Quality & Features

Dr. Tushar Sandhan

What is the need

- Quantitative assessment
  - performance evaluation of image processing algorithms
    - e.g. denoising, compression
- Quantitative achievement
  - performance improvement via optimization based methods
    - e.g. enhance images to minimize MSE or improve PSNR

What is the need

- Quantitative assessment
  - performance evaluation of image processing algorithms
    - e.g. denoising, compression
- Quantitative achievement
  - performance improvement via optimization based methods
    - e.g. enhance images to minimize MSE or improve PSNR

PSNR

- Full reference measure
  - need a clear GT or reference image

- Generic error
  - $p-$ norm
  - minkowski norm

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N} |e_i|^p \right)^{1/p} \quad \text{where} \quad e_i = x_i - y_i$$

- Mean sq error (MSE)

- PSNR
  - L is dynamic range

PSNR

- Full reference measure
  - need a clear GT or reference image

- Generic error
  - $p-$ norm
  - minkowski norm

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N} |e_i|^p \right)^{1/p} \quad \text{where} \quad e_i = x_i - y_i$$

- Mean sq error (MSE)

$$\mathrm{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2$$

- PSNR
  - L is dynamic range

PSNR

- Full reference measure
  - need a clear GT or reference image

- Generic error
  - $p-$ norm
  - minkowski norm

$$d_p(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{N} |e_i|^p\right)^{1/p} \quad \text{where} \quad e_i = x_i - y_i$$

- Mean sq error (MSE)

$$\mathrm{MSE}(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

- PSNR
  - L is dynamic range

$$\mathrm{PSNR} = 10\log_{10}\frac{L^2}{\mathrm{MSE}}$$

MSE

- Good metric for optimization based methods
  - MSE: convex and differentiable
  - parameter-free, memoryless
  - energy minimization methods: relation to energy

- Uniformity with data communication signal measurements

- Distance metric

MSE

- Good metric for optimization based methods
  - MSE: convex and differentiable
  - parameter-free, memoryless
  - energy minimization methods: relation to energy

- Uniformity with data communication signal measurements

- Distance metric

  - nonnegativity: $d_p(\mathbf{x}, \mathbf{y}) \geq 0$
  - symmetry: $d_p(\mathbf{x}, \mathbf{y}) = d_p(\mathbf{y}, \mathbf{x})$
  - identity: $d_p(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$
  - triangular inequality: $d_p(\mathbf{x}, \mathbf{z}) \leq d_p(\mathbf{x}, \mathbf{y}) + d_p(\mathbf{y}, \mathbf{z})$

PSNR

reference image

Image features

- Image representation
  - summarize image content via set of numbers (e.g. a vector)
  - capture important image properties
    - object recognition
    - image matching
    - segmentation (via supervised learning)

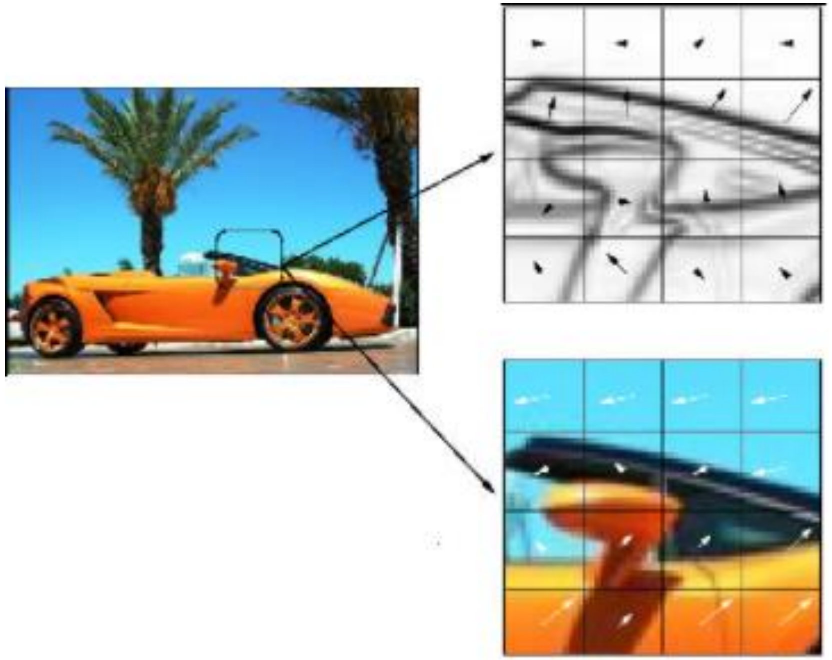
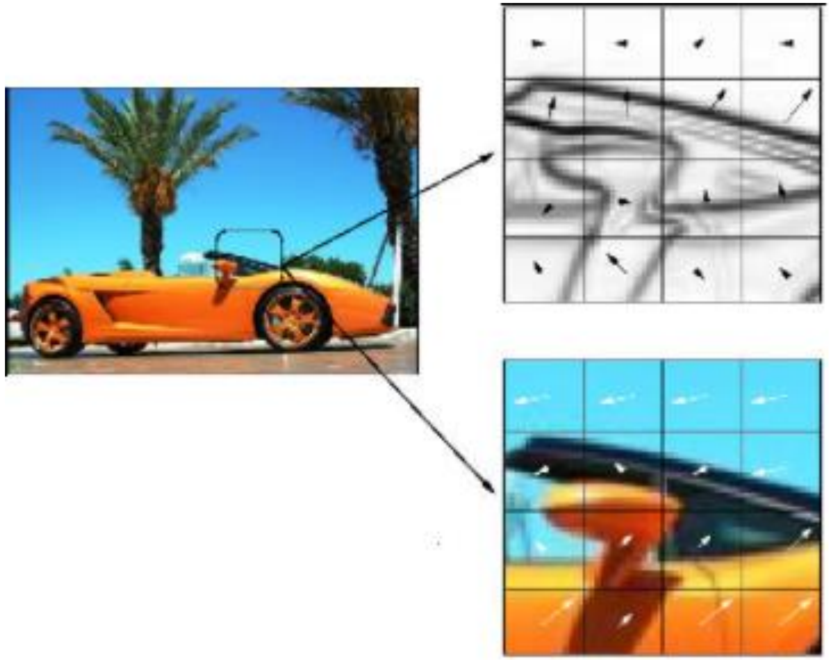


