Basic Image Processing Algorithms

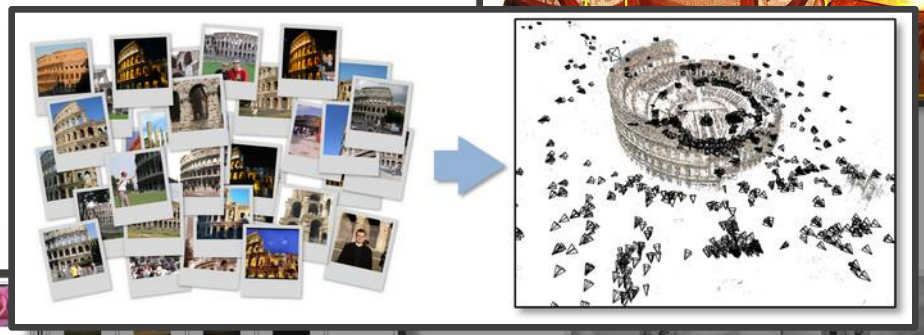
PPKE-ITK, 2016

Lecture 8.

Local Feature Descriptors

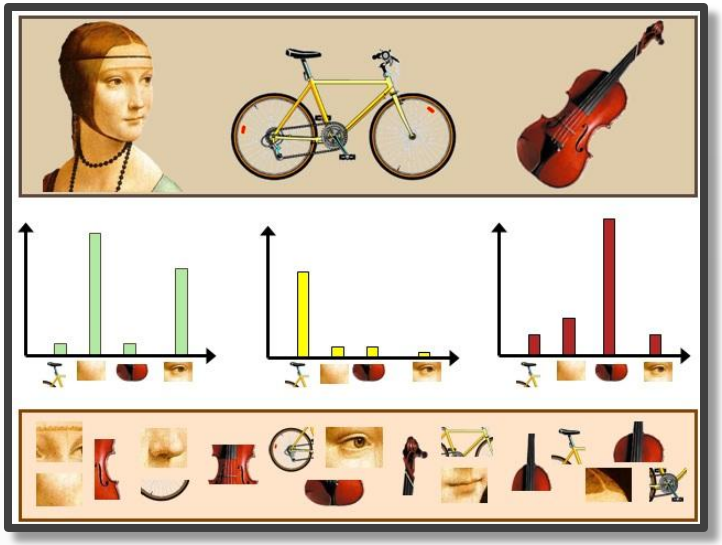
Local Feature Descriptors

- The detection and description of local features has an important role in many applications:
  - Object recognition/detection/tracking
  - Image and video retrieval
  - Image registration, motion estimation
  - Wide baseline matching
  - Texture classification
  - Structure from Motion
  - etc.



Local Feature Descriptors

- There are different types of use of the descriptors:
  - Description and matching of key points:
    1. Feature/Keypoint detection
    2. Local feature description around the key points
    3. Keypoint matching
  - Bag-of-Features (or bag-of-words)
    1. Feature detection
    2. Feature description
    3. Feature clustering
    4. Frequency histogram construction for image or image part description
  - Description of a specific area:
    1. Find the region of interest (ROI) (e.g. scanning through the image)
    2. Description of the ROI
    3. Classification/Clustering of the ROI descriptor



Local Feature Descriptors

- When we are talking about local feature descriptors we usually talking about one or both of the following two tasks:
  - Keypoint or feature detection
  - Feature extraction: generation of a descriptor for the feature point's local neighborhood.
- There are method that does both or only one of the tasks.

Feature point detectors	**Feature descriptors**
Hessian/Harris corner detector	SIFT
Laplacian of Gaussian	SURF
Difference of Gaussian (in SIFT)	HOG
SURF (uses Hessian Blob detector with integral image)	BRIEF
…	LBP
	…

SIFT: Scale-Invariant Image Transform

- Published by David Lowe in 1999.
- Advantages:
  - Invariant to translation, scaling, and rotation
  - Robust to illumination changes, noise, minor changes in viewpoint
  - Robust to local geometric distortion
  - Highly distinctive
  - SIFT based object detectors are robust to partial occlusion
- Steps of the Algorithm:
  1. Scale-space extrema detection
  2. Keypoint localization
  3. Orientation assignment
  4. Keypoint description

Lowe, „*Object recognition from local scale-invariant features*”, *In: The Proceedings of the Seventh IEEE International Conference on*, Vol. 2 (1999), pp. 1150-1157 vol.2.

SIFT: Scale-Invariant Image Transform

- I. Scale-space extrema detection:
  - Key point detection with ***Difference of Gaussians*** (DoG):
    - I(x, y) is the original image
    - G(x, y, kσ) is the Gaussian blur at scale kσ
    - The original image convolved with Gaussian kernel at different scales:

$$L(x, y, k \cdot \sigma) = G(x, y, k \cdot \sigma) * I(x, y)$$

    - The convolved images are grouped by octave (in an octave σ is doubled). The difference of consecutive convolved images is taken in an octave:

$$D(x, y, \sigma) = L(x, y, k_i \cdot \sigma) - L(x, y, k_j \cdot \sigma)$$

SIFT: Scale-Invariant Image Transform

- I. Scale-space extrema detection:

