Image Features

C. Maskaleris, I. Karakostas, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 2.2

Artificial Intelligence &
Information Analysis Lab

Image features

- SIFT, ORB, SURF, LSK, HOG, LBP, etc
- Gabor features
- Convolutional (CNN) features
- Bag of Features
- Feature point matching
- Image Feature Applications

Image Features

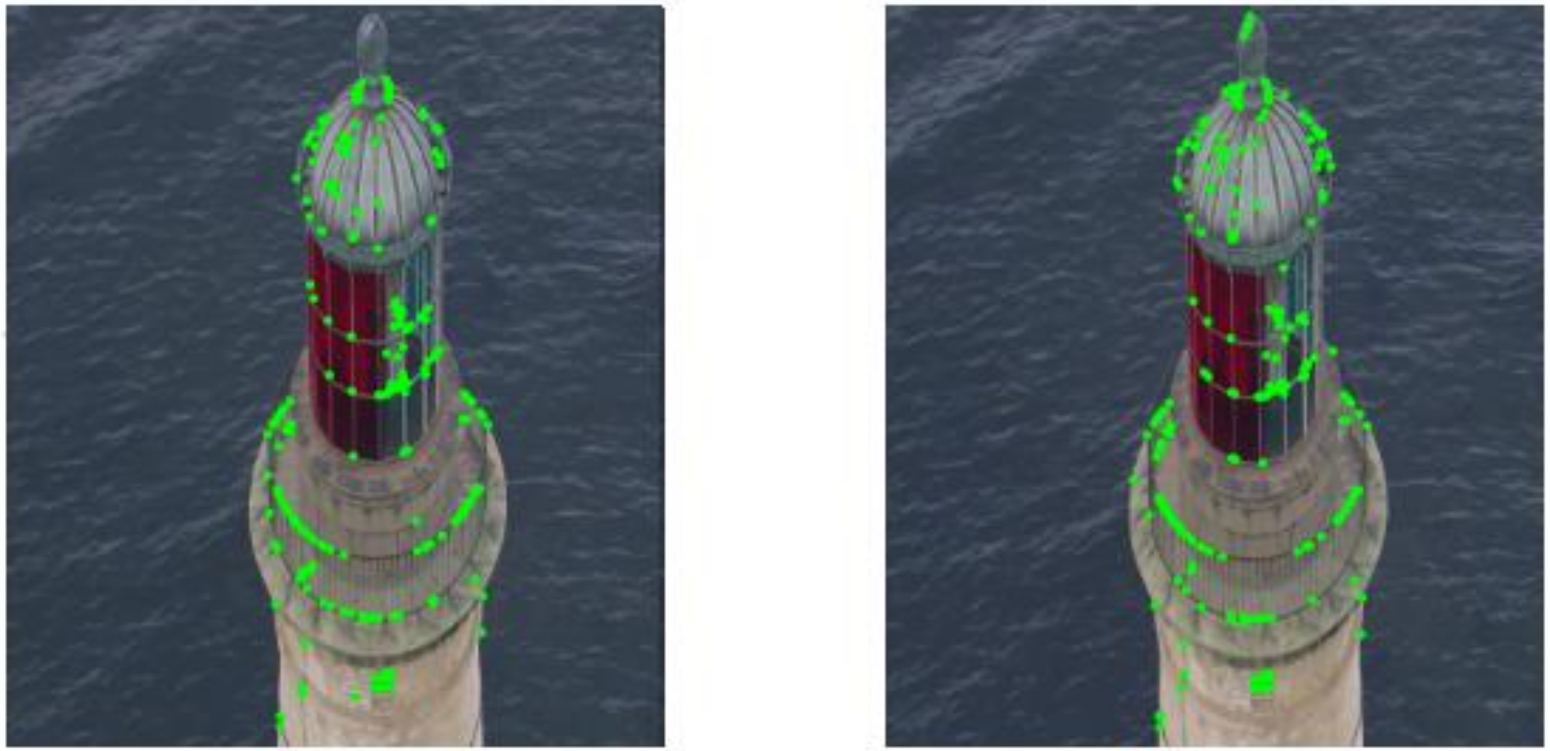


Image Landmarks.

Image Features

- Applications:
- Object description and recognition
- Feature correspondence
  - Image registration
  - Object tracking
  - Motion estimation
  - Disparity estimation
  - 3D object shape reconstruction.

Feature Extraction

- ***Local feature***: a small image region having interesting spatial characteristics (e.g., image corner).
  - It can be described by a ***feature descriptor*** (N-dimensional vector).
- A feature descriptor is a useful local image representation.
- Typically (not always), a feature detector/descriptor produce description vectors that are ***invariant*** to several image transformations (e.g., geometrical ones).

Feature Extraction

- ***Feature detectors***:
  - SIFT, AGAST, SURF, Hessian Affine, CeNSuRe, BRISK, ORB, AKAZE, or simply dense sampling.

- ***Feature descriptors***:
  - SIFT, SURF, DAISY, HOG, LIOP, LUCID, BRIEF, BRISK, FREAK, ORB, AKAZE, LATCH, CENTRIST, BinBoost, LMoD.

Image Features

- Trivial local image features:
  - Local greyscales values or RGB vectors.
  - Local greyscale, RGB, or other color histograms.

Image Features

- Many image features are based on image edge magnitude and orientation.
- Local ***image differentiation*** techniques can produce local ***image gradients***:

$$\nabla f(x,y) = [\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}]^T \triangleq [f_x \quad f_y]^T.$$

- ***Gradient magnitude***:

$$e(x,y) = \sqrt{f_x^2(x,y) + f_y^2(x,y)}.$$

- ***Gradient direction***:

$$\varphi(x,y) = \arctan(\frac{f_y}{f_x}).$$

Artificial Intelligence & Information Analysis Lab

Image Features

Gradient estimates can be obtained by using gradient operators of the form that perform local image differentiation:

$$\widehat{f_x} = \mathbf{w}_1^T \mathbf{x},$$
$$\widehat{f_y} = \mathbf{w}_2^T \mathbf{x}.$$

$\mathbf{x}$: local image pixel vector,

$\mathbf{w}_1$, $\mathbf{w}_2$: weight vectors (***gradient masks***).

Image Features

Gradient mask examples:

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

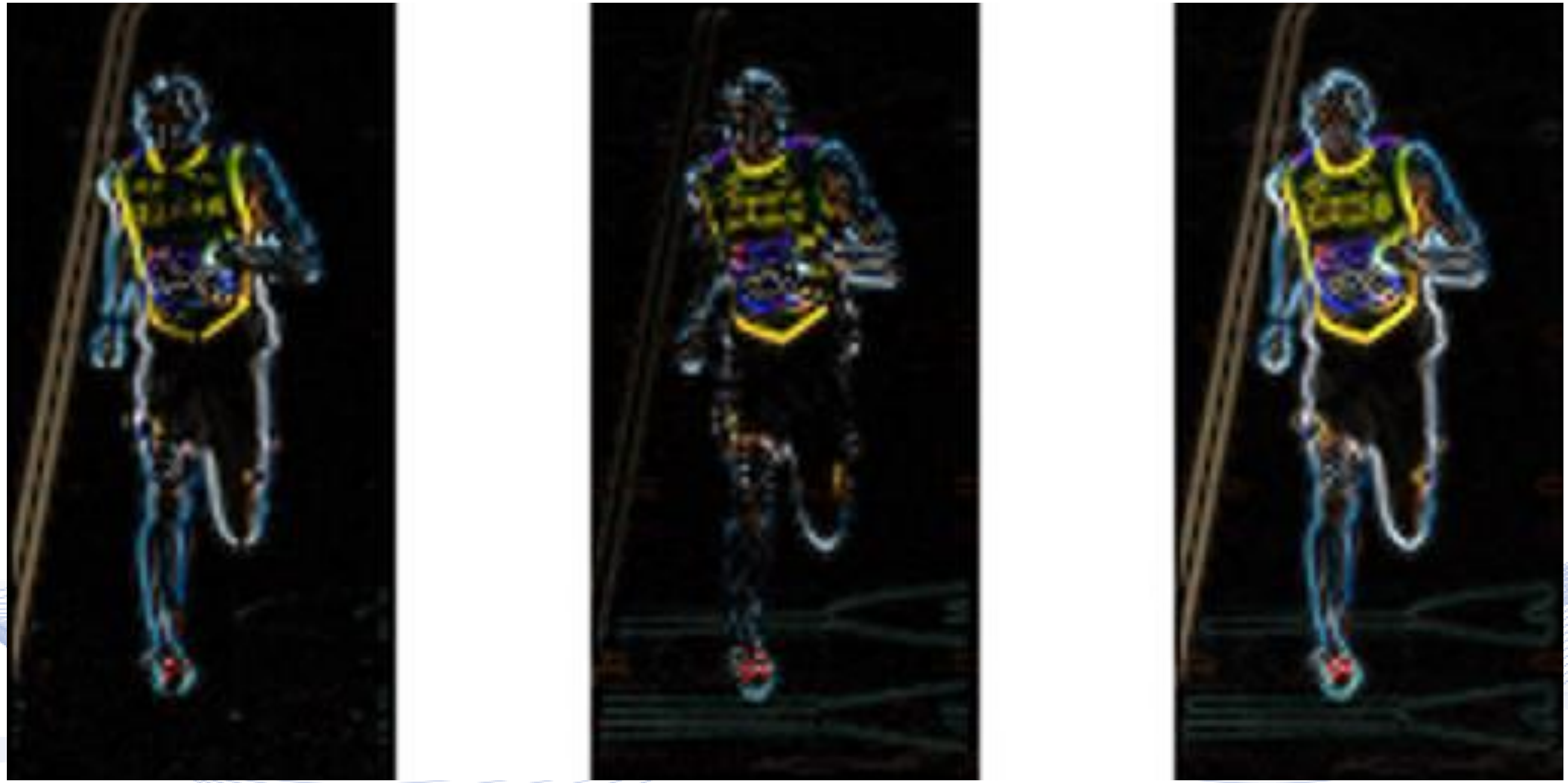
Prewitt edge detector masks.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Sobel edge detector masks.

