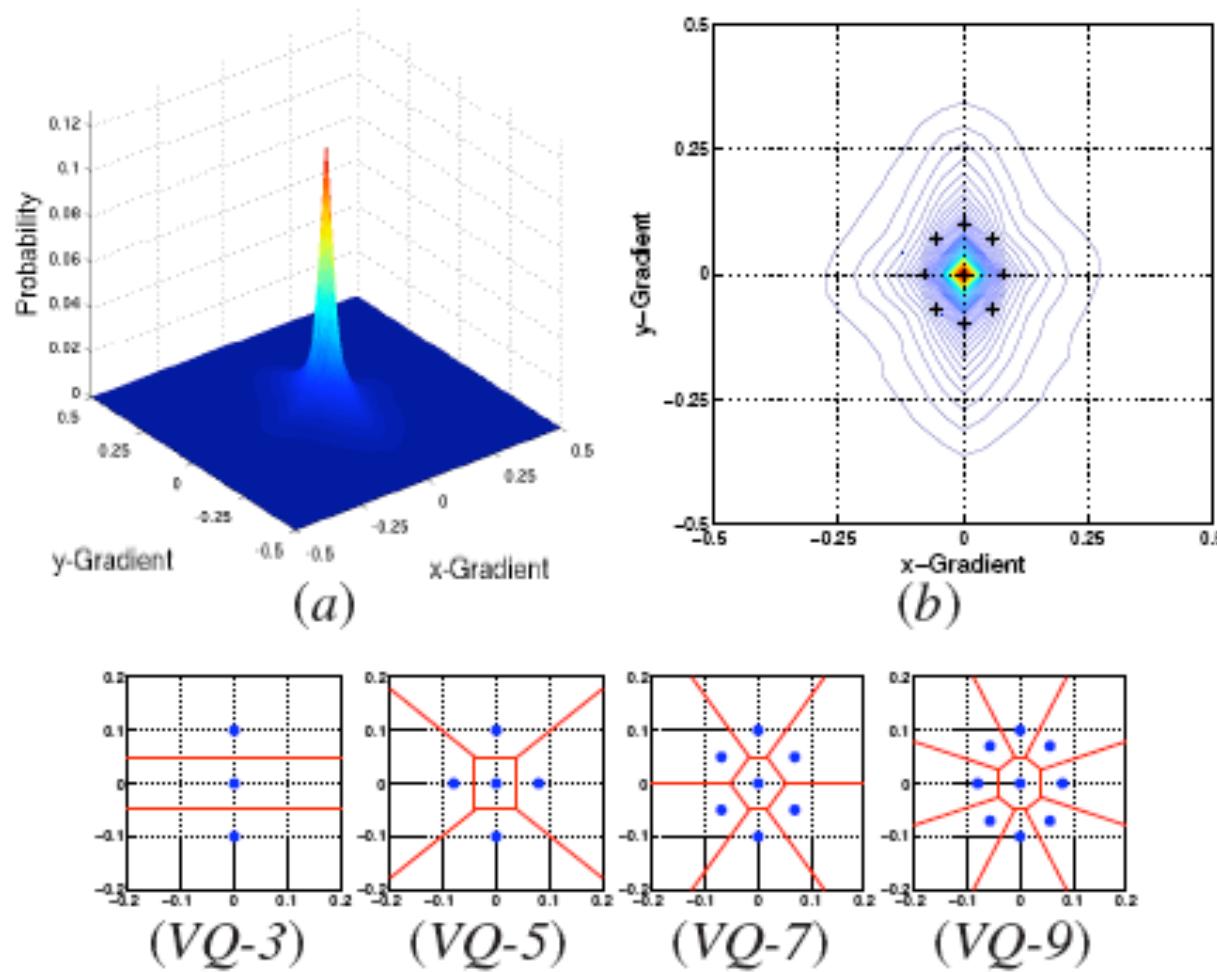


Feature compression

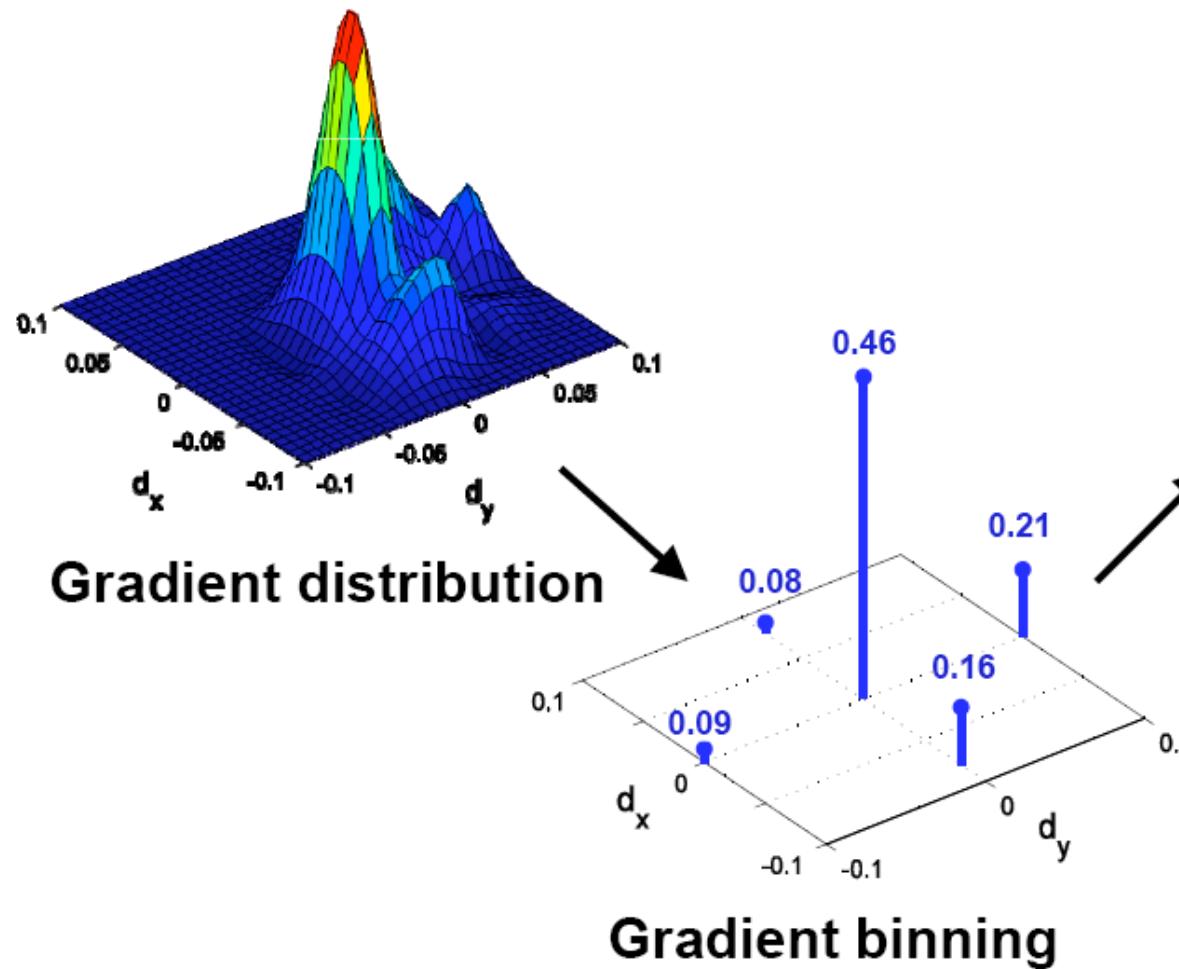


Chandrasekhar et al. CHoG: Compressed Histogram of Gradients - A low bit rate feature descriptor, CVPR 2009
Chandrasekhar et al. Compressed Histogram of Gradients: A Low-Bitrate Descriptor , IJCV 2011