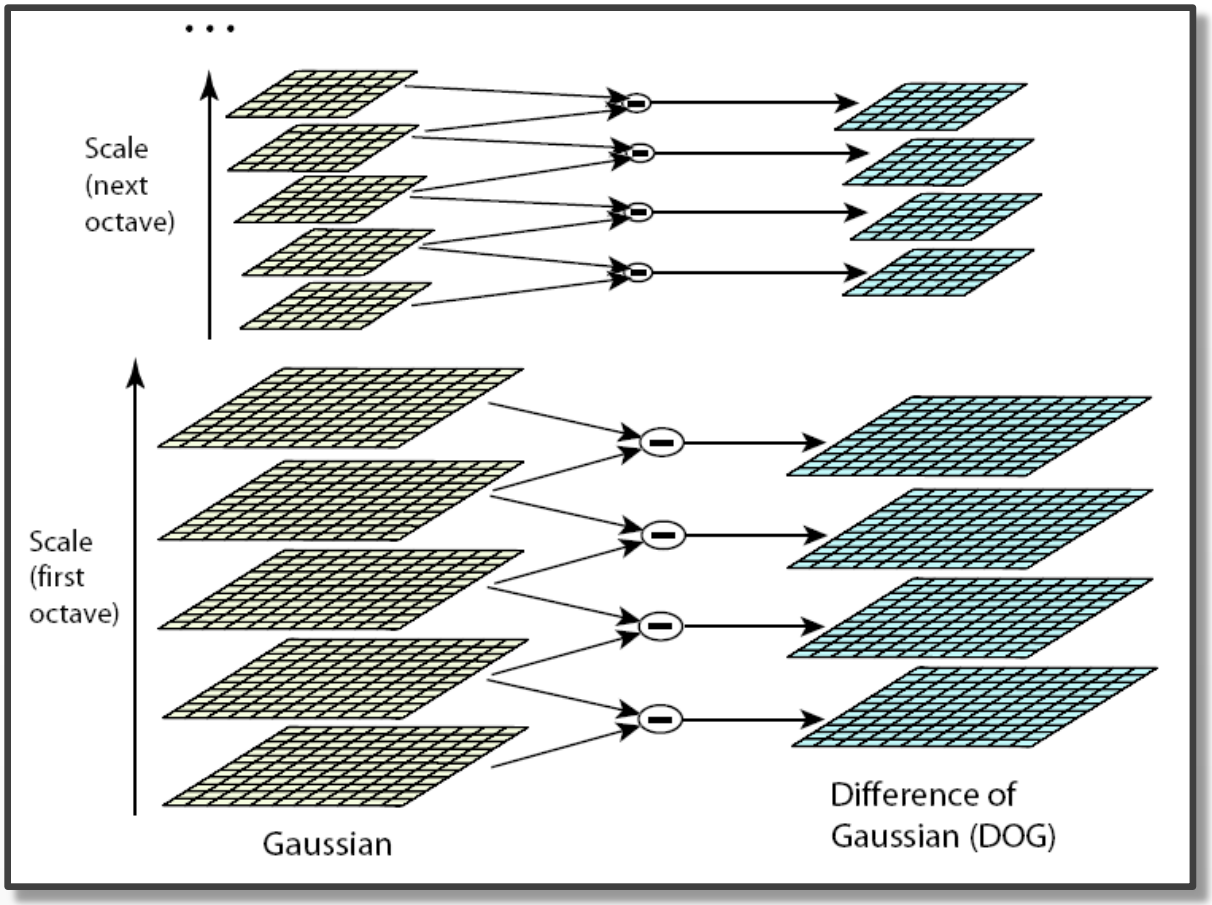


Image from: Ofir Pele

SIFT: Scale-Invariant Image Transform

- I. Scale-space extrema detection:
  - Choose all extrema within a 3x3x3 scale-space neighborhood. These extremas are the keypoints:

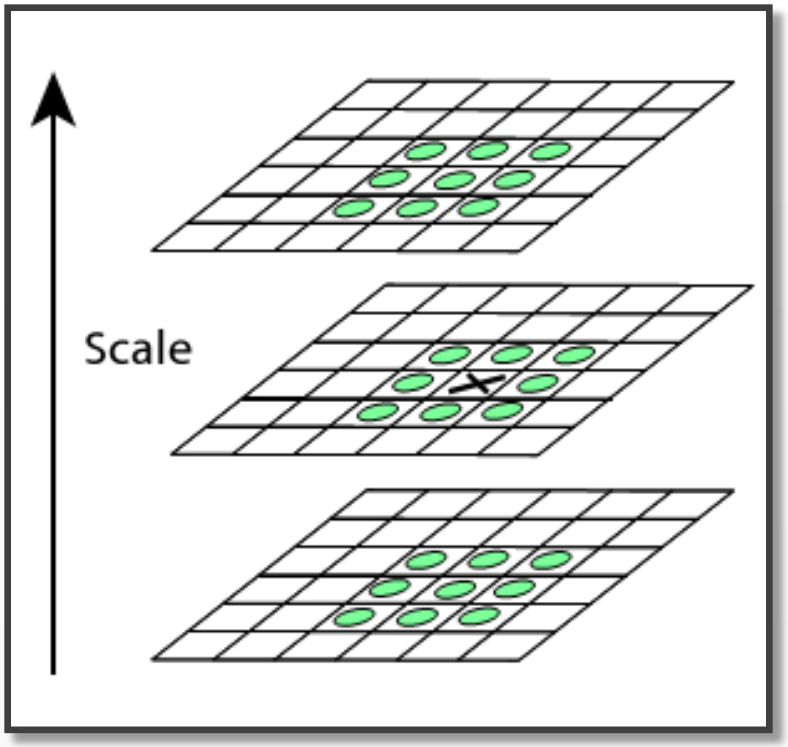
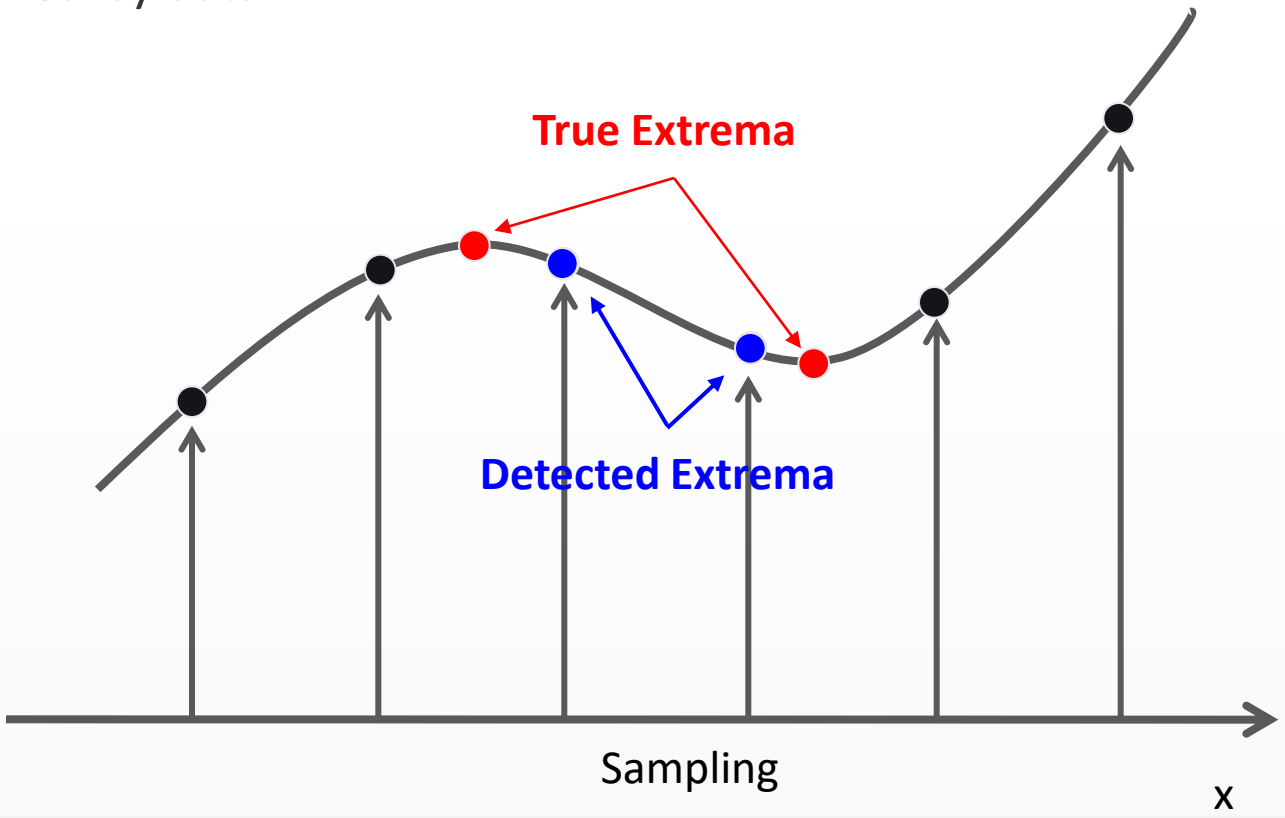


Image from: Ofir Pele

SIFT: Scale-Invariant Image Transform

- II. Keypoint localization:
  - Localization is done with sub pixel accuracy, based on the interpolation of nearby data:



SIFT: Scale-Invariant Image Transform

- II. Keypoint localization:
  - Rejection of weak candidates:
    - Low contrasted points
    - Poorly localized points along edges:
      - The DoG function will have strong responses along edges, but these points are not stable, since their location is poorly defined.
      - These points will be removed based on the principal curvature across and along the edge.