Image features

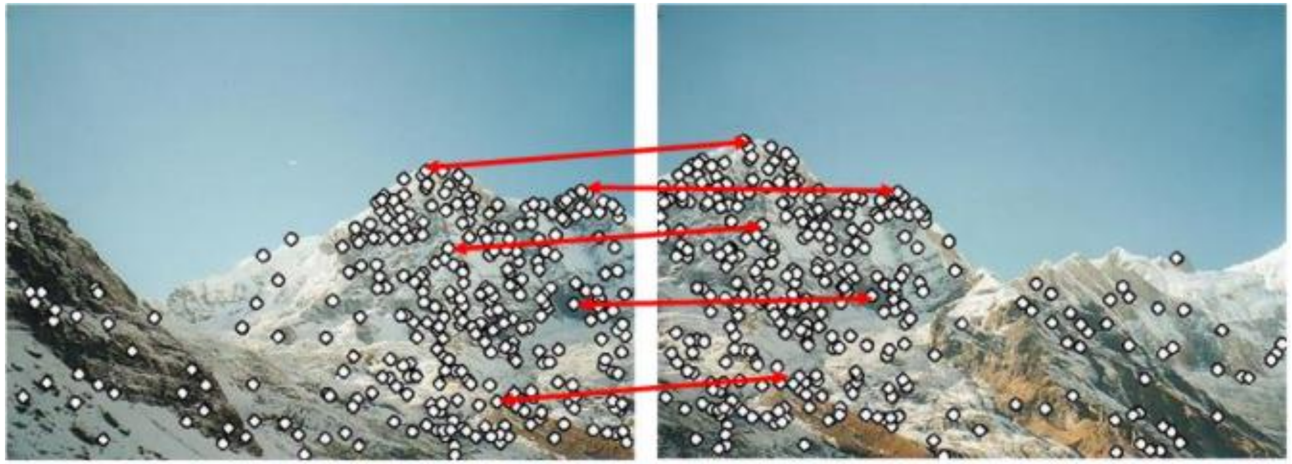
- Image representation
  - summarize image content via set of numbers (e.g. a vector)
  - capture important image properties
    - object recognition
    - image matching
    - segmentation (via supervised learning)



- Type of features
  - local, global
  - capture only certain property
    - texture, color, shape
    - deformation, object relative relations
    - motion