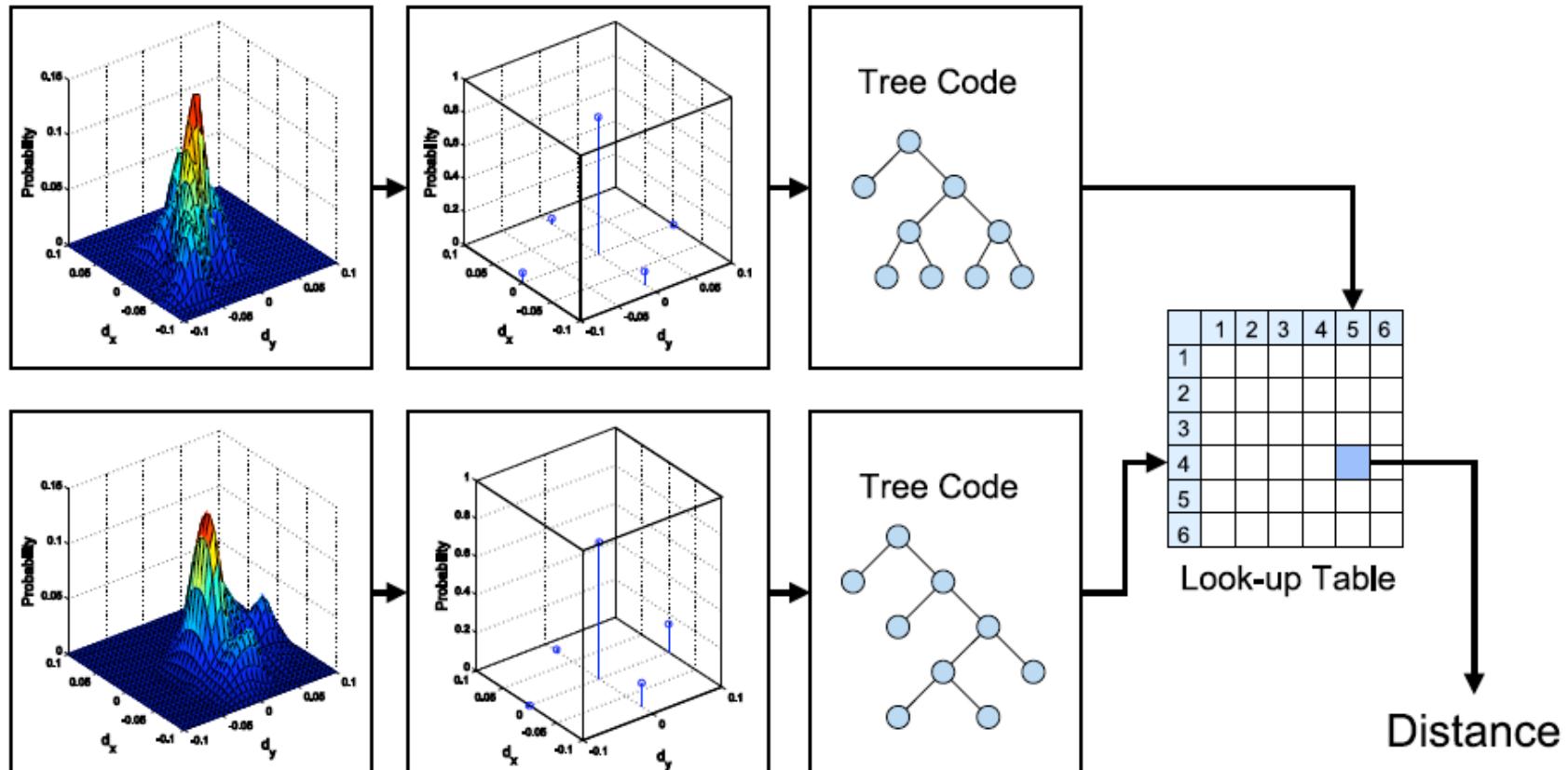
Histogram binning



Huffman coding



Search and match



SIFT: Scale Invariant Feature Transform



- Extraction of salient points (keypoints) in the image for:
 - Object detection
 - Tracking
 - Image matching
- The idea is to make the image of concern scale-invariant

Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

SIFT: how to



1. Construct a subspace representation of the image and progressively apply a Gaussian smoothing filter
2. At every iteration, each image becomes a blurred version of the previous one
3. Find keypoints
4. Compute the descriptor

SIFT - filtering



$$L(x,y,\sigma) = I(x,y) * \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

- $I(x,y)$ is the image
- $L(x,y,\sigma)$ is the scale space of the image after convolution.
- $L(x,y,\sigma)$ is the result of the filter subject to σ
- σ defines the strength of the filter