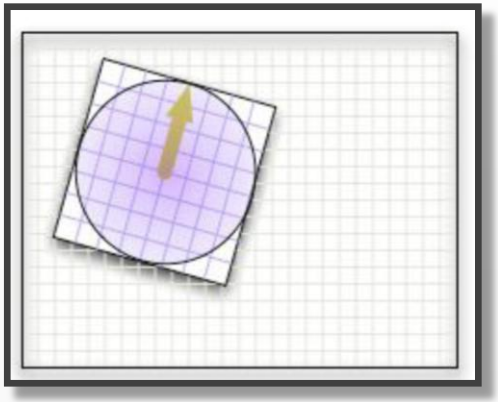
http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

SIFT: Scale-Invariant Image Transform

- III. Orientation Assignment:
  - Goal is to ensure rotation invariance:
    - Find the main orientation(s) and assign it to the key point and give the description of the keypoint relative to this orientation.
  - Steps:
    - ***Gaussian smoothed*** image is taken at the scale of the keypoint.
    - The ***edge magnitude and orientation*** is calculated ***for each point*** in the neighborhood.
    - A ***36 bin orientation histogram is composed***, where each bin represents a 10 degree interval, and each neighboring point's bin is determined based on its edge orientation and its weight based on the edge magnitude.
    - Also the points are ***weighted with a Gaussian window***, so the points farther away has less effect than the points closer to the keypoint.
    - The orientation of the keypoint will correspond to the peak of the histogram.

SIFT: Scale-Invariant Image Transform

- IV. Keypoint Descriptor:
  - For every keypoint (x,y,σ,θ):
    - Take a 16x16 point neighborhood around the keypoint and divide it into 4x4 gradient window.
    - Build the orientation histogram of the 4x4 samples in each window with 8 direction bins.
    - Gaussian weighting around center (size is based on σ)
    - 4x4x8 = 128 dimensional feature vector



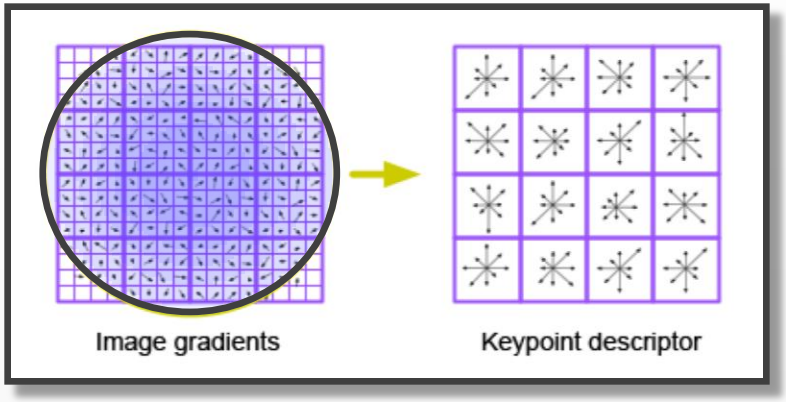


Image from: Ofir Pele

SIFT: Scale-Invariant Image Transform

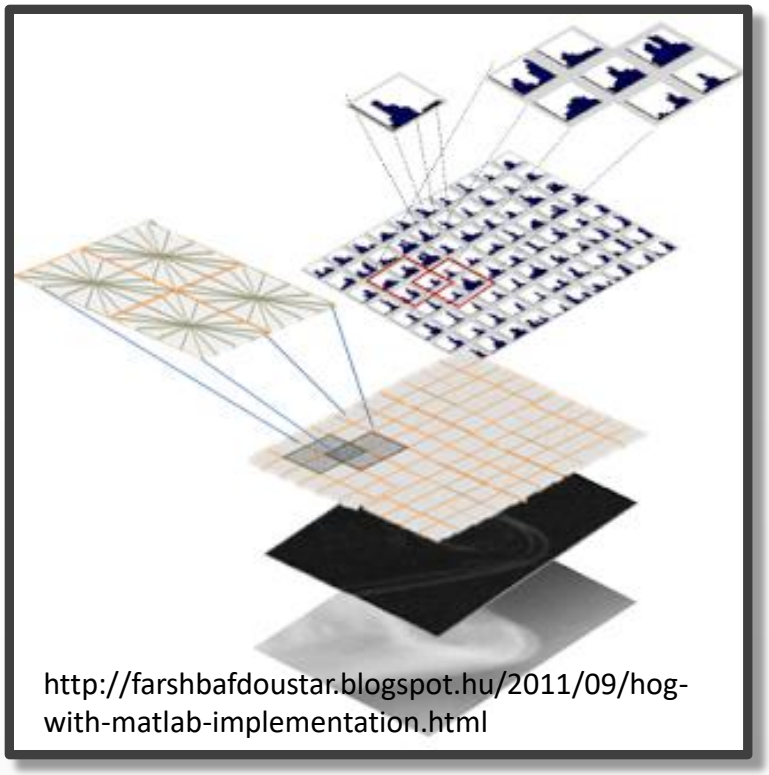
- SIFT inspired methods:
  - PCA-SIFT:
    - Reduce dimensionality, only keeps 20 dimension out of 128.
  - SURF:
    - Inspired by SIFT, but uses box filters (Haar like filters) with Integral Image implementation for fast calculation.
    - Has similar results as SIFT, but more sensitive to viewpoint and illumination changes.
  - …

Y. Ke and R. Sukthankar, "*PCA-SIFT: A More Distinctive Representation for Local Image Descriptors*," Proc. Conf. Computer Vision and Pattern Recognition, pp. 511-517, 2004.
H. Bay, T. Tuytelaars, L. Van Gool "*SURF: Speeded Up Robust Features*", Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1, pp 404--417, 2006.

HOG: Histogram of Oriented Gradients

- Originally developed for pedestrian detection by N. Dalal, B. Triggs in 2005.

- Steps of the Algorithm:
  1. Gradient Computation
  2. Orientation Binning
  3. Block Description
  4. Block Normalization
  5. Classification



http://farshbafdoustar.blogspot.hu/2011/09/hog-with-matlab-implementation.html

N. Dalal,B. Triggs, „*Histograms of Oriented Gradients for Human Detection*” In Proceedings of IEEE Conference *Computer Vision and Pattern Recognition*, San Diego, USA, pages 886-893, June 2005.

HOG: Histogram of Oriented Gradients

- I. Gradient Computation:
  - Many gradient detector was tested (Sobel, Prewitt, ..)
  - The simple [-1 0 1] and $[-1\ 0\ 1]^T$ gradient detectors gave the best result.

- II. Orientation Binning for a cell:
  - A ***cell*** is rectangular (or circular) shaped, 8x8 window.
  - Histogram of gradient orientations is calculated over the cell.
  - Each pixel votes based on its magnitude on the gradient image.
  - 9 bin histogram: 0-180°

N. Dalal,B. Triggs, „*Histograms of Oriented Gradients for Human Detection*" In Proceedings of IEEE Conference *Computer Vision and Pattern Recognition*, San Diego, USA, pages 886-893, June 2005.