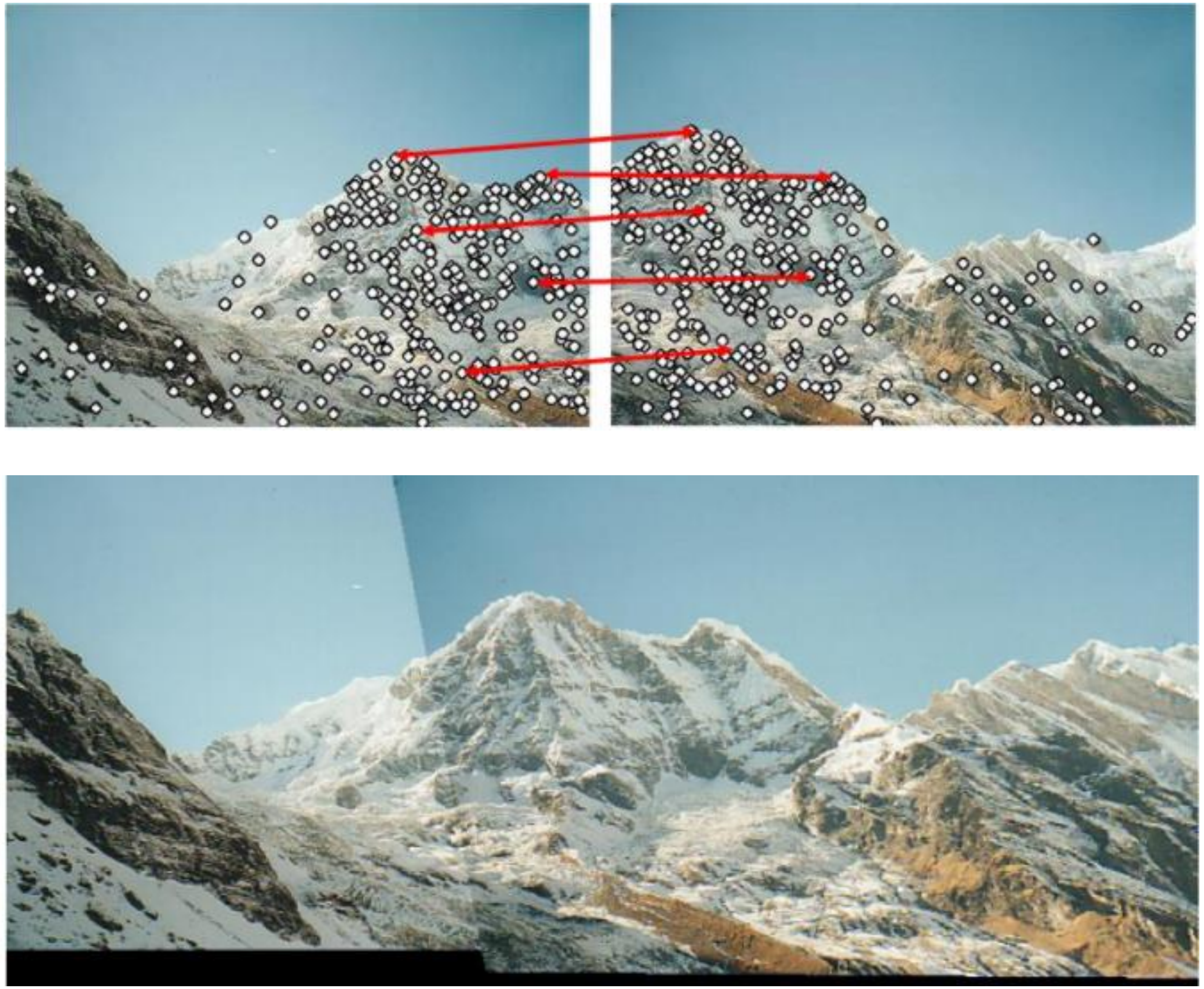
Feature matching

- Panorama



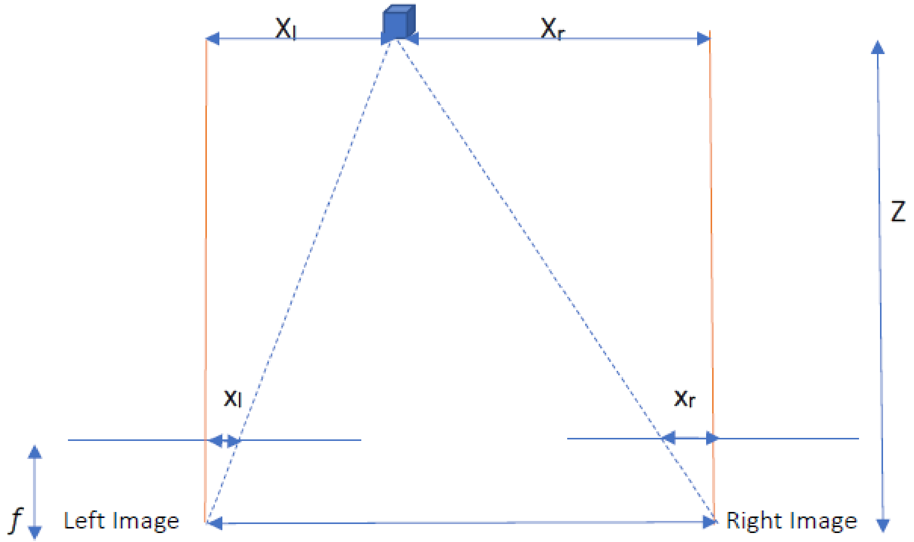
Feature matching

- Panorama



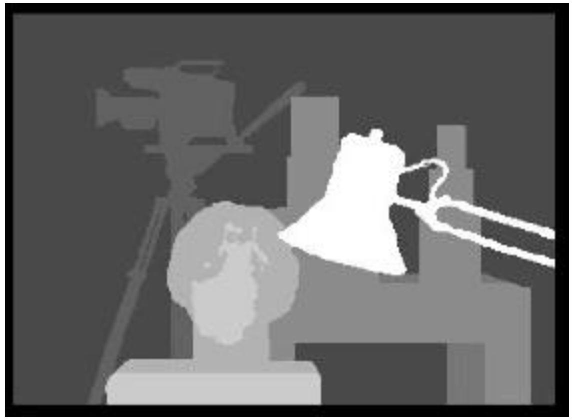
Feature matching

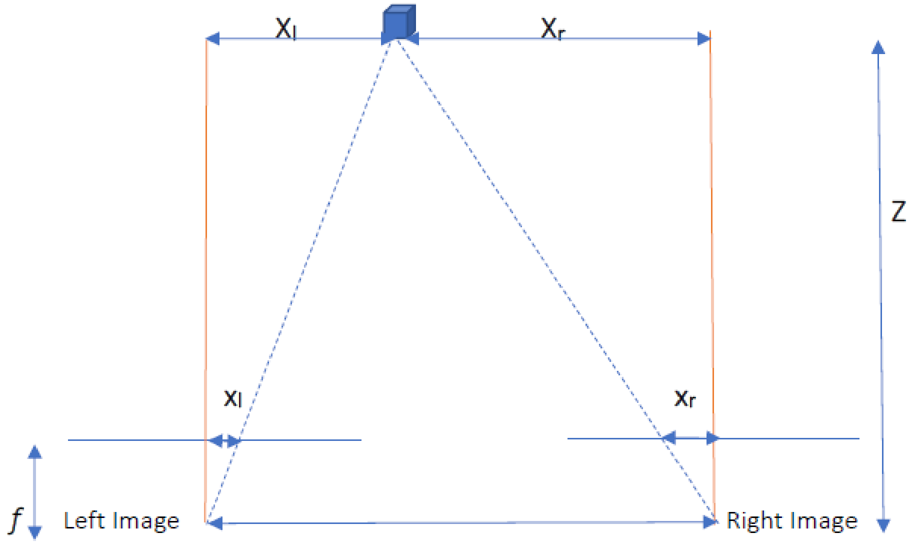
- Depth estimation



Feature matching

- Depth estimation





Feature matching

- Object tracking

courtesy: Au-Zone Tech

Feature matching

- Segmentation

Image features

- Feature detector
  - for a given image, outputs interesting locations (e.g. x, y)
  - tells nothing about the image properties at that region
  - capture important regions
    - corner detector

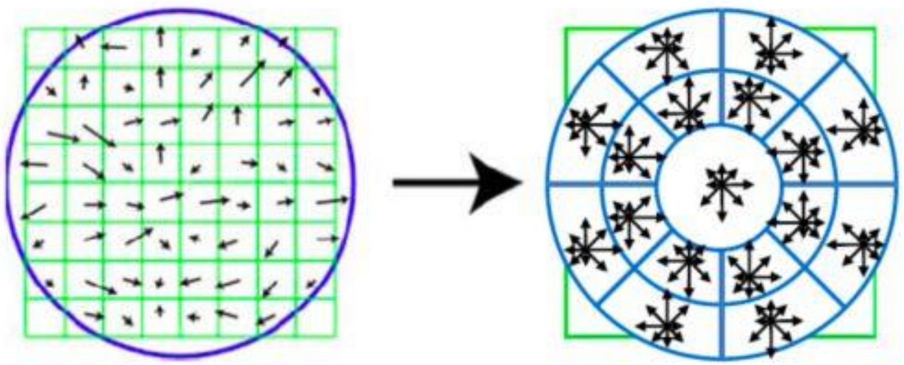
Image features

- Feature detector
  - for a given image, outputs interesting locations (e.g. x, y)
  - tells nothing about the image properties at that region
  - capture important regions
    - corner detector

- Feature descriptor
  - for a given image, outputs interesting properties via feature vector
  - encode interesting info into a series of stable numbers
    - stability in the sense that those numbers do not change drastically over image transformations (invariant)
    - e.g. scale, rotation invariance
  - capture important properties of regions
    - Local binary pattern

Image features

- Feature detector
  - for a given image, outputs interesting locations (e.g. x, y)
  - tells nothing about the image properties at that region
  - capture important regions
    - corner detector



(a) image gradients (b) keypoint descriptor

- Feature descriptor
  - for a given image, outputs interesting properties via feature vector
  - encode interesting info into a series of stable numbers
    - stability in the sense that those numbers do not change drastically over image transformations (invariant)
    - e.g. scale, rotation invariance
  - capture important properties of regions
    - Local binary pattern