Image Features



Color representation of local image edge directions.

Scale-Invariant Feature Transform (SIFT)

Scale-Invariant Feature Transform (***SIFT***)

- It finds ***image landmarks*** (***key points***) and outfits them with an appropriate feature descriptor.

- The descriptors should be have certain invariance properties against image transformations that do not alter significantly object visual appearance:
  - Geometrical invariance.
  - Illumination change invariance.

Scale-Invariant Feature Transform (SIFT)

Five steps involved in SIFT algorithm.

1. Scale-space Extrema Detection
2. Keypoint Localization
3. Orientation Assignment
4. Keypoint Descriptor
5. Keypoint Matching

Scale-Invariant Feature Transform (SIFT)

Scale-space extrema detection

- All image scales and locations are searched.
- It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation changes.

Keypoint localization

- At each candidate location, a detailed local model is fit to determine location and scale.
- Keypoints are selected based on stability measures.

Scale-Invariant Feature Transform (SIFT)

Orientation assignment

- One or more orientations are assigned to each keypoint location, based on local image gradient direction.
- All future operations are performed on image data that has been transformed relative to the assigned orientation, scale and location.
- This provides geometrical transformation invariance.

Keypoint descriptor

- Local image gradients are calculated at the selected scale in each keypoint neighborhood.
- They are transformed into a representation that is insensitive to significant local shape distortion and intensity changes.

Scale-Invariant Feature Transform (SIFT)

Orientation Assignment

- An orientation is assigned to each keypoint to achieve invariance to image rotation.
- A keypoint neigborhood is taken around the keypoint location depending on the image scale.

- For an image sample $f(x,y)$ at scale σ, the gradient magnitude $e(x,y)$ and orientation $\varphi(x,y)$ are calculated using numerical image differentiation.

Scale-Invariant Feature Transform (SIFT)

- An orientation histogram with 36 bins covering 360 degrees is created and is weighted by gradient magnitude and Gaussian-weighted circular window with radius equal to 1.5 times the scale of keypoint.

- The highest peak in the histogram and any of its peak above 80% of the highest peak is also considered to calculate the orientation. Keypoints with same location and scale, but different directions are created to feature point matching stability.

Scale-Invariant Feature Transform (SIFT)

Keypoint Descriptor

- A 16×16 pixel keypoint neighborhood is divided into 16 4×4 pixel sub-blocks.
- For each sub-block, an 8 bin orientation histogram is created, totaling 128 bin values to form the keypoint descriptor.
- Several measures are taken to achieve robustness against illumination changes, rotation etc.

Scale-Invariant Feature Transform (SIFT)

Keypoint Matching

- Keypoints between two images are matched by identifying their nearest neighbors.
- In some cases, the second closest-match may be very near to the first one, e.g., due to noise.
- In this case, if the ratio of closest-distance to second-closest distance is taken is greater than 0.8, they are rejected.
- This eliminates around 90% of false matches, while discards only 5% correct matches.

Histogram of Oriented Gradients (HOG)

The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest. Essentially represents a distribution of intensity fluctuations along different orientations (directions).

Steps to calculate HOG:

1. Preprocessing (resizing).
2. Calculate gradient Images.
3. Calculate the gradient histogram in 8×8 cells.
4. Block Normalization.
5. Form HOG feature vector.

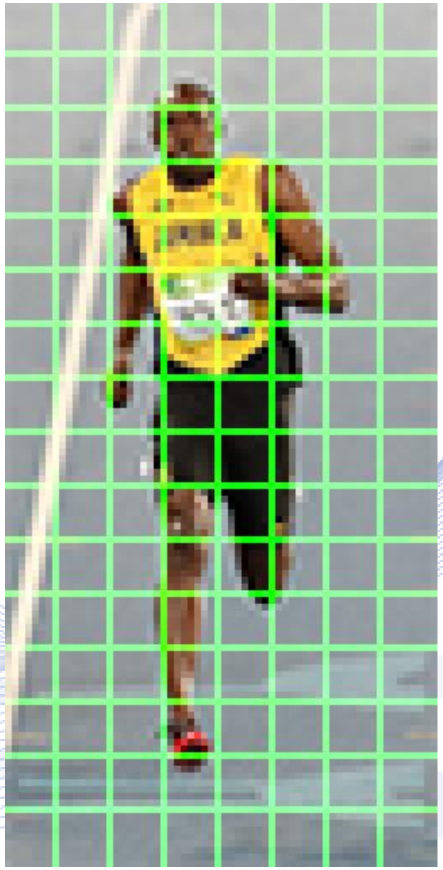
Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (***HOG***)

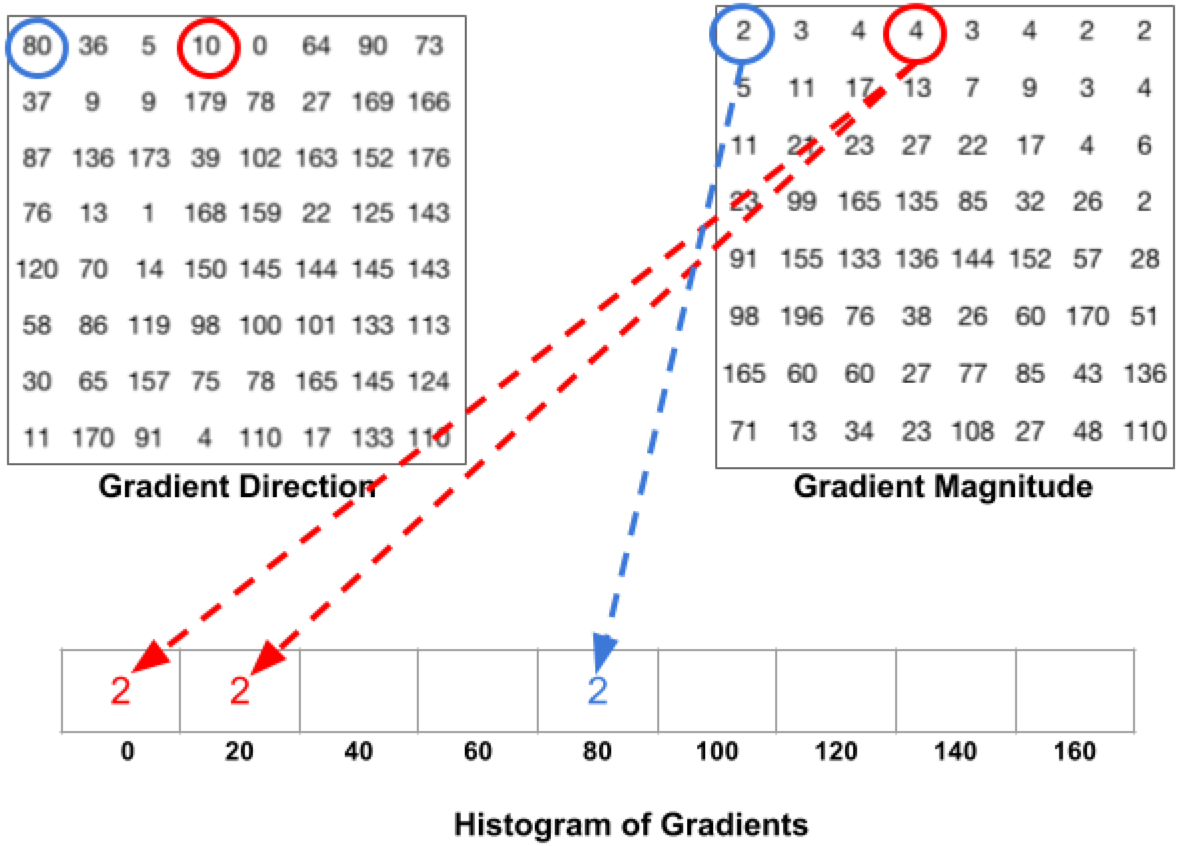
- Its feature vector is the (local) image gradient direction histogram.
- Small image patches can be used (e.g., $4 \times 4, 8 \times 8$ pixels).
- Typically, HOG histogram has 9 orientation bins:
  - 8 local neighborhood directions, ‘no direction’.