HOG: Histogram of Oriented Gradients

- Rectangular HOG is similar to SIFT, with a few differences:
  - there is no dominant orientation alignment
  - single scale
  - spatial position is coded

◉ III. Block description:

- A block contains 2x2 cells
- Pixels in the block are weighted by a Gaussian window.

◉ IV. Block Normalization:

- The blocks are overlapping, every cell is used 4 times in 4 different blocks. Also there are different versions of the normalization:

L1-norm: $v \rightarrow \frac{v}{\|v\|_1 + \varepsilon}$

L1-sqrt: $v \rightarrow \sqrt{\frac{v}{\|v\|_1 + \varepsilon}}$

L2-norm: $v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}}$

L2-Hys: max value of v is limited to 0.2

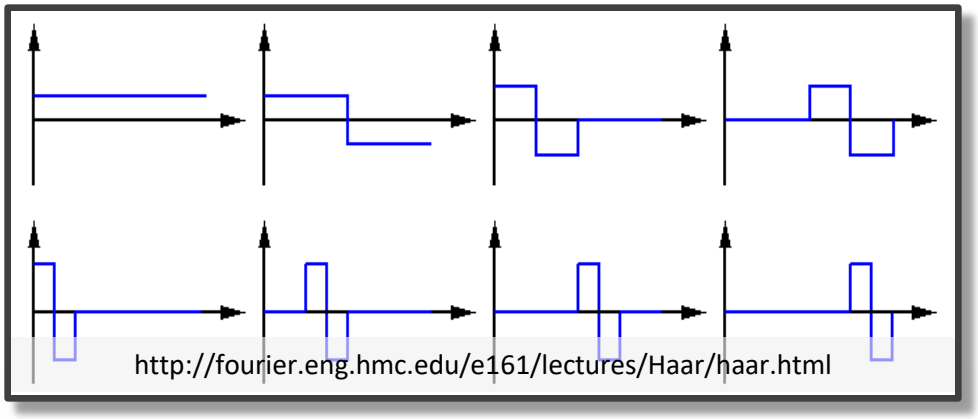
HOG: Histogram of Oriented Gradients

- V. Classification
  - Linear SVM
  - For pedestrian detection it was trained on 64x128 sized samples (positive and negative images).
- HOG inspired methods
  - A fast version of HOG with..
    - variable block size,
    - AdaBoost for feature selection,
    - cascade of rejectors and integral image representation.
    - 70x faster than the original
  - Motion Flow based HOG

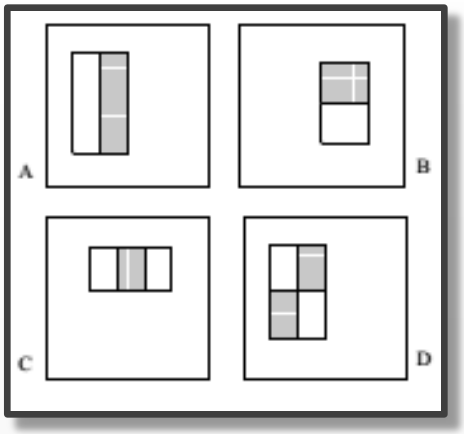
Zhu, Q., Yeh, M., Cheng, K., and Avidan, S., „***Fast Human Detection Using a Cascade of Histograms of Oriented Gradients***". In *Proceedings of the 2006 IEEE* Computer Society, CVPR, Washington, DC, 1491-1498.
N. Dalal, B. Triggs, C. Schmid, „***Human Detection Using Oriented Histograms of Flow and Appearance***", In Proceedings of the *European Conference on Computer Vision*, Graz, Austria, May 2006.

Haar-Like Features

- Named after the Haar wavelets:
  - a sequence of rescaled "square-shaped" functions
  - together the Haar wavelets form a basis.



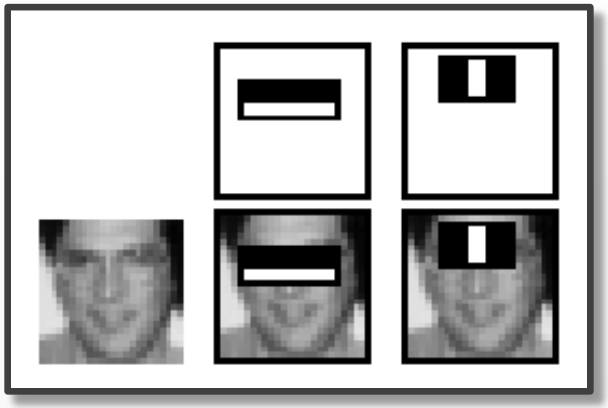
- Haar-like features were used in the first real-time face detector, published by Paul Viola and Michael Jones in 2001.
- For each face candidate window a lot of Haar-like feature is calculated and organized in a cascade form.



Paul Viola, Michael Jones, „Robust real-time object detection," *International Journal of Computer Vision*, 2001.

Haar-Like Features

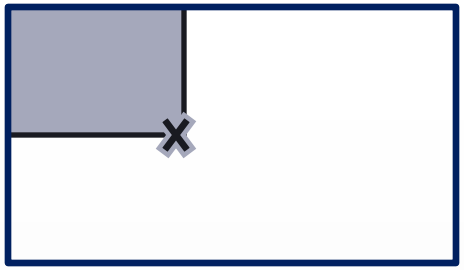
- Cascade classifier:
  - The goal is to be able to reject many obvious non-face samples quickly and concentrate the computational power on the more difficult samples.
  - By the concatenation of a lot of ***weak classifiers*** a highly effective classifier is built.
  - Each weak classifier can reject a sample, so the following weak classifiers don't have to evaluate it.
  - Each weak classifier is tuned to compensate the previous classifiers' errors.



Paul Viola, Michael Jones, „Robust real-time object detection," *International Journal of Computer Vision*, 2001.

Haar-Like Features

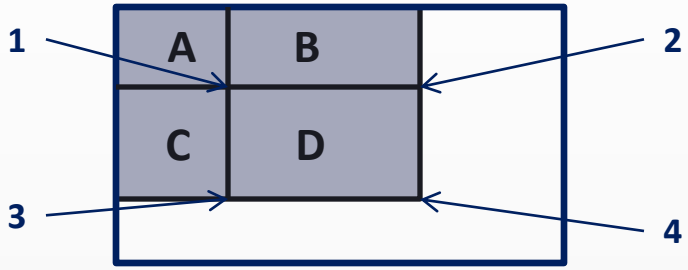
- Integral Image trick:
  - A way to calculate Haar-like features very quickly, in constant time, regardless of the size of the feature.
  - Definition of the Integral Image:



$$I_{\text{int}}(x, y) = \sum_{\substack{x' \le x \\ y' \le y}} I(x', y')$$

  - Using the integral image the sum of any rectangular shaped area can be calculated with 4 operation:

D = 4+1-(2+3)



Paul Viola, Michael Jones, „Robust real-time object detection," *International Journal of Computer Vision*, 2001.