SIFT - octaves



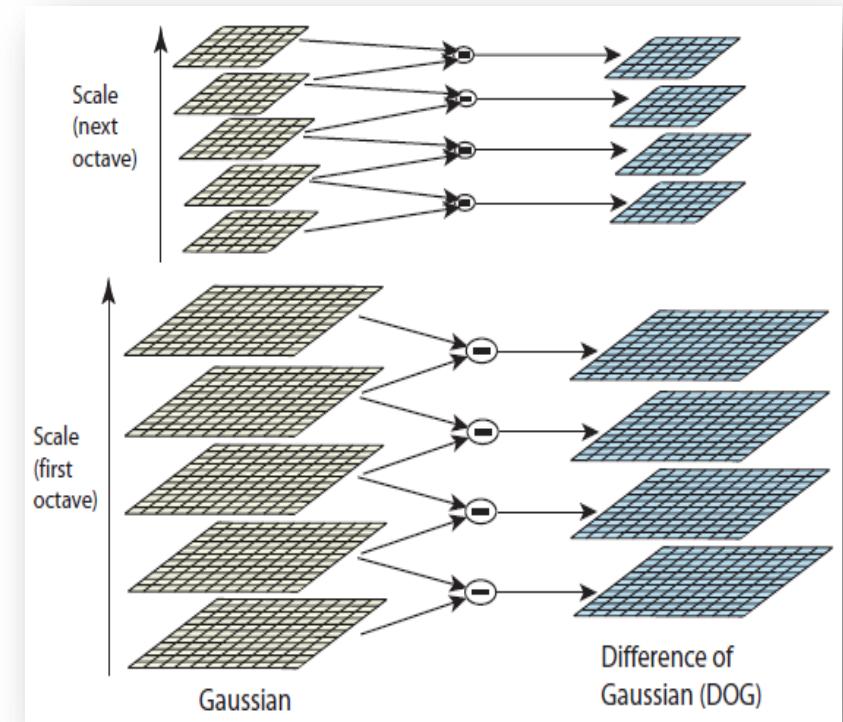
- We build-up a so-called **octave** collecting the images of the **same size** and each one represents a **blurred** version of the previous one
- Mathematically, the blurring is controlled via the value of σ , hence if the current image is blurred with σ , the next one goes with $k\sigma$ and so on
- To build-up the **next octave**, the original image is down-sampled to **half** its size and the blurring operation is repeated in the same manner
- Theoretically octaves can be created as long as the image can be downsampled.

SIFT - DoG



- A difference of Gaussians (DoG) is then obtained by subtracting each Gaussian image **within** an octave from the previous one. Hence, for N image samples, (N-1) DoGs can be obtained subject to:

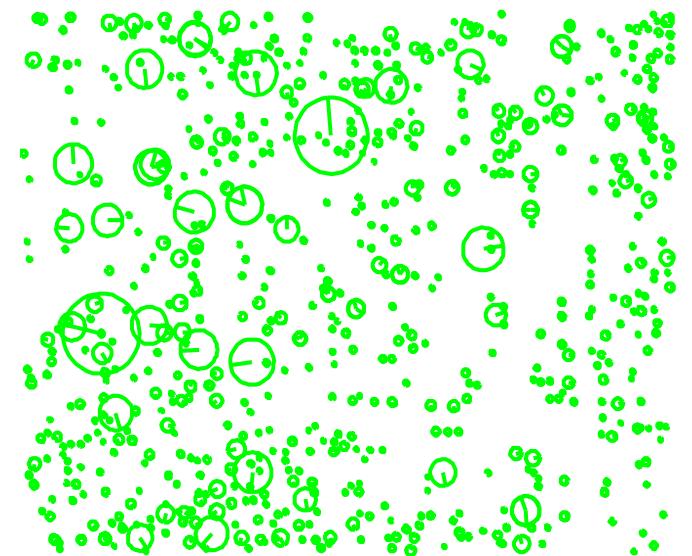
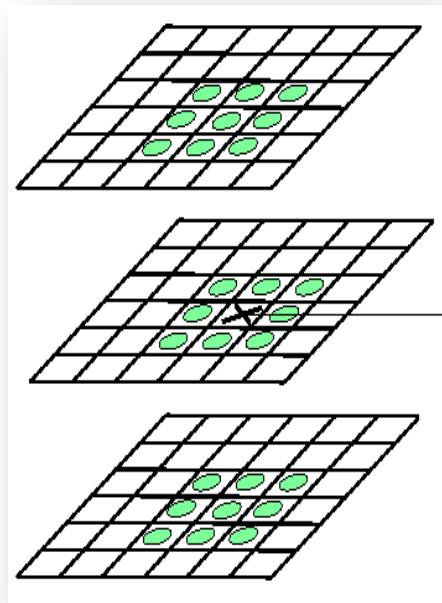
$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$



SIFT - Keypoint selection



- To grab the most salient points from the DOGs, each pixel from a given DOG is compared to its 26 neighbors (8 in the same scale, 9 above and 9 below).
- A pixel is kept as a key-point only if it is greater (i.e. maxima) or smaller (i.e. minima) than its neighbors.