Histogram of Oriented Gradients

- Image is divided into small $4 \times 4, 8 \times 8$ pixel patches.
- In each patch, gradient magnitude and direction are calculated.
- For each patch, a 9 entry feature vector (image gradient direction histogram) is calculated.



Histogram of Oriented Gradients



Histogram of Oriented Gradients

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

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

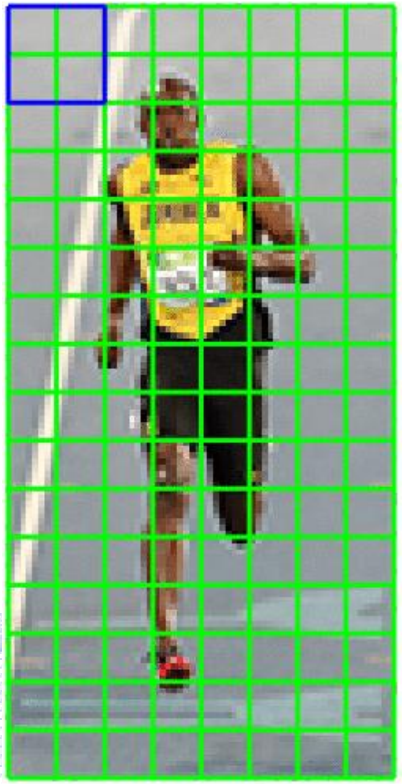
Gradient Magnitude

0	20	40	60	80	100	120	140	160
21.25								63.75

Histogram of Gradients

Histogram of Oriented Gradients

- HOG histograms are normalized over larger regions:
  - e.g., if HOGs are calculated in 4×4 pixel paches, they are normalized over 8×8 pixel regions.
  - Thus, a 36-entry vector normalization is performed.
- Normalization helps rendering HOG features more invariant to image intensity variations.



Speeded Up Robust Features (SURF)

Speeded Up Robust Features (***SURF***)

- It is a fast and robust algorithm for local, similarity invariant representation and for image comparison.
- The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications, such as object tracking and recognition.

Speeded Up Robust Features (SURF)

SURF Feature Extraction

- ***Integral image*** $f_\Sigma(\mathbf{x})$ at a location $\mathbf{x} = [x, y]^T$ represents the sum of all pixels in the input image f within a rectangular region formed by the origin and $\mathbf{x}$:

$$f_\Sigma(\mathbf{x}) = \sum_{i=0}^{x} \sum_{j=0}^{y} f(i, j)$$

- With $f_\Sigma(\mathbf{x})$ calculated, it only takes four additions to calculate the sum of the intensities over any upright, rectangular area, independent of its size.

Speeded Up Robust Features (SURF)

- It is used as a quick way of calculating the sum of pixel values in a rectangular image ROI, e.g., for calculating the average intensity within a given image ROI.
- It allows fast computation of box type convolution filters.
- Interest point location and the scale detection uses the determinant of a very basic ***Hessian matrix*** approximation, because of its speed and accuracy:

$$\mathbf{H}(f(x,y)) = \begin{pmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial y \\ \partial^2 f/\partial x \partial y & \partial^2 f/\partial y^2 \end{pmatrix}.$$

Speeded Up Robust Features (SURF)

- Image scale adaptation is performed by image filtering using a Gaussian kernel before differentiation.
- It leads to the use of an LoG kernel at scale σ:

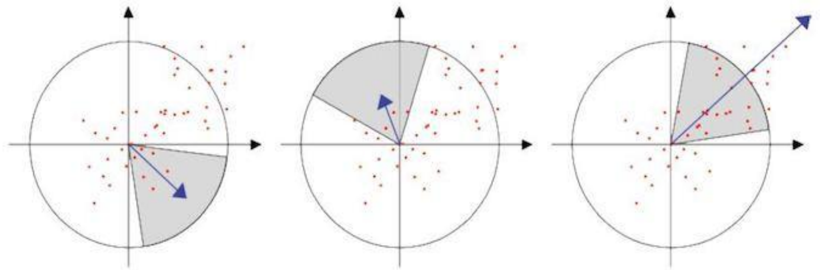
$$\mathbf{H}'(f(x,y)) = \begin{pmatrix} \partial^2(f ** G)/\partial x^2 & \partial^2(f ** G)/\partial x \partial y \\ \partial^2(f ** G)/\partial x \partial y & \partial^2(f ** G)/\partial y^2 \end{pmatrix}.$$

Speeded Up Robust Features (SURF)

SURF Feature Description

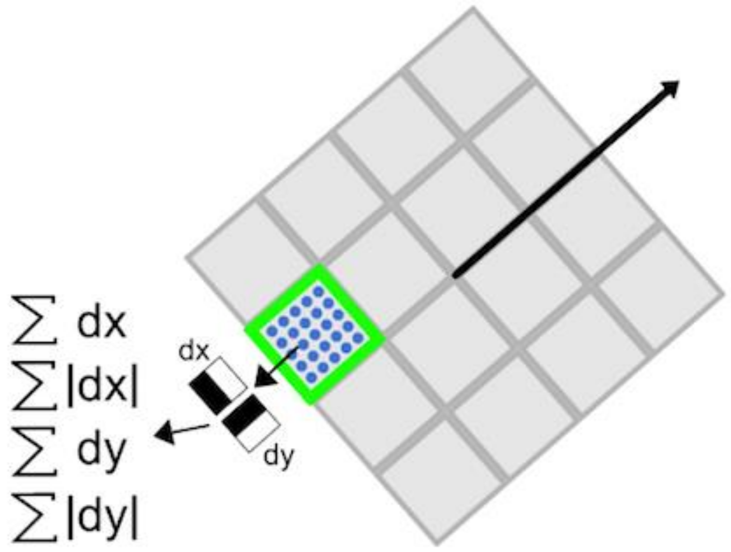
- The creation of SURF descriptor takes place in two steps.
- The first step consists of fixing a reproducible orientation based on information from a circular region around the keypoint.
- Then, a square region is constructed that is aligned to the selected orientation and is used to extract the SURF descriptor.

- ***Orientation Assignment.*** In order to be invariant to rotation, SURF tries to identify a reproducible orientation for the interest points.



Speeded Up Robust Features (SURF)

SURF descriptor components



Oriented FAST and Rotated BRIEF (ORB)

Oriented FAST and Rotated BRIEF (***ORB***)

- ORB performs as well as SIFT on feature detection (and is better than SURF).
- It is almost two orders of magnitude faster.
- ORB builds on the well-known FAST keypoint detector and the BRIEF descriptor.
- Both of these techniques are attractive because of their good performance and low computation cost.