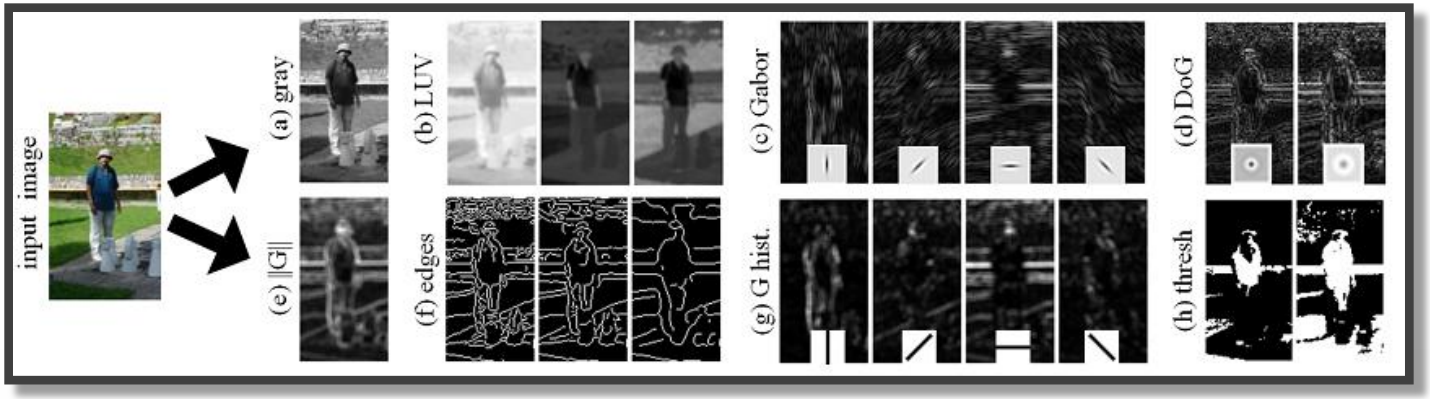
Haar-Like Features

Example of training faces for the Viola&Jones Face detector

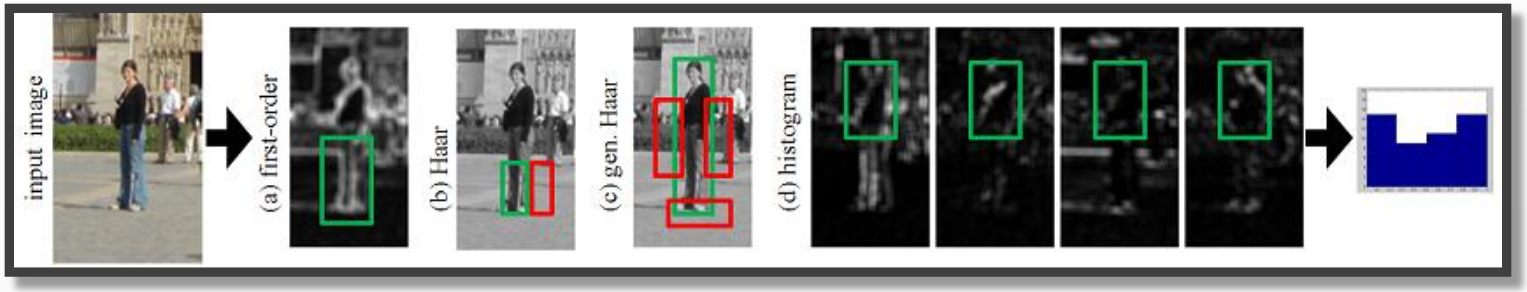
Paul Viola, Michael Jones, „Robust real-time object detection,” *International Journal of Computer Vision*, 2001.

Integral Channel Features

- Multiple registered image channels are computed:



- Features (local sums, histograms, and Haar features) are efficiently computed using integral images:



Piotr Dollar, Zhuowen Tu, Pietro Perona and Serge Belongie. Integral Channel Features. In A. Cavallaro, S. Prince and D. Alexander, editors, *Proceedings of the British Machine Conference,* pages 91.1-91.11. BMVA Press, September 2009. doi:10.5244/C.23.91.

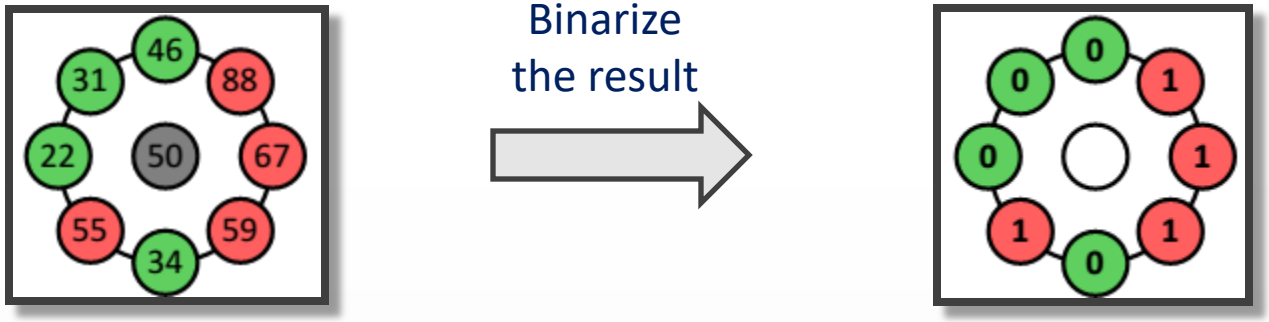
LBP: Local Binary Patterns

- Computationally effective texture descriptor.
  - Comparable to the state of the art,
  - while computable in O(n) time.
  - Robust to monotonic changes in the illumination, no image preprocessing or parameter tuning is required.
  - Produces a compact, 59 bin descriptor (SIFT has 128 bins), so it is faster to match.
  - Proved it's efficiency in many applications:
    - Face detection, recognition, facial expression analysis
    - Image retrieval, biometrics
    - Texture analysis, image segmentation
    - …

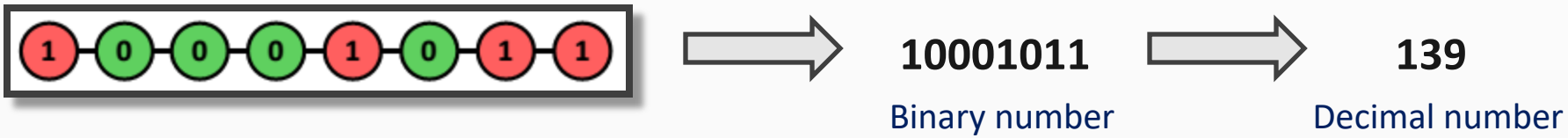
Ojala T, Pietikäinen M & Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971-987.

LBP: Local Binary Patterns

- Main steps of the algorithm:
  - Calculate the difference of a pixel and its neighbors in a fixed radius circular pattern:



  - Invariant to local contrast magnitude and global illumination changes.
  - Represent the result as a decimal number:



Marc Norvig's talk: Introduction to Local Binary Patterns
http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf

LBP: Local Binary Patterns

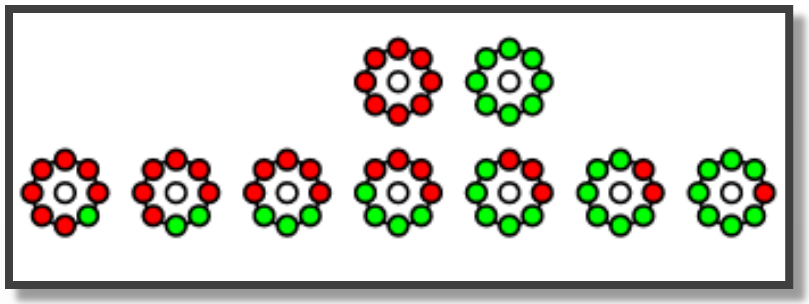
- Note:
  - the circular pattern does not fit very well on the squared sampling grid.
  - Nearest neighbor approximation is usually good enough.