Image features

- Without feature detector
  - uniform sampling

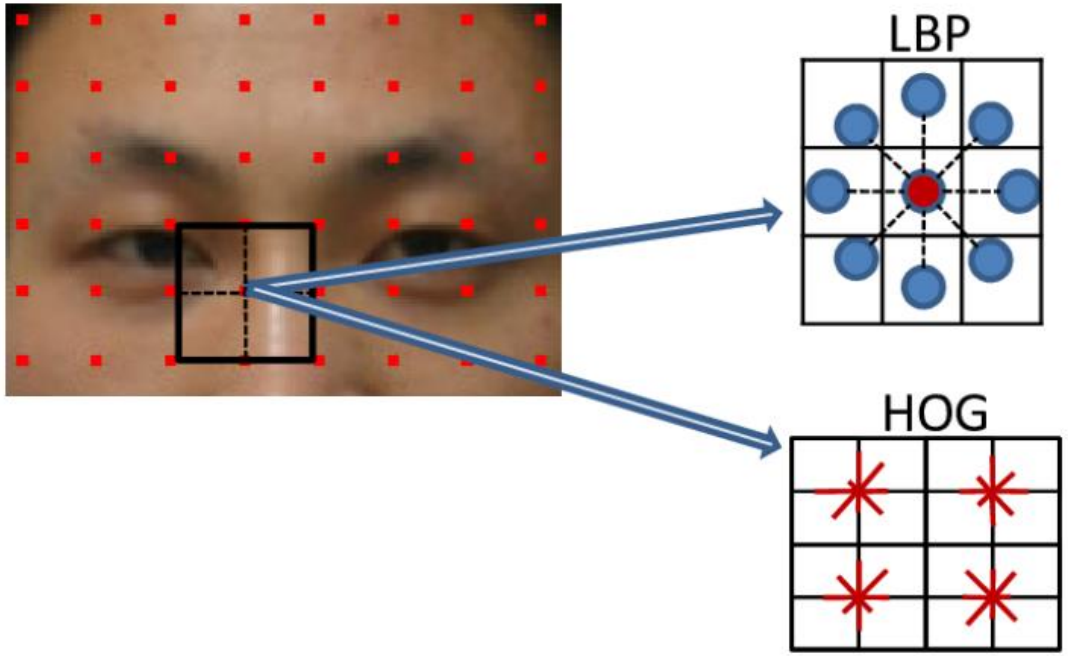
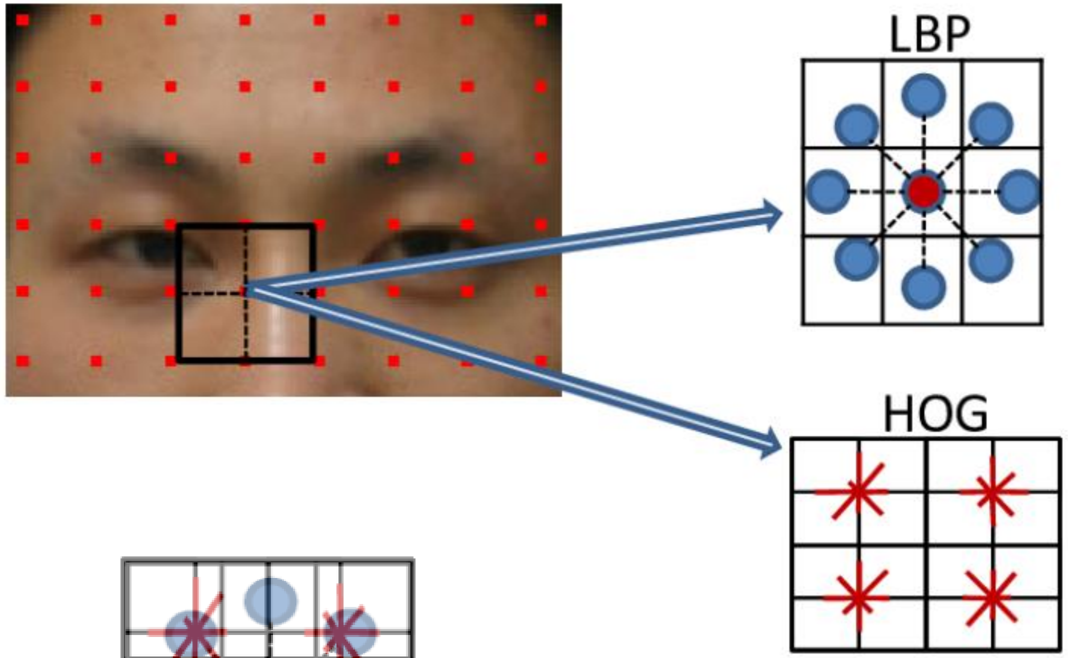
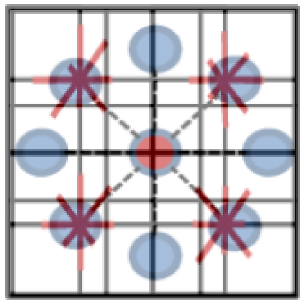


Image features

- Without feature detector
  - uniform sampling



- Feature combination
  - local to global transition



Feature detector

- Interest points
  - robust to
    - noise
    - distortions
    - transformations
  - distinctive
  - why are they located mostly at
    - line endings
    - intersection of edges
    - local extrema points

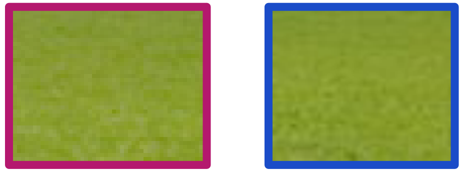
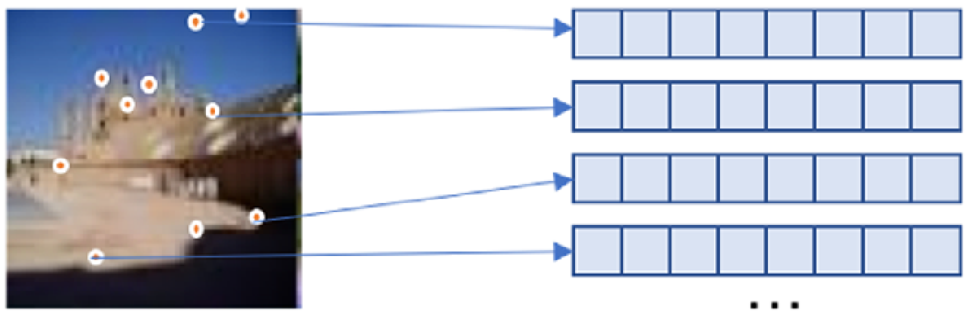


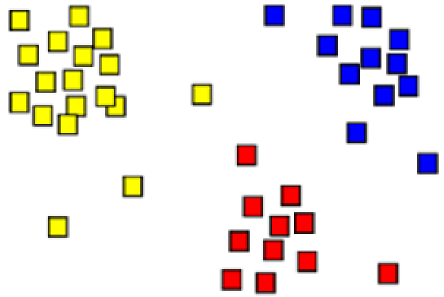
Image features

Extract keypoints

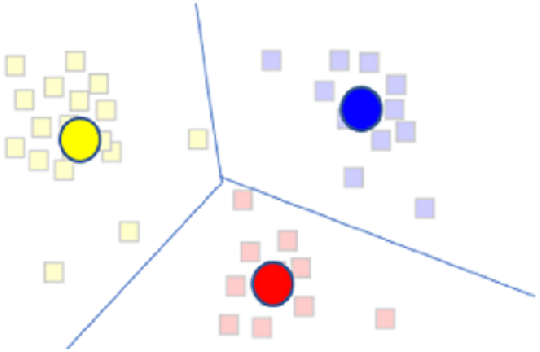
Feature descriptors



Clustering

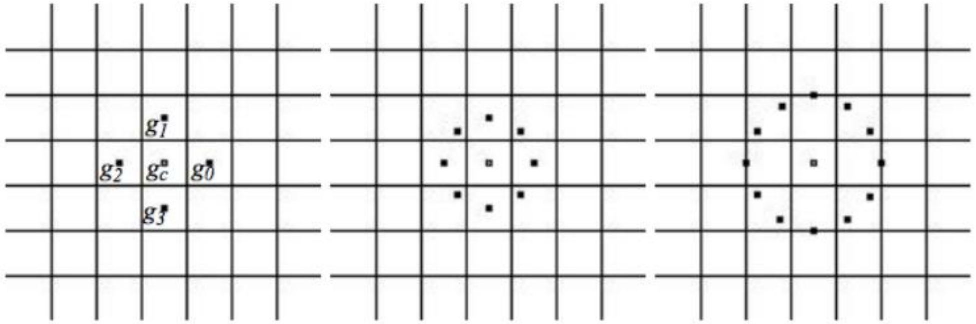


Visual vocabulary



LBP

- Local Binary Pattern
  - texture and local pattern detection
  - textures have no specific definition
    - complex patterns having more sub-patterns