Oriented FAST and Rotated BRIEF (ORB)

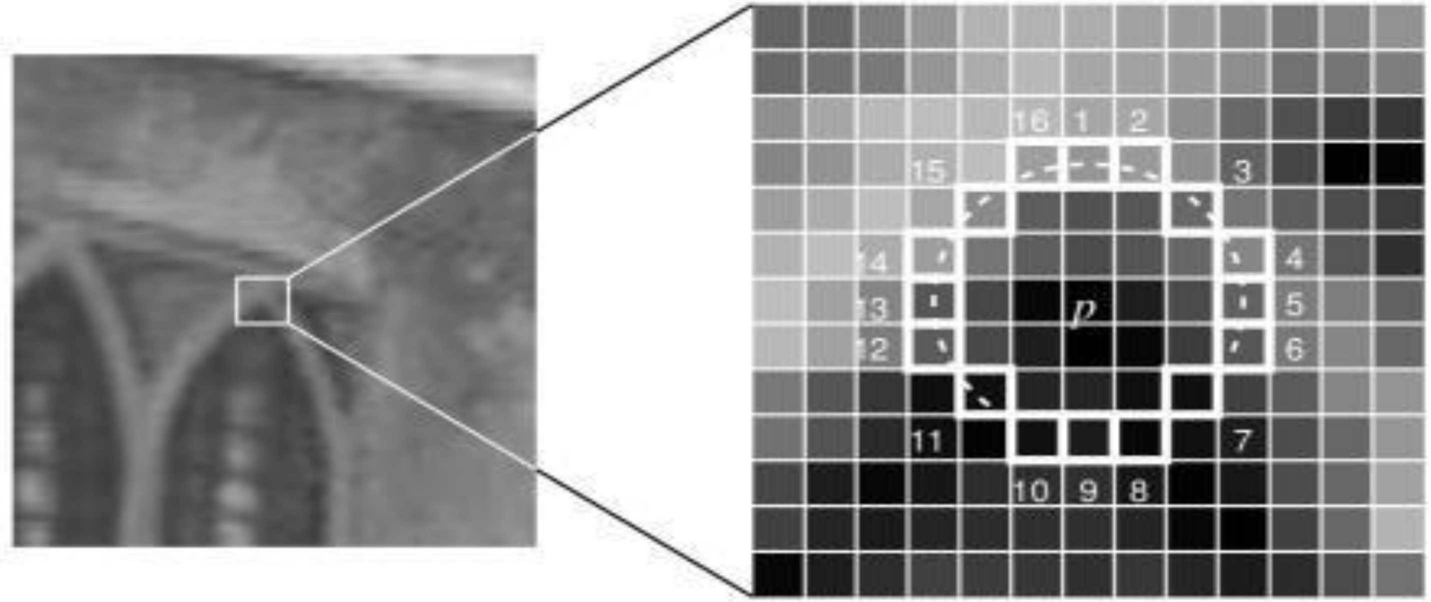
ORB main characteristics:

- The addition of a fast and accurate orientation component to FAST.
- The efficient computation of oriented BRIEF features.
- Analysis of variance and correlation of oriented BRIEF features.
- A learning method for decorrelating BRIEF features under rotational invariance, leading to better performance in nearest-neighbor applications.

Oriented FAST and Rotated BRIEF (ORB)

- ORB has a 256-bit binary descriptor.
- It is extracted and matched very fast.
- Hamming distance is used for ORB descriptor matching.
- Good for SLAM (tracking, relocation and loop detection).
- Multi-scale detection at same point appears at several scales.

Oriented FAST and Rotated BRIEF (ORB)



FAST corner detector

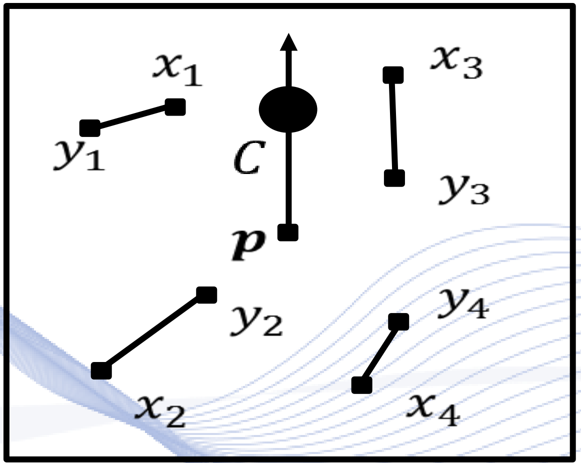
- Pixel $\mathbf{p}$ surrounded by consecutive pixels all brighter/darker than $\mathbf{p}$.
- Much faster than other feature point detectors.

Oriented FAST and Rotated BRIEF (ORB)

rBRIEF binary descriptor

- It is computed around a FAST corner.
- It has orientation.

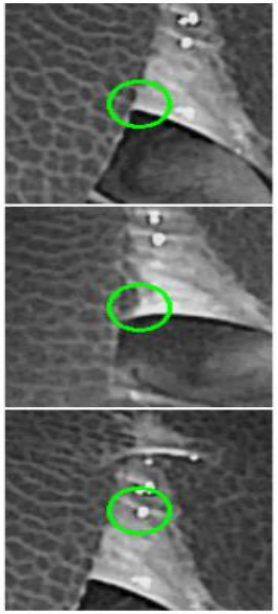
$$D_i(\boldsymbol{p}) = \begin{cases} 1 & \text{if } I(\boldsymbol{p}+\boldsymbol{x_i}) < I(\boldsymbol{p}+\boldsymbol{y_i}) \\ 0 & \text{otherwise} \end{cases}$$



$\boldsymbol{p} \triangleq$ interest point
$C \triangleq$ intensity centroid

Hamming distance 5

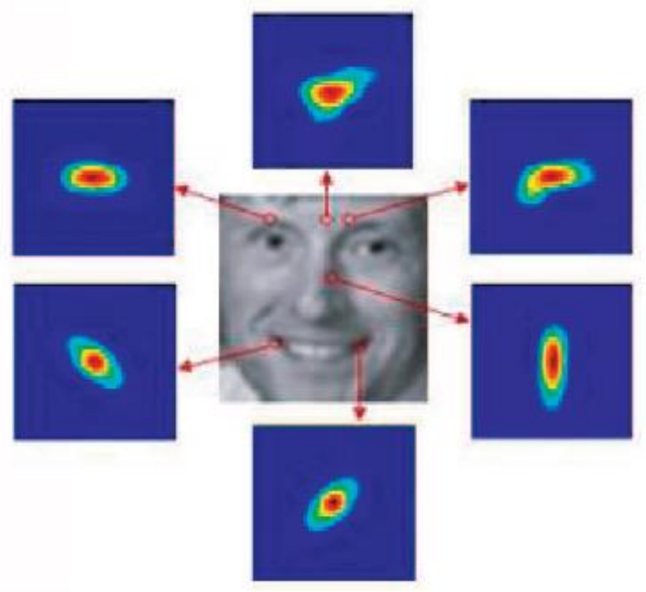
Hamming distance 51



Local Steering Kernels

Local Steering Kernels (LSKs)

- They are a non-linear combination of weighted distances between a pixel and its surrounding pixels.
- They exploit both spatial and edge detection information.
- One LSK vector per pixel is derived.
- LSKs are invariant to brightness & contrast variations and noise.



[SEO2010]

Local Steering Kernels

- Local Steering Kernels:

$$K(\mathbf{x}_l - \mathbf{x}) = \frac{\sqrt{\det(\mathbf{C}_l)}}{h^2} \exp\left\{-\frac{(\mathbf{x}_l - \mathbf{x})^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x})}{2h^2}\right\}.$$

  - $\mathbf{C}_l$: Covariance matrix of $k \times k$ neighboring pixel gradient matrix.
  - It rotates, elongates, and scales the Gaussian kernel along the local edge.

Gabor features

Simple-cells in V1 visual cortex area are ***orientation-selective***, responding to spatial intensity changes only along a certain orientation (and scale).