- Main steps of the algorithm (continued):
  - Calculate the LBP for each point of the patch.
  - Build the histogram of the LBP values.
  - This histogram is the descriptor of the patch.
  - To match histograms the following measures is commonly used:
    - Histogram Intersection
    - Chi-Squared
    - Log-Likelihood

Marc Norvig's talk: Introduction to Local Binary Patterns
http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf

LBP: Local Binary Patterns

- Uniform LBP:
  - So far we have 256 dimension descriptor..
  - In general 90% of the LBPs has one or two continuous regions in it:
    - There are 2 patterns with one region
    - There are 7 patterns with 2 regions



  - For each of the 7 pattern with 2 regions there can be 8 different orientations.
  - We keep one joker bin for everything else: 2+7*8+1 = 59 bin descriptor
  - We reduced the #dimensions from 256 to 59 and as a bonus the resulted descriptor is more robust to noise.

Marc Norvig's talk: Introduction to Local Binary Patterns
http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf

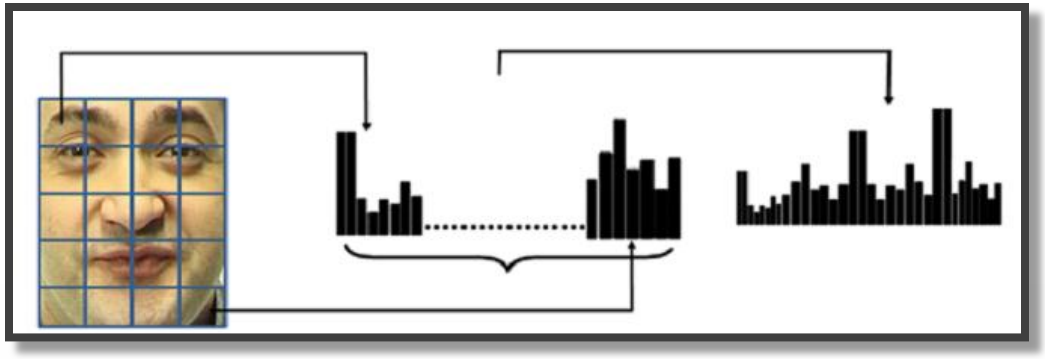
LBP: Local Binary Patterns

- Robustness against monotonic changes of illumination

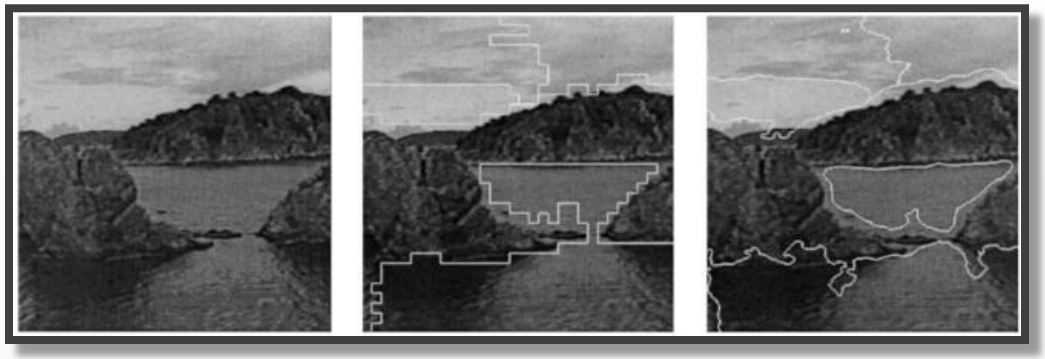
Marc Norvig's talk: Introduction to Local Binary Patterns
http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf

LBP: Local Binary Patterns

- Application of LBP for…
  - Face recognition:



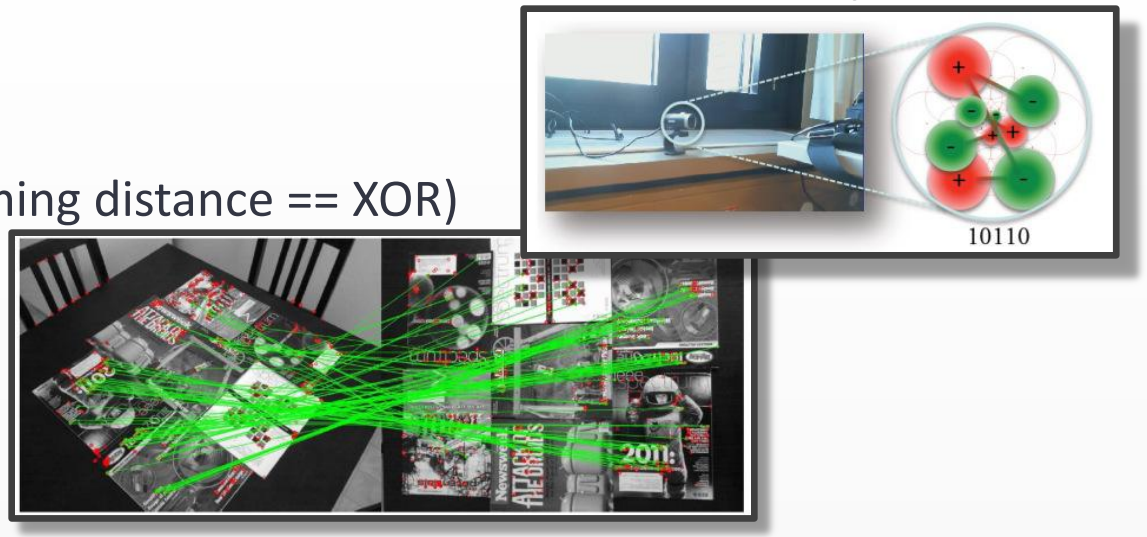
  - Image segmentation:



Marc Norvig's talk: Introduction to Local Binary Patterns
http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf

Binary Descriptors

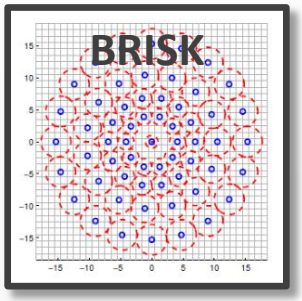
- SIFT, SURF and HOG:
  - are based on histograms of gradients, which is costly to compute,
  - the size of the descriptors can be problematic if we have many of them
  - also SIFT is patent protected.
- Binary descriptors use simple intensity value comparisons to create binary strings to encode the information of the patch:
  - Fast to compute,
  - Easy to store
  - Fast to match (Hamming distance == XOR)
- Binary Descriptors:
  - BRIEF, ORB
  - FREAK
  - BRISK