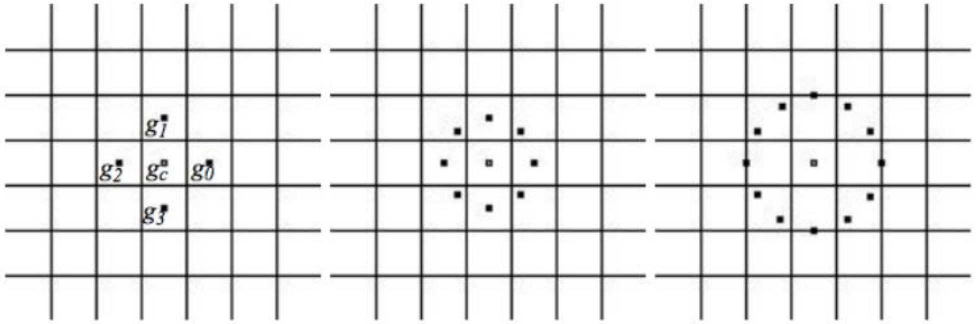
courtesy: Ojala

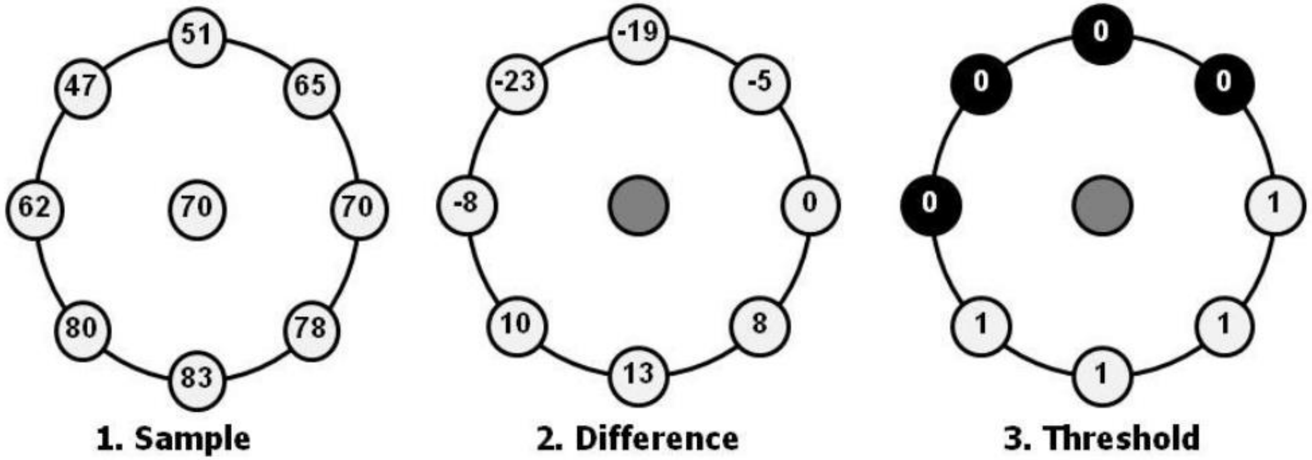
LBP

- Local Binary Pattern
  - texture and local pattern detection
  - textures have no specific definition
    - complex patterns having more sub-patterns

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, & if\ x \geq 0; \\ 0, & otherwise. \end{cases}$$





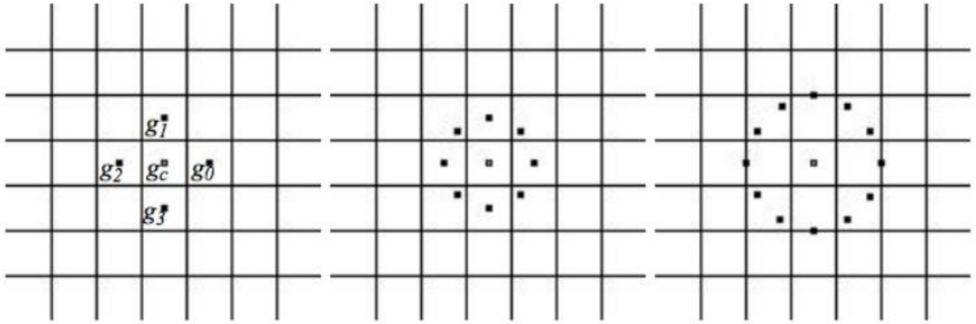
courtesy: Ojala

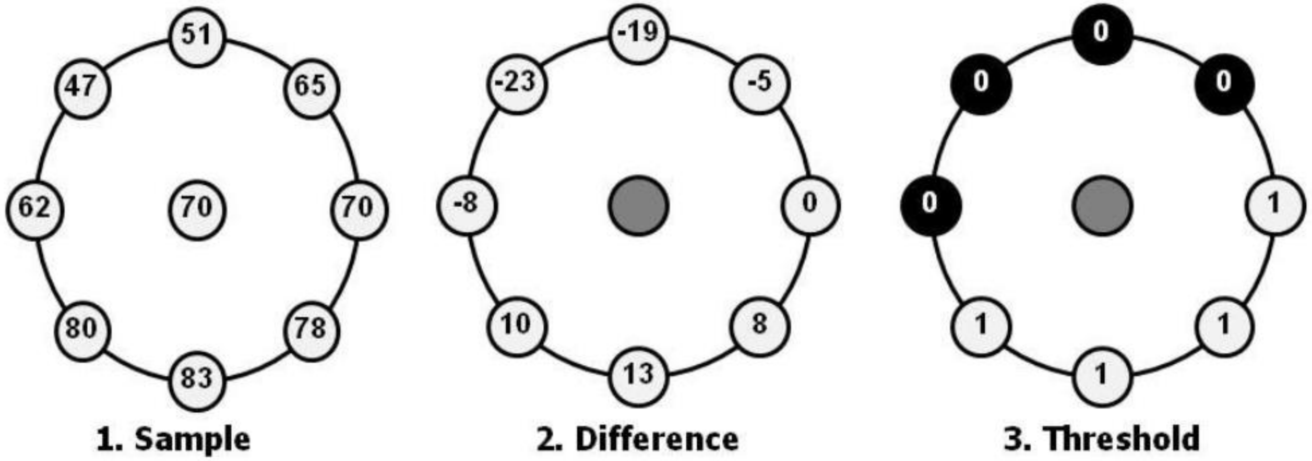
LBP

- Local Binary Pattern
  - texture and local pattern detection
  - textures have no specific definition
    - complex patterns having more sub-patterns

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, \; if \; x \geq 0; \\ 0, \; otherwise. \end{cases}$$





1*1 + 1*2 + 1*4 + 1*8 + 0*16 + 0*32 + 0*64 + 0*128 = **15**

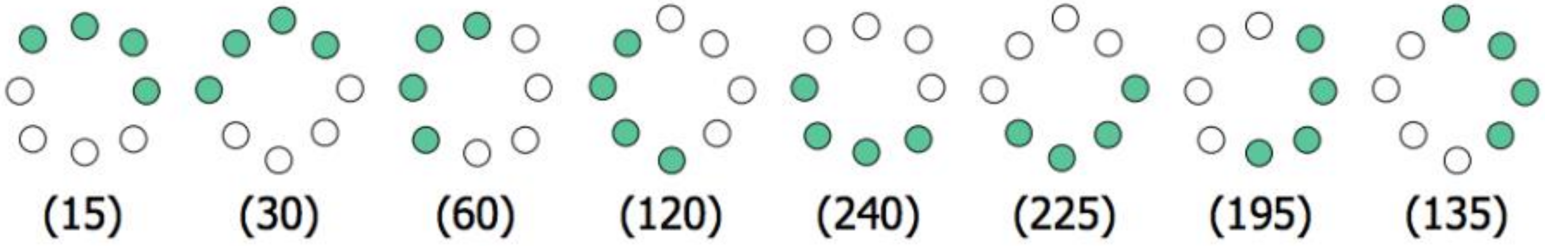
courtesy: Ojala

LBP

- Invariant to
  - ilumination
    - shadow, reflection, brightness
    - relative difference between intensities remain same
  - rotation?