- Simple-cells can be modeled by ***Gabor functions***:

$$h(x,y)=\frac{1}{2\pi\sigma}\exp\left\{-\frac{1}{2\sigma^2}\left(x_r^2+r^2y_r^2\right)\right\}\exp\left\{i\left(\frac{2\pi x_r}{\lambda}+\varphi\right)\right\},$$

$$x_r = x\cos\theta + y\sin\theta,$$

$$y_r = -x\cos\theta + y\sin\theta.$$

- Gabor kernels can be convolved with an image to produce ***Gabor image features***.

Human Vision Model

Gabor function parameters:

- θ: Gabor filter normal orientation.
- φ: phase offset.
- σ: Gaussian standard deviation (scale).
- r: spatial aspect ratio defining the Gabor function ellipticity.
- λ: sinusoidal wavelength.
- φ: sinusoidal phase.

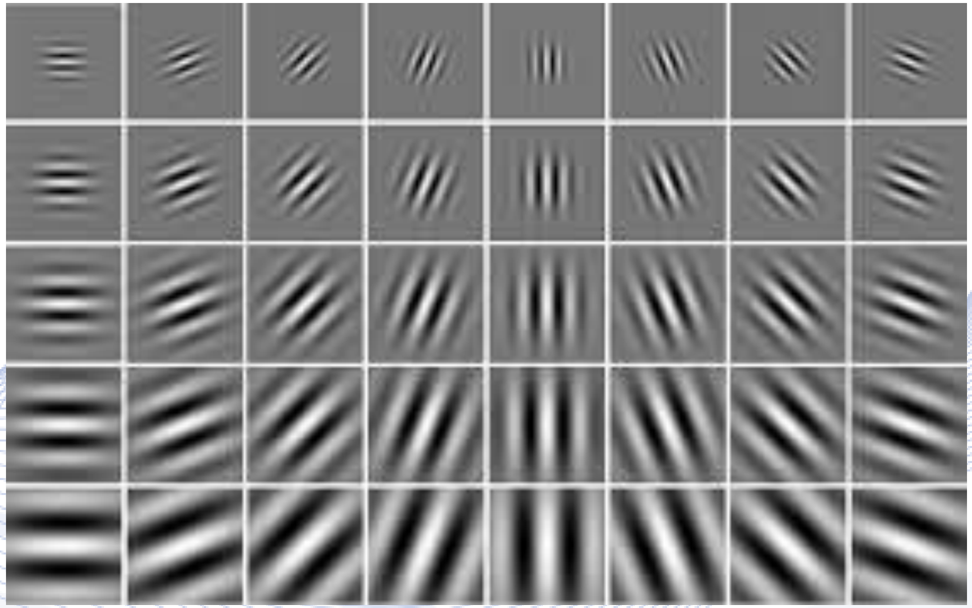
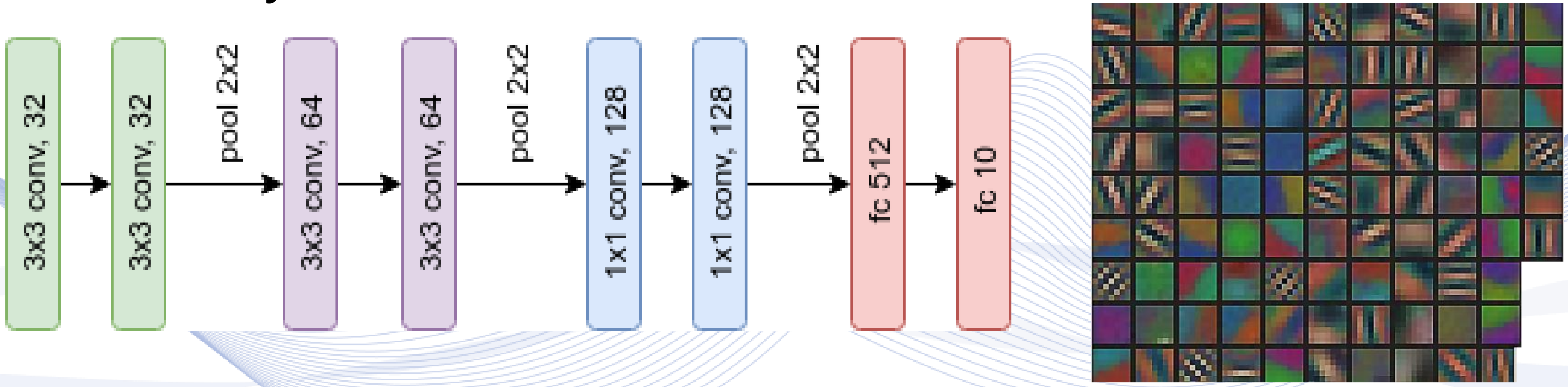


Image Features

Convolutional image features:

- Output of CNN convolutional layers.
- They encode local directional and color information.



a) Multilayer CNN architecture; b) Convolution kernels.

Bag-of-features

- ***Origin: texture recognition***

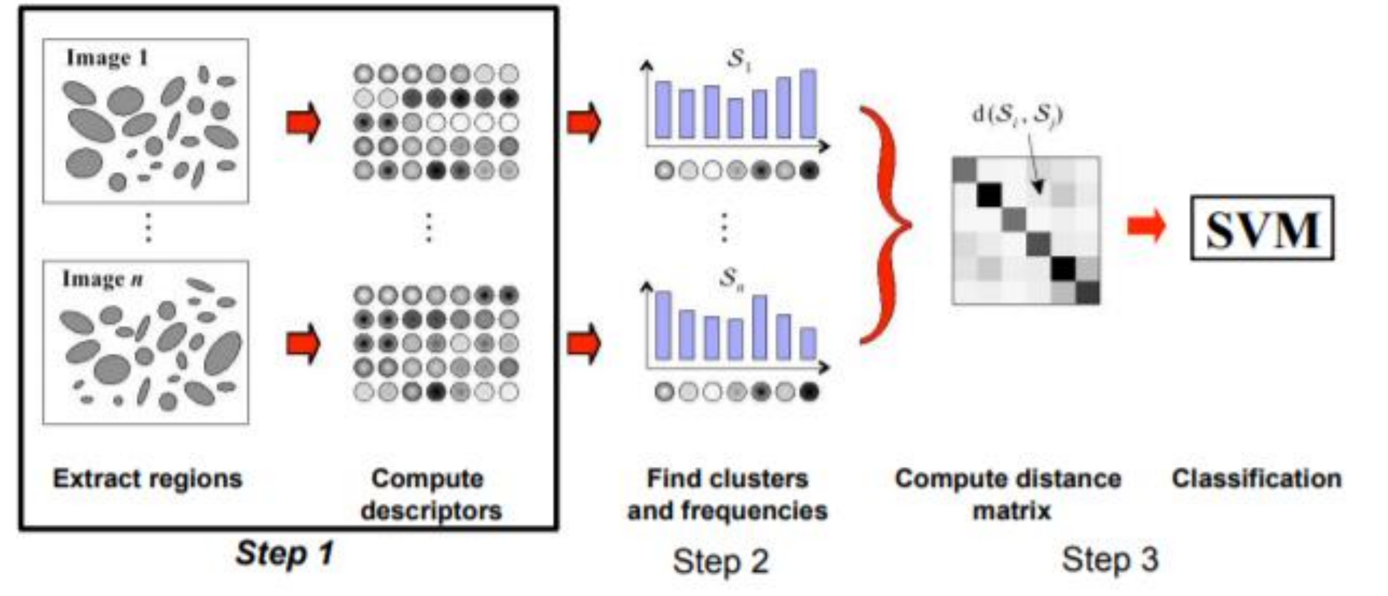
-Texture is characterized by the repetition of basic elements or ***textons***.

- ***Origin: bag-of-words***

-Orderless document representation: word frequencies in a ***dictionary***.

- It can be used for document classification.

Bag-of-features



Use of bag-of-words for image description and classification.

Bag-of-features

Step1: ***Feature detection and description.***

- Any feature descriptor can be used.
- Invariant descriptors are preferred.

- Dense descriptors
  - Interest points do not necessarily capture “all” features.
- Color-based descriptors
- Shape-based descriptors

Bag-of-features

Step 2: Quantization

- Feature descriptor clustering by:
  - K-means,
  - Gaussian mixture models.
- Assign a ‘visual word’ to each cluster.
  - Typically, the cluster centroid can be used.
- Hard/soft assignment of feature descriptors to a cluster.
- Build visual word frequency histogram.