(a)

SIFT – Stability of the keypoint and refinement



- The selection of the point might be noisy
- Apply a second order Taylor expansion, where $X=(x,y,s)$

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$$

- Setting the derivative of $D(X)$ to zero, the maxima and minima can be obtained

SIFT - Edges



- DoG function exhibits strong response over the edges
- Edges are not necessarily good keypoints
- At the candidate keypoint location, the Hessian matrix is computed

$$H = \begin{pmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{pmatrix}$$

- Let us consider $r=\alpha/\beta$, where α and β are the largest and smallest eigenvalue in magnitude. If the equation is satisfied for a certain r_{th} , then the keypoint is strong

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

SIFT - Orientation



- The eigenvalues of H are proportional to the principal curvatures of D
- For the points that exhibit a good curvature in both dimensions, the gradient and orientations are computed.
- This step achieves the invariance to rotation

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

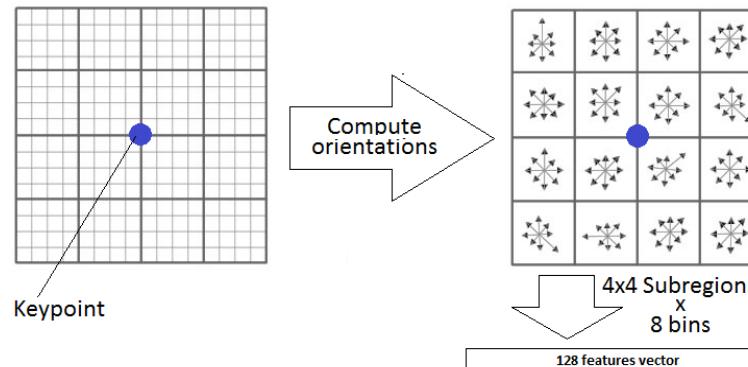
$$\theta(x, y) = \tan^{-1}(L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))$$

- The region within which m and θ are computed is relative to the scale of the keypoint, the higher the scale (smoothing), the larger the computation region

SIFT – The feature vector



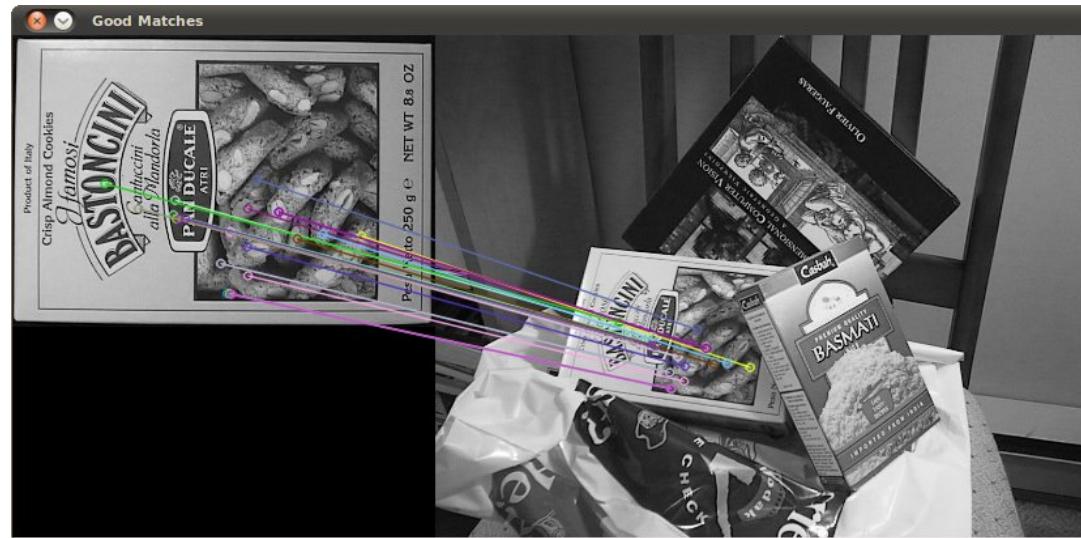
- From the orientations a 36-bin histogram is computed
- If a keypoint exhibits multiple peaks in the histogram that are higher than 80% of the highest peak, a new keypoint is instantiated with same location and scale.
- The 16×16 area (oriented according to the key-point orientation computed before) around the selected keypoint is divided in 4×4 regions, and for each of them a histogram of the gradients (8bins) is computed
- This turns out in a descriptor of 128 elements: $4 \times 4 \times 8$



Feature Matching



- Once the descriptor is constructed, search and match can be performed
- Matching usually done by
 - Nearest neighbor search
 - Optimization (e.g. RANSAC)



https://docs.opencv.org/2.4/doc/tutorials/features2d/feature_flann_matcher/feature_flann_matcher.html