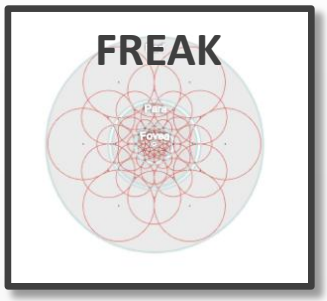


Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

Binary Descriptors

- In general, Binary descriptors are composed of three parts:
  - sampling pattern,
  - orientation compensation and
  - sampling pairs.
- I. Sampling pattern:
  - ***The use of binarized intensity value differences***: take a sample at point ***A*** and compare its value to a sample in an other point, ***B***. If ***A***'s intensity is higher add a 1 to the descriptor string, otherwise add 0.
  - The sampling pattern defines the way we take samples:





BRIEF and **ORB** use random pattern

Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

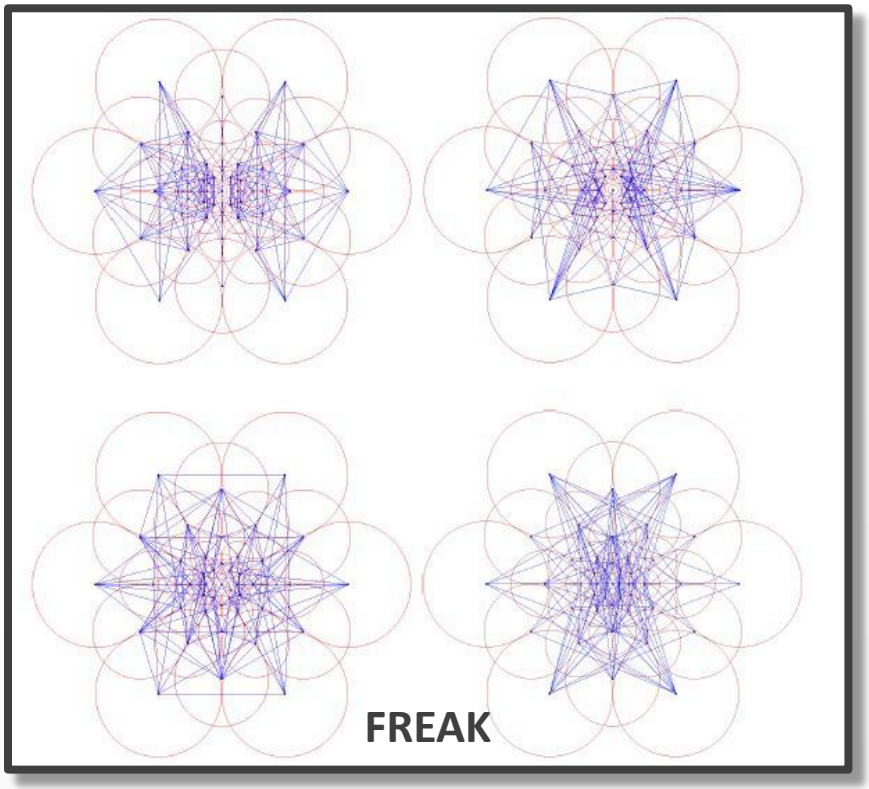
Binary Descriptors

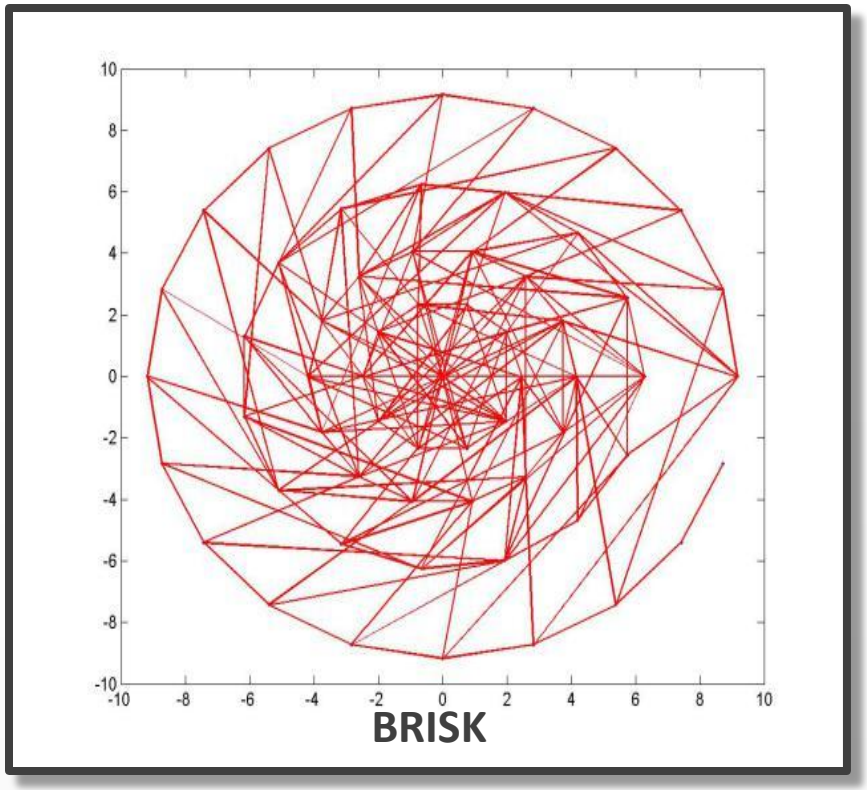
- Sampling pairs:
  - BRIEF uses random sampling pattern and selects random pairs from them.
  - BRISK uses only short distance pairs from the predefined pattern
  - FREAK and ORB ***learns*** the sampling pairs so that
    - their ***information content is maximal,*** the redundancy is minimal between the pairs,
    - and the variance of the pairs is high to make the feature more discriminative.
  - In case of FREAK the resulted pairs follow a coarse-to-fine structure:
    - The first pairs selected are comparing points in the outer ring
    - The last selected points make comparisons in the dense region
    - This resembles to the way the human vision operates.

Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

Binary Descriptors

- Sampling pairs:





Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

Binary Descriptors

- II. Orientation Compensation:
  - The binary pairs are sensitive to rotation.
  - In the orientation compensation phase the orientation angle of the patch is measured and the pairs are rotated by that angle to ensure that the description is rotation invariant.

  - Different descriptors have different methods for orientation compensation:
    - BRIEF: does not have orientation compensation
    - ORB: based on the moments of the patch
    - BRISK: comparing gradient of long pairs
    - FREAK: comparing gradient of preselected pairs

Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

Binary Descriptors

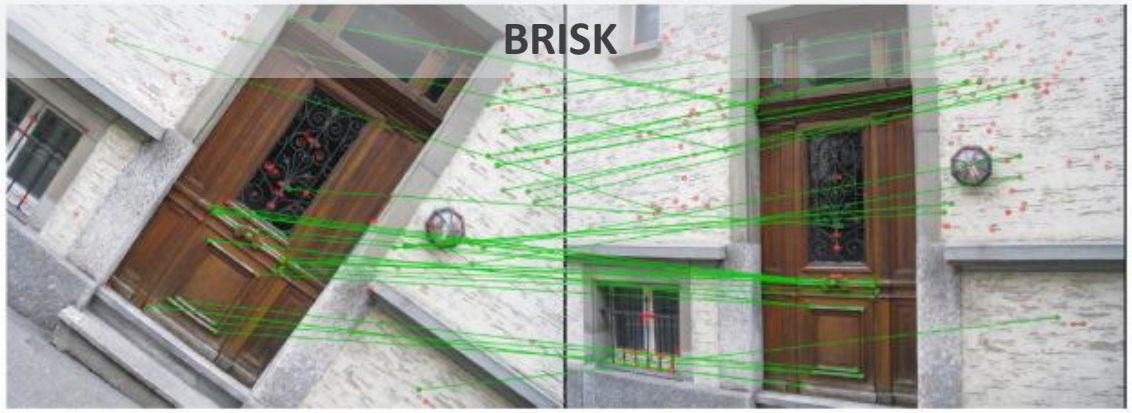
- III. Matching:
  1. Given two keypoint, first procedure the binary descriptor for both of them, using the same sampling pattern and the same sequence of pairs.
  2. Once we have two binary string, just count the number of bits where the strings are different.