Bag-of-features

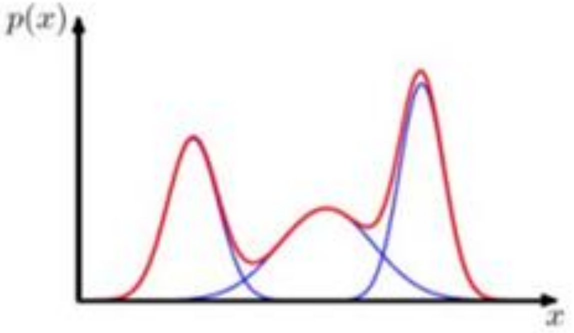
K-means clustering

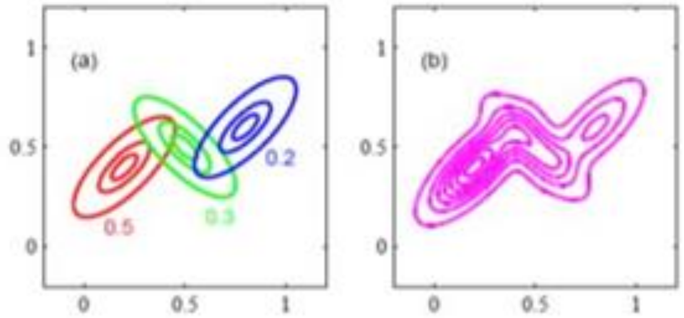
- It minimizes the sum of squared Euclidean distances between points xi and their nearest cluster centers m k

Gaussian mixture model (GMM)

- ***Mixture of Gaussians*** is weighted sum of Gaussians:

$$f_{\mathbf{X}}(\mathbf{x}) = \sum_i \pi_i \frac{1}{(2\pi)^{\frac{n}{2}} \det(\mathbf{C}_i)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T \mathbf{C}_i{}^{-1}(\mathbf{x}-\mathbf{m}_i)\right\}.$$





Bag-of-features

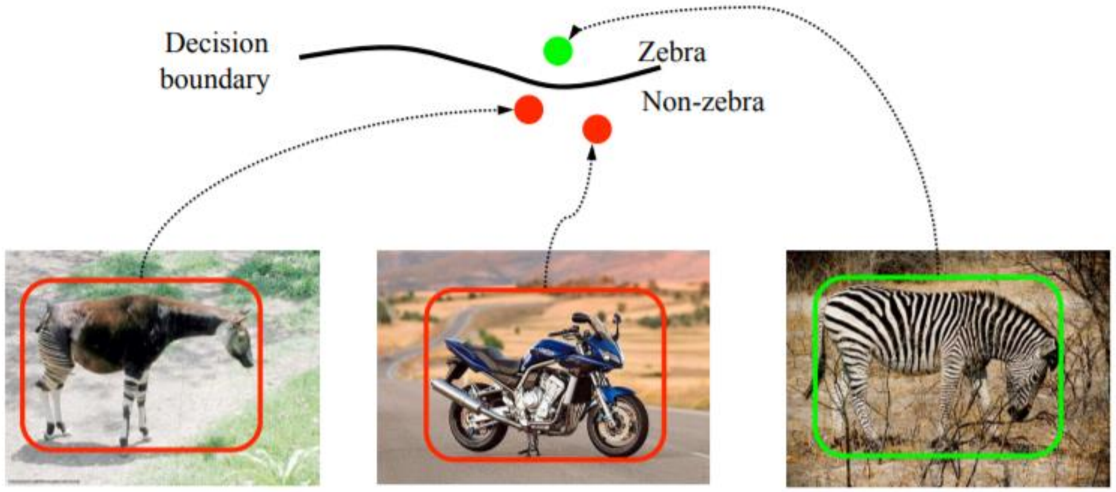
Hard or soft assignment

- *K*-means corresponds to hard assignment
  - Assign to the closest cluster center
  - Count number of descriptors assigned to a cluster.
- Gaussian mixture model for soft assignment
  - Estimate distance to all centers
  - Sum over number of descriptors.

Bag-of-features

Step 3: Classification

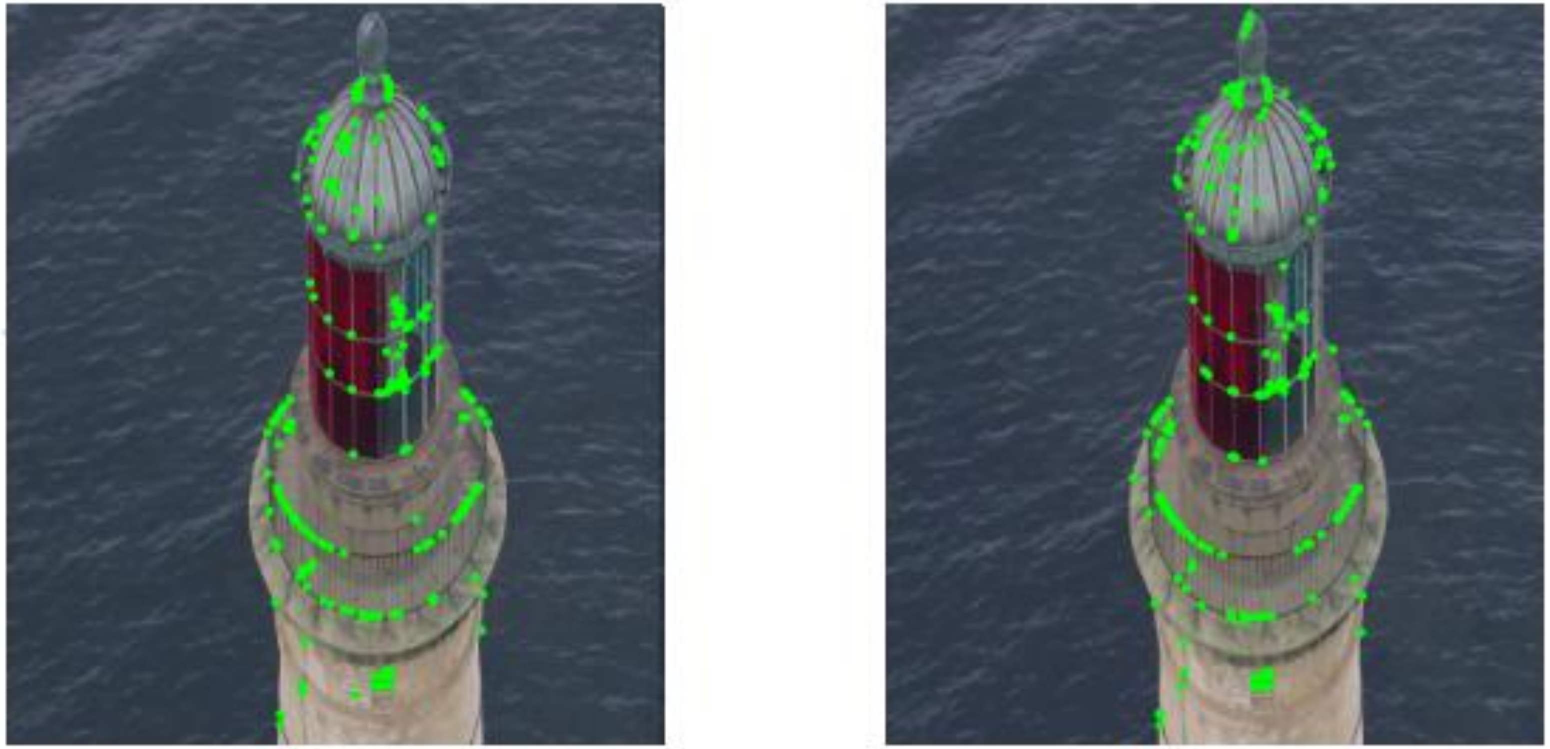
- Train/test a classifier assigning bag-of features representations of images to different image classes.



Feature Correspondence

- 2D point correspondence
  - Pixel correspondence
  - Feature point correspondence.
- Point correspondences can be estimated between different views of the same object.

Feature Correspondence



Feature Extraction

- ***2D point correspondence***:
  - Pixel correspondence
  - Feature correspondence.
- Local feature: a small image region having interesting spatial characteristics (e.g., corner).
  - It can be described by a N-dimensional vector.
- Typically (not always), a feature detector/descriptor tries to produce description vectors invariant to several image transformations.