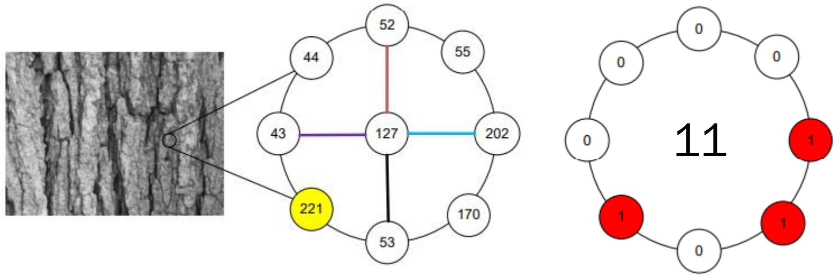
LBP

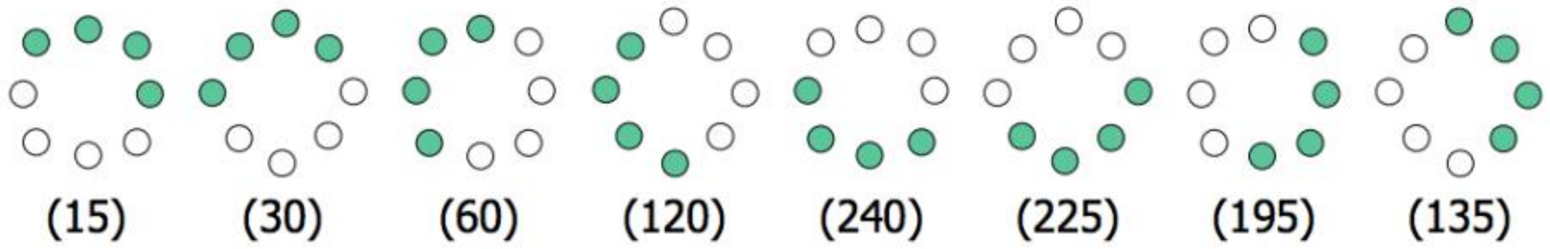
- Invariant to
  - ilumination
    - shadow, reflection, brightness
    - relative difference between intensities remain same
  - rotation?



LBP

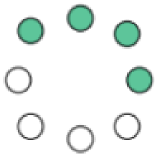
- Invariant to
  - ilumination
    - shadow, reflection, brightness
    - relative difference between intensities remain same
  - rotation?

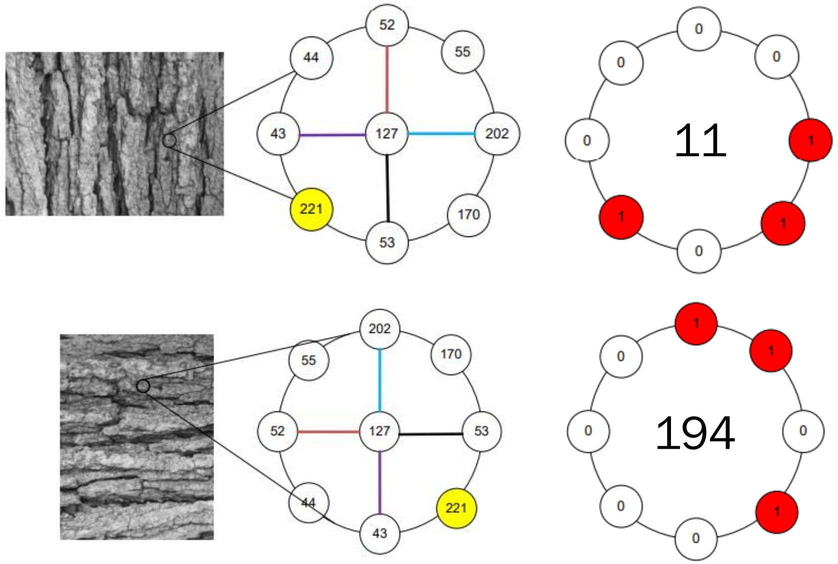




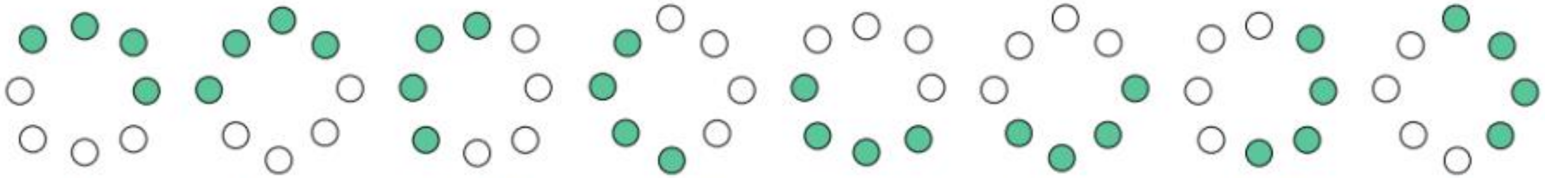
LBP

- Invariant to
  - ilumination
    - shadow, reflection, brightness
    - relative difference between intensities remain same
  - rotation?





edge (15)



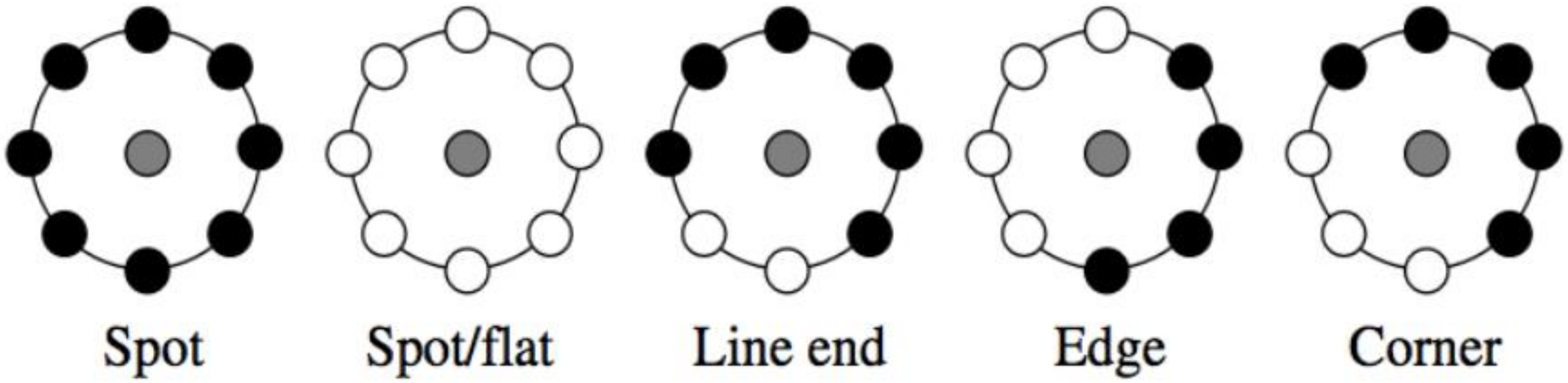
(15) (30) (60) (120) (240) (225) (195) (135)