  - FREAK uses a cascade approach (based on its special coarse to fine organization of sampling pairs):
    - It starts with matching only the first 128 bits, which can eliminate 90% of the candidates can be rejected (in average).
    - If the difference for the first 128 bits is lower than a threshold it checks the next 128 bits.

Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/

Binary Descriptors

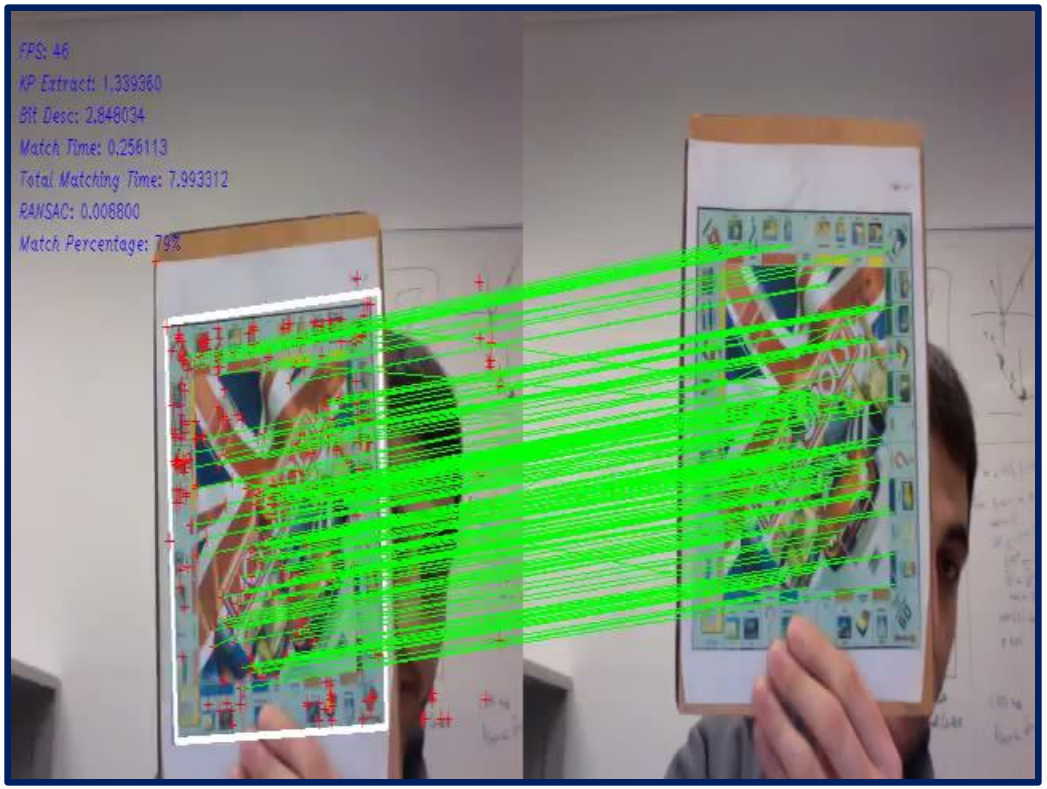
- Performance:
  - Which one is the best depends on the task:
    - BRIEF outperforms BRISK, ORB and FREAK in photometric changes – blur, illumination changes and JPEG compression.
    - When there are view-point changes, FREAK slightly outperforms ORB and BRISK, but performs worse than BRIEF.
    - When there are zoom + rotation changes, FREAK slightly outperforms BRIEF and BRISK, but performs worse than ORB.



Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/11/08/a-tutorial-on-binary-descriptors-part-4-the-brisk-descriptor/

Feature Matching

- BRIEF (Binary Robust Independent Elementary Features)



M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, ”Computing a Local Binary Descriptor Very Fast,” IEEE Transactions on Pattern Analysis and Machine Intelligence 2012

Feature Learning

- All of these features are hand crafted!
- To come up with a new feature is not trivial:
  - Requires domain knowledge,
  - a lot of experimantation,
  - for different tasks, probably slightly different features would be optimal,
  - etc.
- Feature learning:
  - Would be nice to do the feature engineering automatically!
  - Learn the ideal representation from the data.
  - How could we do that?
    - With Convolutional Neural Networks!

Main Sources and Further Readings

- Introduction to descriptors:
  http://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/
- C. Harris and M. Stephens (1988). "A combined corner and edge detector". "Proceedings of the 4th Alvey Vision Conference". pp. 147–151.
- Lowe, „*Object recognition from local scale-invariant features*”, *In: The Proceedings of the Seventh IEEE International Conference on*, Vol. 2 (1999), pp. 1150-1157 vol.2.
- Y. Ke and R. Sukthankar, “*PCA-SIFT: A More Distinctive Representation for Local Image Descriptors*,” Proc. Conf. Computer Vision and Pattern Recognition, pp. 511-517, 2004.
- H. Bay, T. Tuytelaars, L. Van Gool "*SURF: Speeded Up Robust Features*", Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1, pp 404--417, 2006.
- N. Dalal,B. Triggs, „*Histograms of Oriented Gradients for Human Detection*” In Proceedings of IEEE Conference *Computer Vision and Pattern Recognition*, San Diego, USA, pages 886-893, June 2005.
- Zhu, Q., Yeh, M., Cheng, K., and Avidan, S., „***Fast Human Detection Using a Cascade of Histograms of Oriented Gradients***”. In *Proceedings of the 2006 IEEE* Computer Society, CVPR, Washington, DC, 1491-1498.
- N. Dalal, B. Triggs, C. Schmid, „***Human Detection Using Oriented Histograms of Flow and Appearance***”, In Proceedings of the *European Conference on Computer Vision*, Graz, Austria, May 2006.

Main Sources and Further Readings

- Gil's Computer vision blog: http://gilscvblog.wordpress.com/2013/08/26/tutorial-on-binary-descriptors-part-1/
- Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, "BRIEF: Binary Robust Independent Elementary Features", 11th European Conference on Computer Vision (ECCV), Heraklion, Crete. LNCS Springer, September 2010.
- Alahi, Alexandre, Raphael Ortiz, and Pierre Vandergheynst. "Freak: Fast retina keypoint." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.
- Leutenegger, Stefan, Margarita Chli, and Roland Y. Siegwart. "BRISK: Binary robust invariant scalable keypoints." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.
- Rublee, Ethan, et al. "ORB: an efficient alternative to SIFT or SURF." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.
- Ojala T, Pietikäinen M & Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971-987.
- Marc Norvig's talk: Introduction to Local Binary Pattern: http://files.meetup.com/4379272/BIPCVG_LocalBinaryPatterns_2013.01.16.pdf