Feature Matching Algorithms

- ***Uniqueness constraint***: a point in one view has at most one corresponding match in the other view.

- ***Continuity constraint***: adjacent feature points in one view should correspond to adjacent features in the other view.

- ***Topological constraint***: the relative position of 3D points remains unaltered in their projections to all views.

Artificial Intelligence & Information Analysis Lab

Feature Matching Algorithms

- Area-based matching algorithms are the oldest matching methods, used mainly for low-level feature matching.

- Matching of two feature points is based on the minimization of some distance measure of the respective local image windows.

Feature Matching Algorithms

- Assuming feature points $\mathbf{p}_l = [x_l, y_l]^T$ and $\mathbf{p}_r = [x_r, y_r]^T$ to be matched, the grayscale intensity images $f_l(x, y)$ and $f_r(x, y)$ in a $L = (2N + 1) \times (2M + 1)$ local neighborhood window centered around these points are going to be compared.

Feature Matching Algorithms

Distance-based matching measures which can be used:

- ***Mean Absolute Error*** (***MAE***) or L_1 norm:

$$MAE(\mathbf{p}_l, \mathbf{p}_r) = \sum_{i=-N}^{N} \sum_{j=-M}^{M} |f_l(x_l + i, y_l + j) - f_r(x_r + i, y_r + j)|.$$

- ***Mean Square Error*** (***MSE***) or L_2 norm:

$$MSE(\mathbf{p}_l, \mathbf{p}_r) = \sum_{i=-N}^{N} \sum_{j=-M}^{M} \left(f_l(x_l + i, y_l + j) - f_r(x_r + i, y_r + j)\right)^2.$$

Feature Matching Algorithms

- Correlation-based similarity measures which can be used:
  - ***Normalized cross-correlation*** (***NCC***):

$$NCC(\mathbf{p}_l, \mathbf{p}_r) = \frac{\sigma_{lr}^2(p_l, p_r)}{\sqrt{\sigma_l^2(p_l)\sigma_r^2(p_r)}}.$$

where:

$$\sigma_{lr}^2(\mathbf{p}_l, \mathbf{p}_r) = \frac{1}{(2N+1)(2M+1)} \sum_{i=-N}^{N} \sum_{j=-M}^{M} \left(f_l(x_l+i, y_l+j) - \bar{f_l}\right) \cdot \left(f_r(x_r+i, y_r+j) - \bar{f_r}\right).$$

Feature Matching Algorithms

$$\sigma_l^2(\mathbf{p}_l) = \frac{1}{(2N+1)(2M+1)} \sum_{i=-N}^{N} \sum_{j=-M}^{M} \left(f_l(x_l+i, y_l+j) - \bar{f_l}\right)^2.$$

$$\sigma_r^2(\mathbf{p}_r) = \frac{1}{(2N+1)(2M+1)} \sum_{i=-N}^{N} \sum_{j=-M}^{M} \left(f_r(x_r+i, y_r+j) - \bar{f_r}\right)^2.$$

Feature Matching Algorithms

- The somewhat more stable than NCC is the ***Modified Normalized Cross-Correlation*** (***MNCC***):

$$MNCC(\mathbf{p}_l, \mathbf{p}_r) = \frac{2\sigma_{lr}^2(p_l, p_r)}{\sigma_l^2(p_l) + \sigma_r^2(p_r)}.$$

Feature Matching Algorithms

- Other image feature characteristics which can be used for feature matching:
  - Edge attributes (e.g., edge orientation, location, intensity difference between the two sides of the edges).
    - They may suffer from occlusion problems.
  - Corner attributes (e.g., coordinates):
    - Harris detector.
  - Orientation of line segments and coordinates of the end or/and mid points.
    - Detection methods not robust against noise.

Feature Matching Algorithms

- Curve segments.
  - Not frequently used because of high computational complexity and matching ambiguities.
- Curve attributes (e.g., turning points).
- Circles.
- Ellipses.
- Polygonal regions.

Feature Matching Algorithms

- Most of the feature-based stereo or multiview matching systems use a combination of features and compare the descriptor token vectors containing the attributes of each feature point.
  - Edges, curves, surface and region patches.

Feature Matching Algorithms

- General matching approach based on a ***similarity metric*** between a token vector pair – nearest neighbor search:
  - If $\mathbf{x}, \mathbf{y}$ feature descriptor vectors and $\mathbf{w}$ the weight vector of the feature token type, then the similarity is given by:

$$S = \frac{1}{||\mathbf{w}^T(\mathbf{x} - \mathbf{y})||}.$$

  - $||\cdot||$ can be the Euclidean distance metric and $\mathbf{w}$ can be omitted if every feature characteristic is equally important.
  - We search for a pair of feature descriptors maximizing the similarity.

Feature Matching Algorithms

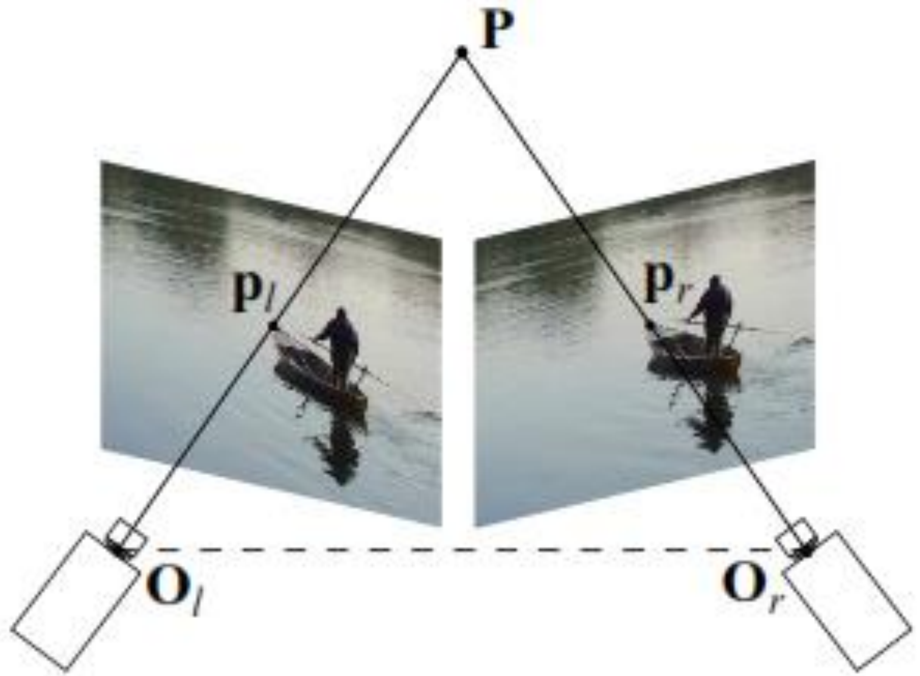
- ***Naïve nearest neighbor search***: a brute force approach.
  - Calculate the similarity of each feature point on one image with every other feature point on the other image and match the pair with maximum similarity.

- ***Best-bin-first search***: a faster but approximate method - modification of kd-tree search.

Image Feature Applications

- Applications:
- Object description and recognition
- Feature correspondence
  - Image registration
  - Object tracking
  - Motion estimation
  - Disparity estimation
  - 3D object shape reconstruction.