LBP

- Local Binary Pattern
  - 8 neighborhood gives 256 possible LBP codes
    - each pixel gets one of the codes
  - LBP histogram 256D
    - probability of occurrence of each LBP code

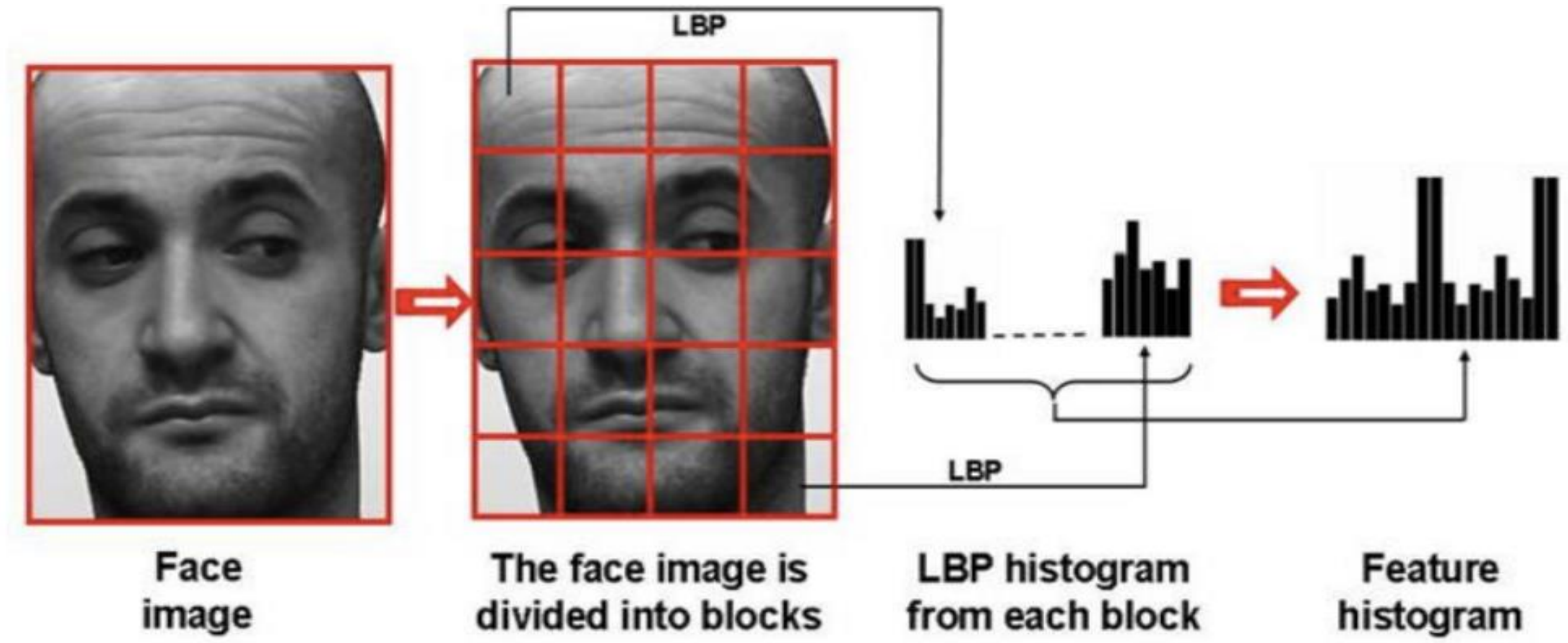
LBP

- Local Binary Pattern
  - 8 neighborhood gives 256 possible LBP codes
    - each pixel gets one of the codes
  - LBP histogram 256D
    - probability of occurrence of each LBP code

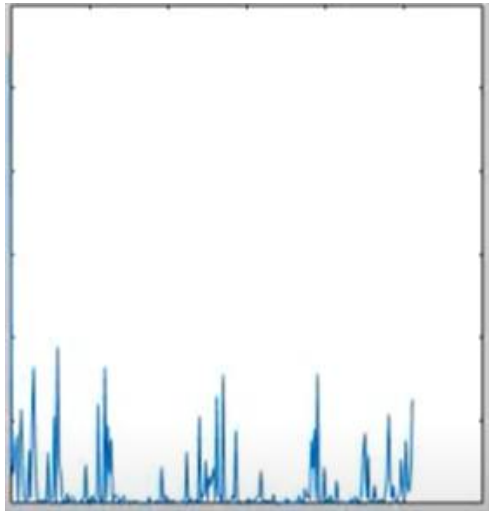


LBP

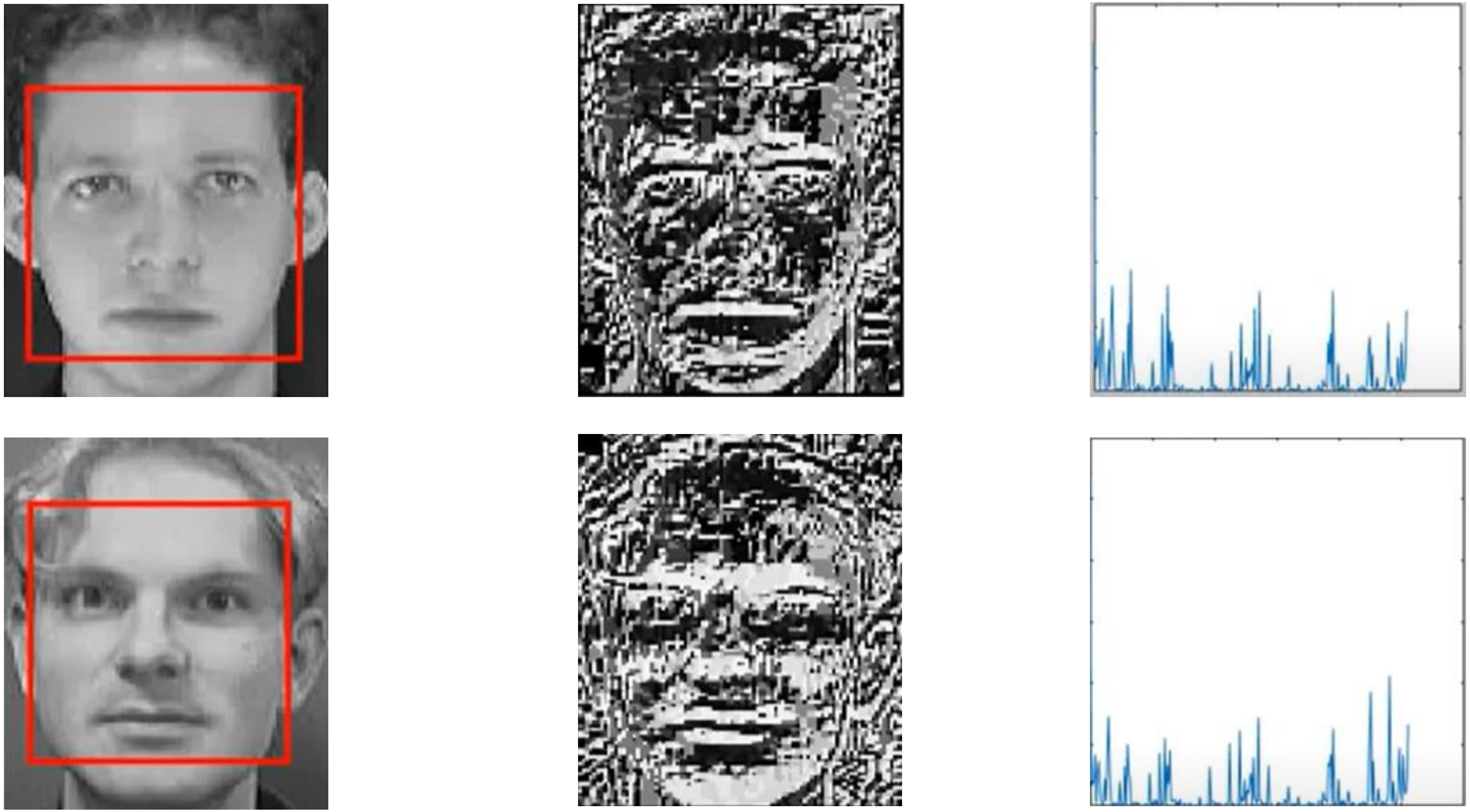
- LBP to global descriptor



LBP

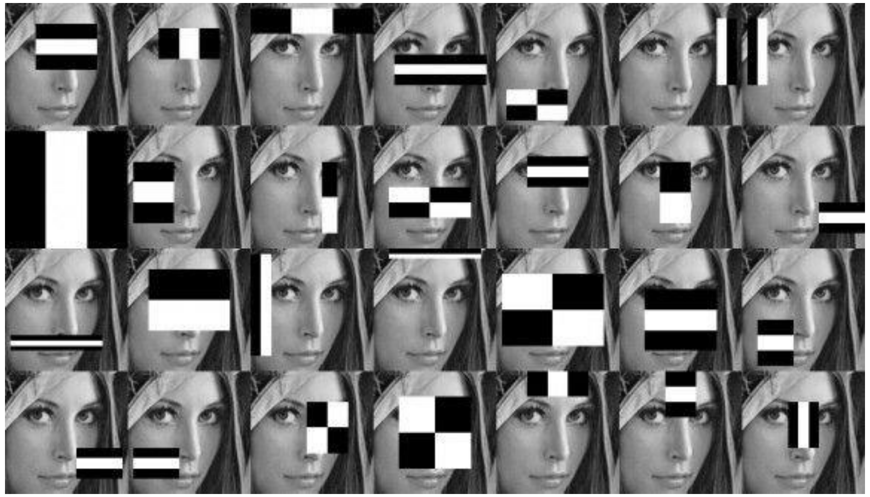


LBP

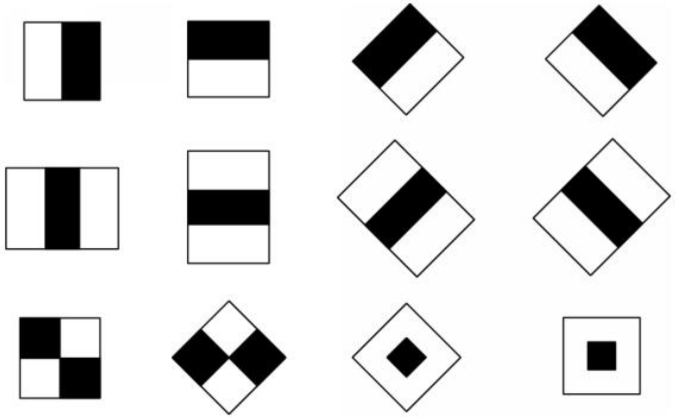


Haar features

- Face detection



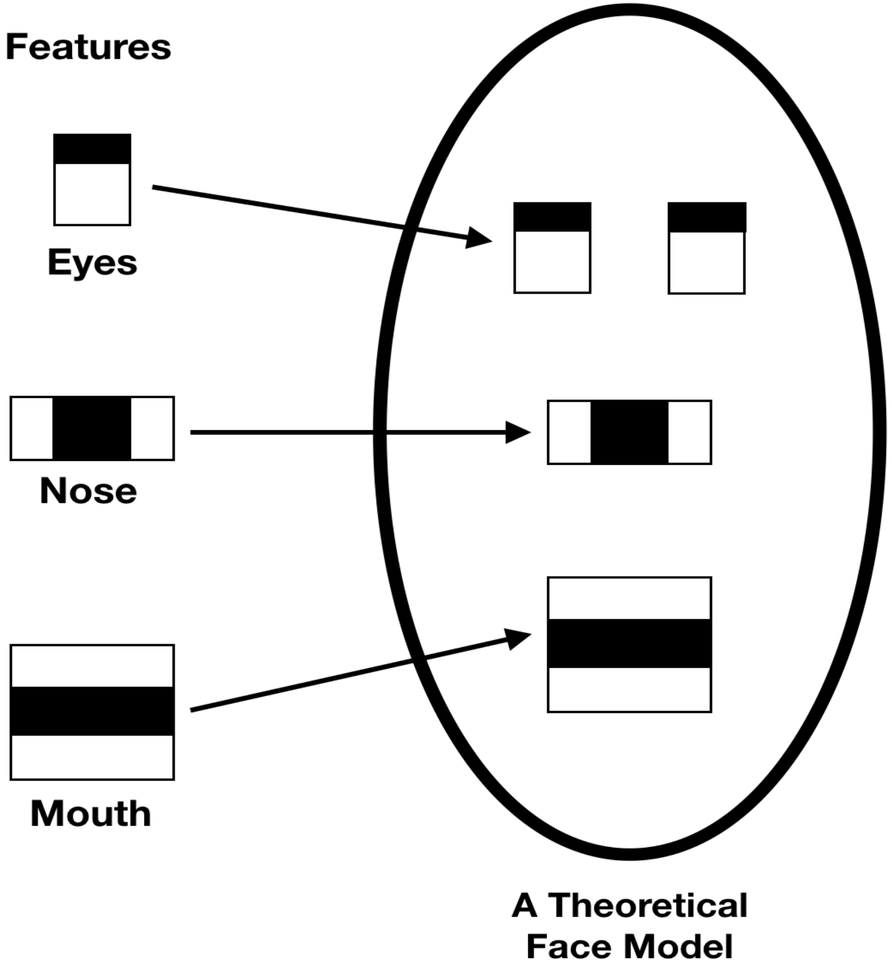
Haar feature: $f(x,y) = \sum I(R_{white}) - \sum I(R_{black})$



Haar filters
(based on Haar wavelets)

Conclusion

- Statistical descriptors
- LBP
- Haar



☺