Stereo



3D shape reconstruction

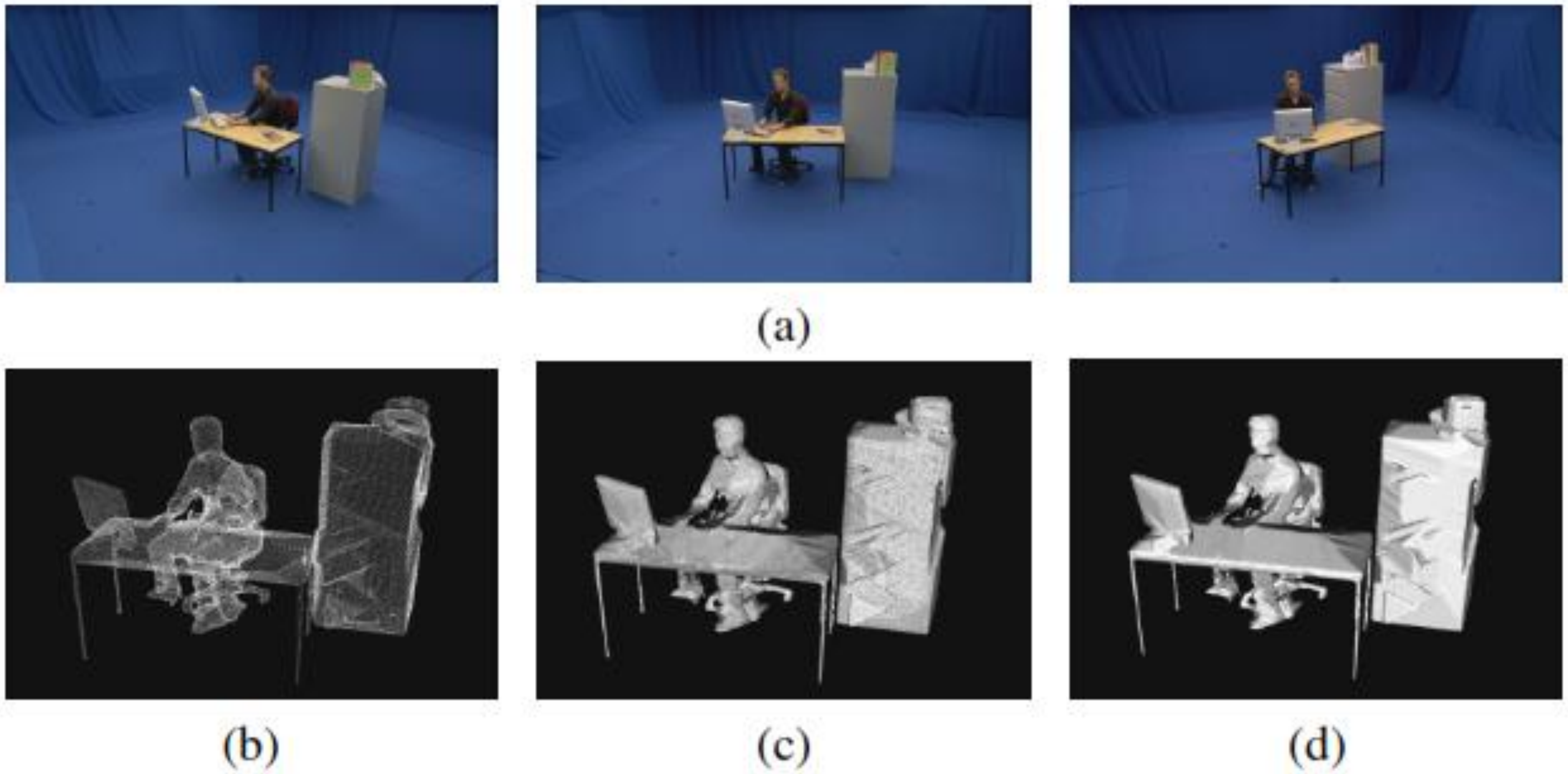
- Image feature matching can be used in 3D shape reconstruction, given multiple 2D scene views.
- The multiple views may come from different view points / cameras:
  - stereoscopic / binocular view, trinocular view, Multiview imaging .
- The may come from a moving camera, if the scene is static: ***Structure-from-Motion*** (***SfM***).

Feature Matching Algorithms

- ***Homologous image features***: projections of the same natural 3D point on each camera view, after ***feature matching***.

- Commonly used constraints for reduction of the search space for feature correspondences:
  - ***Epipolar constraint***: when the projection geometry is known, search for a corresponding feature point can be restricted to the epipolar line on the other image of the stereo pair.

3D shape reconstruction



Object tracking

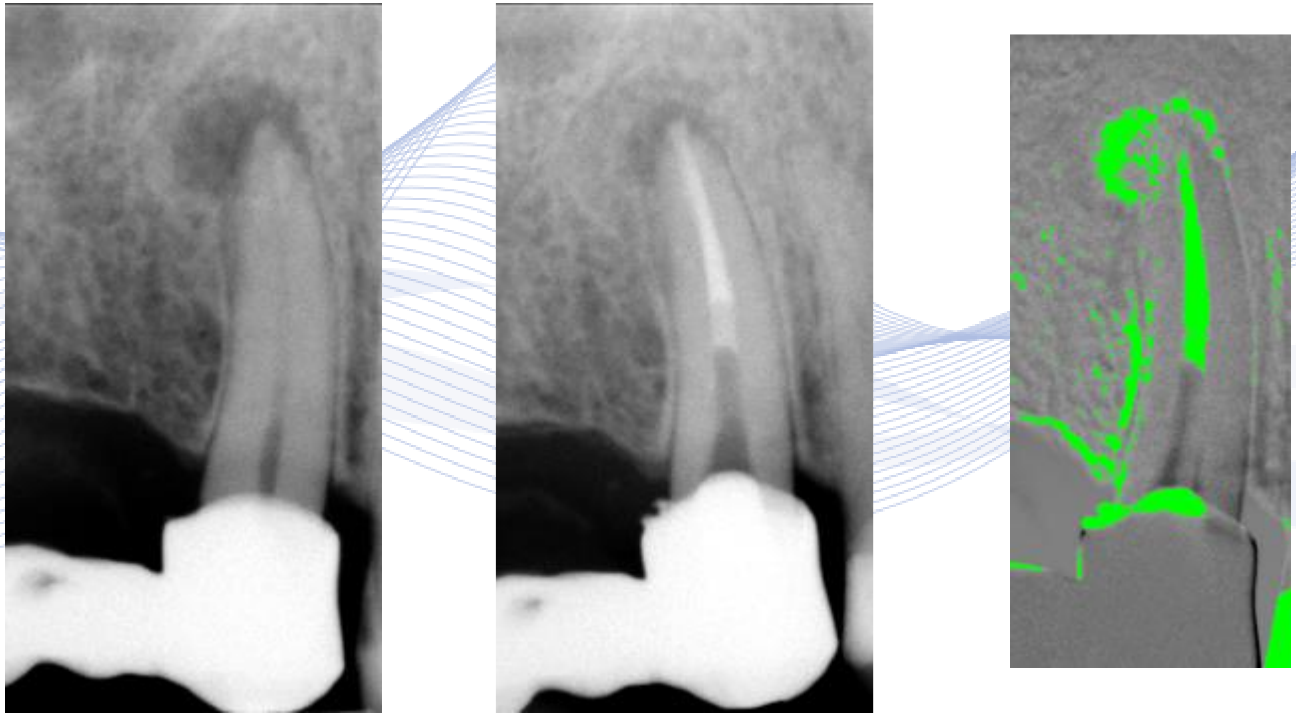
Image features in object tracking:

- During tracking, input is the current object ROI.
- Visual tracker learns a tracking model.
- It can be the grayscale or RGB values of each ROI pixel or calculated ROI features.
- Histogram of oriented gradients (HOG) is a very useful feature descriptor.

Image registration

- Image feature matching.
- It can be used for 2D image registration.

Subtractive radiography.



Artificial Intelligence & Information Analysis Lab

Image registration

- Image feature matching for image mosaicking.

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[SZE2011] R.Szelinski, “ Computer Vision ” , Springer 2011
[PIT2017] I. Pitas, “Digital video processing and analysis ” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television ” , Createspace/Amazon, 2013.
[PIT2000] I. Pitas, Digital Image Processing Algorithms and Applications, J. Wiley, 2000.
[NIK2000] N. Nikolaidis and I. Pitas, 3D Image Processing Algorithms, J. Wiley, 2000.
[APOLLO] http://apolloscape.auto/
[RES] https://www.researchgate.net/figure/1-directional-2-directional-and-the-lic-like-texture-patterns-as-in-KHSI03b_fig8_228697126

References

[SEO2010] H.J. Seo and P. Milanfar, “Training-free, Generic Object Detection using Locally Adaptive Regression Kernels”, *IEEE Trans. On PAMI*, vol. 32, no.9, pp. 1688-1704, Sept. 2010

https://www.slideshare.net/AnirudhKanneganti/hog-122762396

https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html

[RUB2011] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. ORB: an efficient alternative to SIFT or SURF, ICCV 2011

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr