# Pattern recognition using correlation

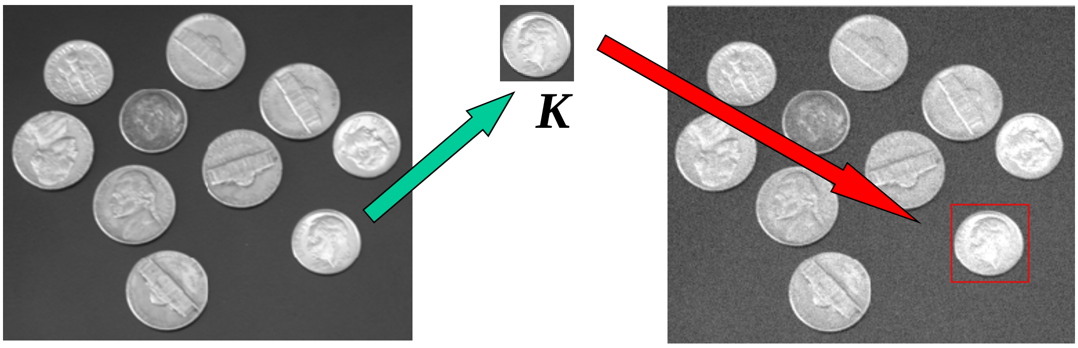
Correlation with a reference image

– Correlation coefficient

$$R = \frac{\sum_{i,j}(J(i,j) - \bar{J})(K(i,j) - \bar{K})}{\sqrt{\left(\sum_{i,j}(J(i,j) - \bar{J})^2\right)\left(\sum_{i,j}(K(i,j) - \bar{K})^2\right)}}$$

– Correlation is often used for simple comparison of the image or parts of the image with the learned pattern. The value of the correlation determines the degree of matching the image region to the reference pattern.

– This method is very simple but it is sensitive to noise, orientation changes of the objects and time-consuming for large patterns



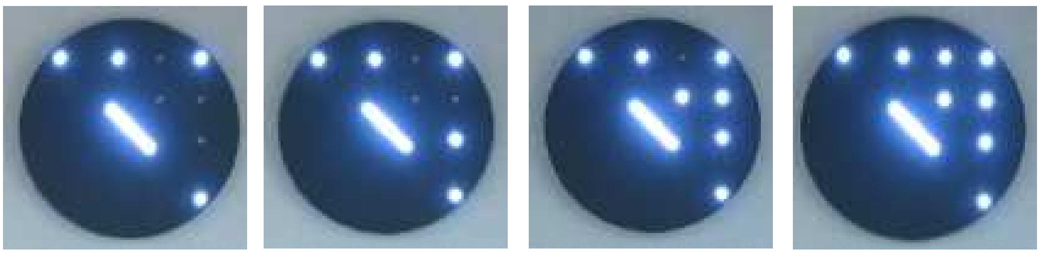

*R*=0,993



# Objects recognition using artificial markers

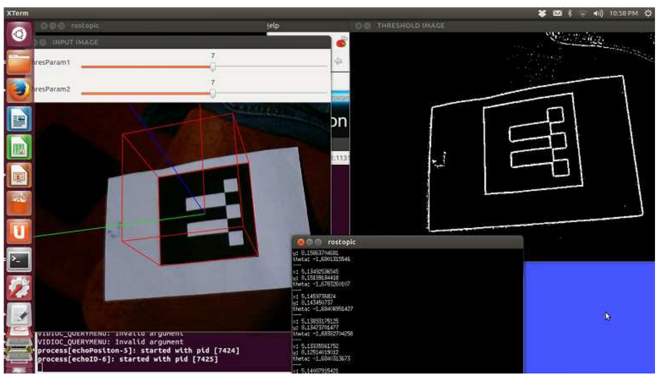
Artificial markers for objects (robots) localization

- LED markers for accurate localization on 2D scene  (up to 32)



- Markers from AruCo library (up to 1024)

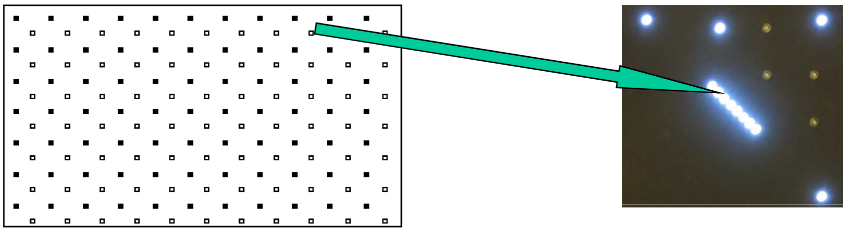
Id = 1 Id = 100 Id = 299



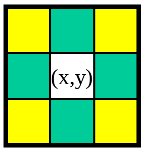
# Objects recognition using artificial markers

## LED marker recognition and localization

- Special image binarization for bright pixels determining (with simultaneous logical filtering)



- Light blobs determining in sense of 4-neighborhood connectivity (labeling algorithm)




- Classification of the blobs based on size limits according to the expected size of 8-LEDs blob
- Calculation of the rough position of the marker’s center and its orientation using statistical normal and central moments



# Objects recognition using artificial markers

## LED marker recognition and localization (continued)

- Finding and determining the position of the three external LEDs of the marker (the triangle LEDs vertices)
- The geometrical relationships analysis of triangle vertices position in order to determine the front of the marker
- Calculation of accurate position and final orientation of the marker in the image space

Robot position

$$\boldsymbol{p} = \frac{1}{2}(\boldsymbol{p}_1 + \boldsymbol{p}_2)$$

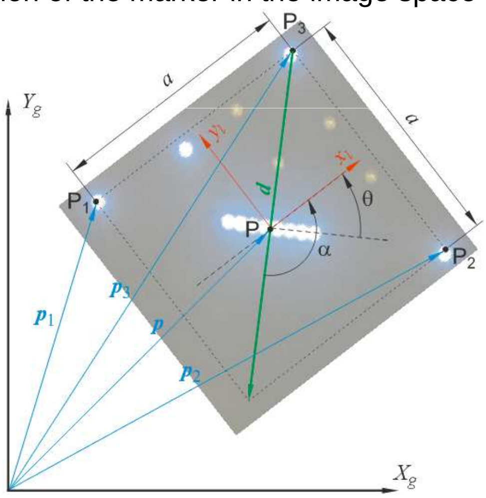
Orientation

$$\theta = \mathrm{atan2}(d_y, d_x) + \alpha$$

$d_y, d_x$ - coordinates of vector $\boldsymbol{d}$

$$\boldsymbol{d} = \boldsymbol{p}_1 + \boldsymbol{p}_2 - 2\boldsymbol{p}_3$$

$$\alpha = 135^\circ$$




- Determination of number of the marker based on five additional LEDs
- Recalculation from image space to task coordinate space using camera calibration parameters

# Objects recognition using local features

Representation of the model of object using local features

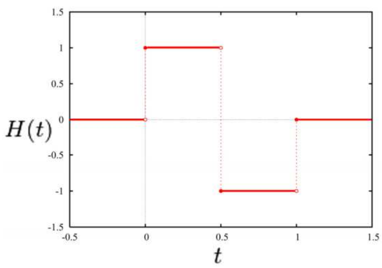
- Key properties of local features for good objects description:
  - Must be highly distinctive, good features should allow for correct object identification with low probability of mismatch;
  - Should be easy to extract (fast extract);
  - Should be tolerant to image noises and changes in illumination;
  - Invariant to changes in object position and orientation (rotation);
  - Invariant to scaling of the object;
  - Invariant to minor changes in viewing direction and angle;
  - Should be easy to match against a large database of local features defining objects.



# Objects recognition using local features

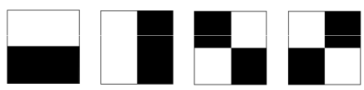
Haar features (wavelets) i Haar-like features

- One dimensional Haar function

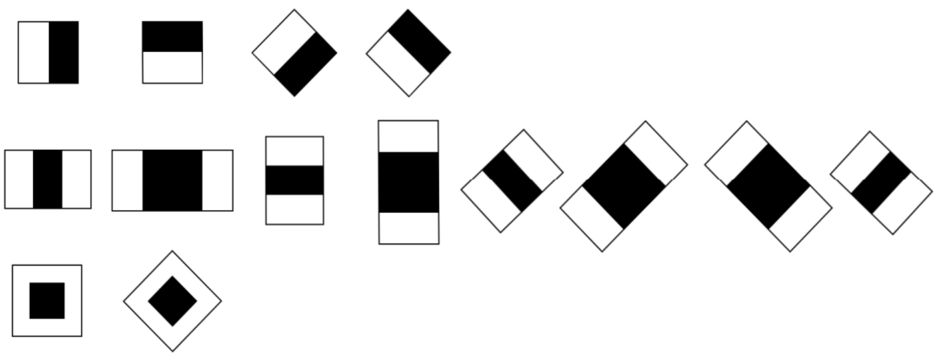



$$H(t) = \begin{cases} 0 & t < 0 \\ 1 & 0 \leq t < 0{,}5 \\ -1 & 0{,}5 \leq t < 1 \\ 0 & t \geq 1 \end{cases}$$

- Two dimensional Haar functions

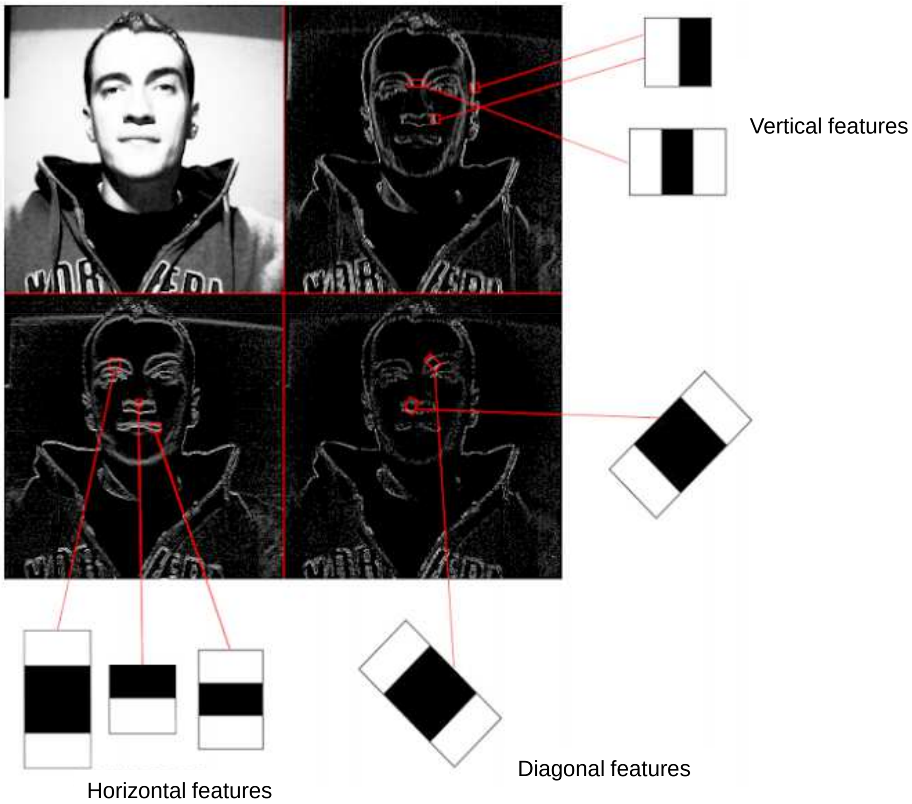


- The set of functions for determining Haar-like features



# Objects recognition using local features

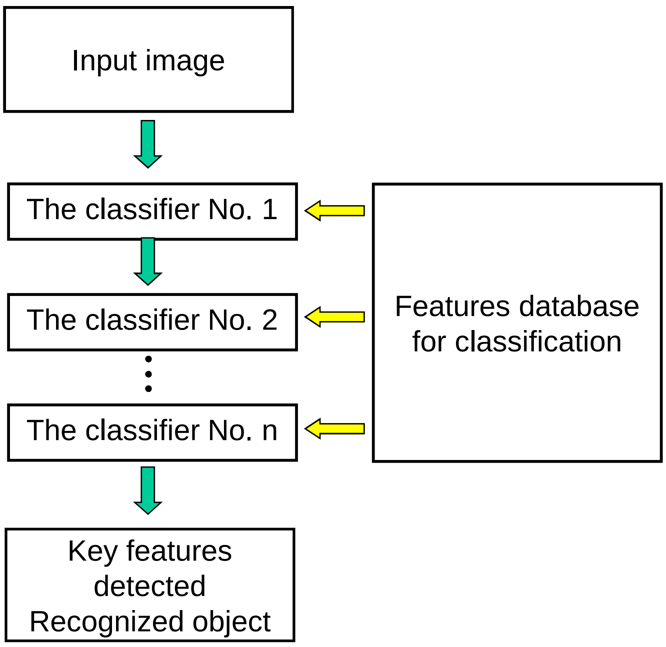
Image processing of selected Haar-like features






# Objects recognition using local features

Cascade classification (cascading detection of key features in the image)



# SIFT algorithm

SIFT – Scale Invariant Feature Transform

David G. Lowe: „*Distinctive Image Features from Scale Invariant Features*", International Journal of Computer Vision, Vol. 60, No 2, 2004, pp. 91-110

- SIFT – transform pixels features from spatial image domain to corresponding local features coordinates domain.
- SIFT – allows for local features detection and then extracting local feature descriptors that are invariant to changes in position, rotation, scaling, illumination, image noise, and small changes in viewpoint.
- The use of SIFT algorithm in image recognition requires the extraction of local features from reference images for database preparing.
- SIFT method allows for correct identification of objects in the clutter scene with a high probability, even in the case of partial objects obstruction!



# SIFT algorithm

Main detection stages for SIFT local features (cascade filtering approach):

1. Scale-space extrema detection,
2. Keypoints localization,
3. Orientation assignment for keypoints,
4. Generation of keypoint descriptors.

# SIFT algorithm

## 1. Scale-space extrema detection

- Points for SIFT local features, which are invariant to scale, correspond to local extrema of difference-of-Gaussian filters at different scales.

  For given Gaussian blurred image

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

  where

  $$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

  is a variable scale Gaussian.

  Difference-of-Gaussian (DoG) filter for two scales separated by coefficient $k$

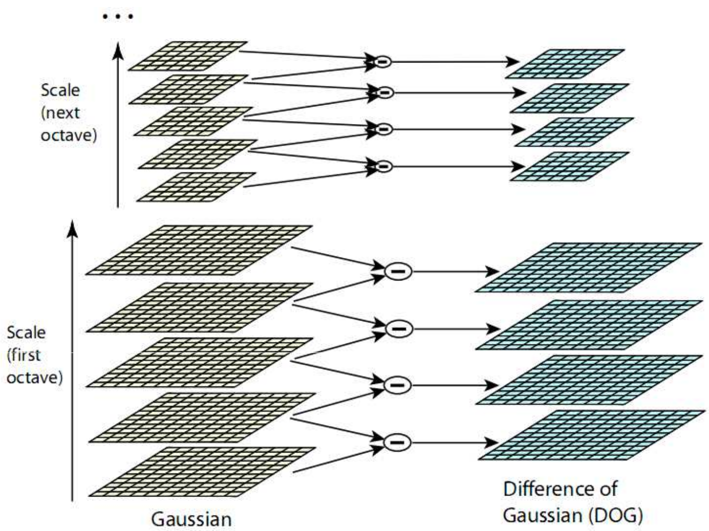
  $$G(x, y, k\sigma) - G(x, y, \sigma)$$

  The result of image convolution with DoG filter is given by

  $$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y)$$
  $$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$



# SIFT algorithm



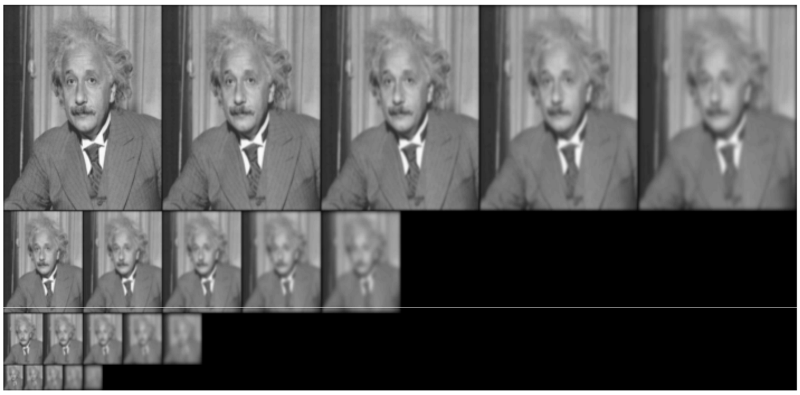

*[Lowe]*

In The SIFT algorithm, the input image is filtered using Gaussian filter at different scales separated by a constant factor $k$ and then the DoG images are generated.

The procedure is repeated by resampling the Gaussian image taking every second pixel in each row and column.

# SIFT algorithm

Example of Gaussian images and DoG images at different scales

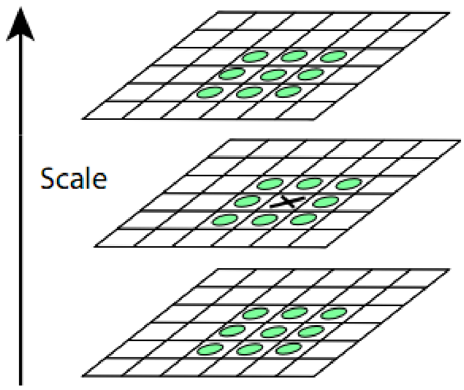




# SIFT algorithm

- Local extrema detection

  Detection of local maxima and minima is done by comparing the point with its neighbors in the surrounding 3x3x3 (comparing to 26 pixels). The point is taken as a candidate for the keypoint feature if it is larger or smaller than all neighboring points.




[Lowe]

# SIFT algorithm

## 2. Keypoint localization

- The keypoint localization is done by interpolating the candidate points found in the first stage by using the 3D quadratic function (Taylor expansion up to the quadratic terms)

$$D(\boldsymbol{x}) = D + \frac{\partial D^T}{\partial \boldsymbol{x}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T \frac{\partial^2 D}{\partial \boldsymbol{x}^2}\boldsymbol{x}, \ \boldsymbol{x} = (x, y, \sigma)^T$$

Accurate location of the extremum is calculated base on derivatives of function $D(\boldsymbol{x})$

$$\hat{\boldsymbol{x}} = -\frac{\partial^2 D^{-1}}{\partial \boldsymbol{x}^2}\frac{\partial D}{\partial \boldsymbol{x}}$$

- The function value at the extremum is given by

$$D(\hat{\boldsymbol{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \boldsymbol{x}}\hat{\boldsymbol{x}}$$

and it is used to reject unstable points due to the low contrast

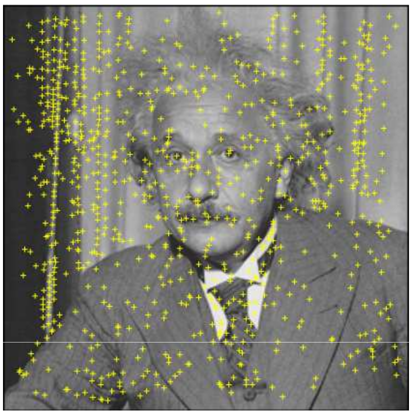
- Extrema points which are response along the edges in the image are rejected on the basis of the condition

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

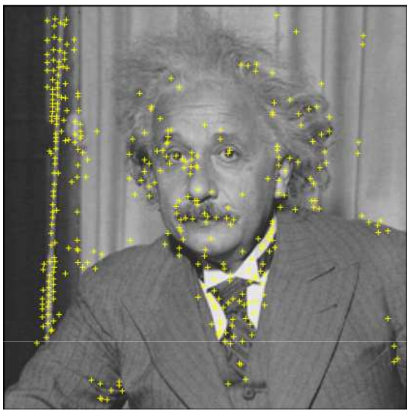
where Hessian $\boldsymbol{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$



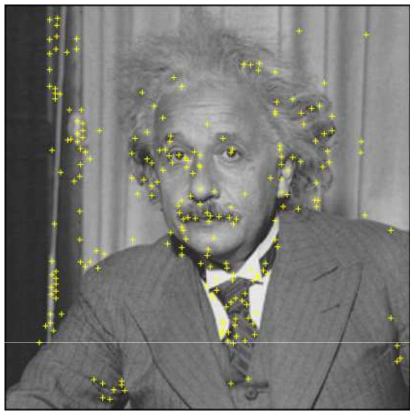
# SIFT algorithm



Candidates for the keypoints - the extremes obtained for the DoG filters at different scales



The keypoints remaining after rejection the candidates with low contrast



The final set of keypoints remaining after rejection the extremes in response along the edges

# SIFT algorithm

## 3. Keypoints orientation

- Orientation of keypoint is determined based on one of the Gaussian images, whose scale corresponds to the scale of the keypoint. For each keypoint of the image the gradient magnitude is calculated and the orientation given by

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

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

  Next an orientation histogram is formed based on gradient orientation of sample points within a region around the keypoint. Each sample is additionally weighted by its gradient magnitude and by a Gaussian-weighted circular window with $1{,}5\sigma$ according to the scale of the keypoint. The highest peak in the histogram and any other local peak that is within 80% of the highest peak is used to create keypoint with that orientation.

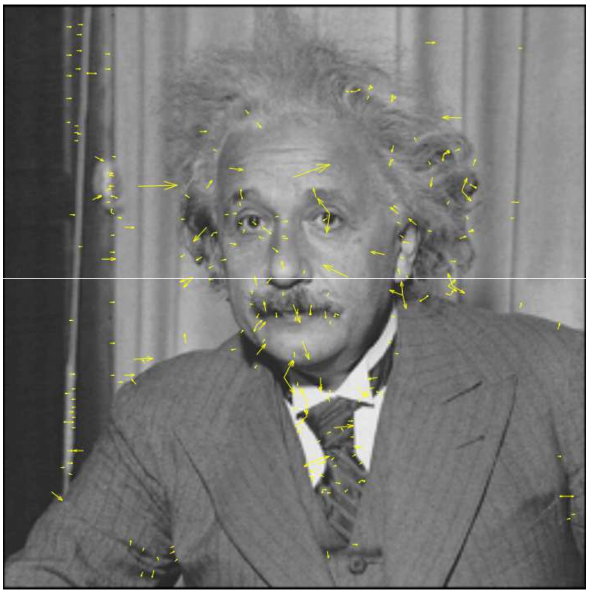
  The SIFT method assumes that the histogram has 36 bins, and the accurate position of the peak is interpolated using a parabola and taking closest values to the peak in the histogram.

- All the features of the keypoint are measured with respect to the orientation so the description is invariant to the rotation on the image.



# SIFT algorithm

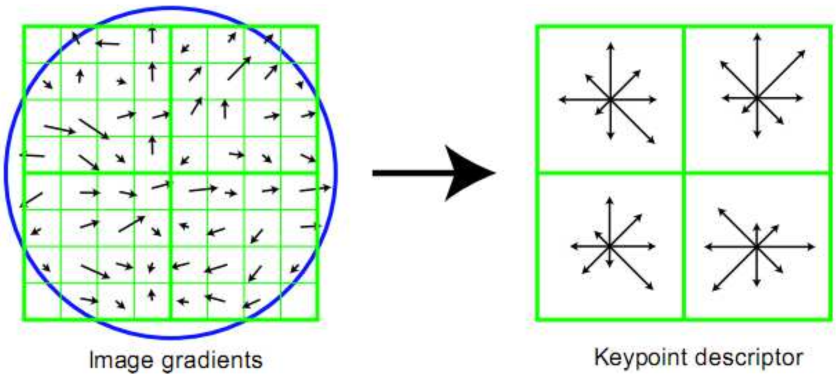
Keypoints location in the image, taking into account the scale and orientation



# SIFT algorithm

## 4. Generation of descriptors of local features for keypoints

- A keypoint descriptor is created by first computing gradient magnitude and orientation at each sample point in a region around the keypoint location (16x16 sample region). These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions. In each subregion for 8 orientation the resultant gradient magnitude is calculated based on gradient magnitude of each sample point.
- Local feature keypoint descriptor is constitute by 4x4x8=128 element feature vector. Then the feature vector is normalized to enhance invariance to changes in illumination.




*[Lowe]*

*Example for sample region 8x8 and subregion 2x2*



# SIFT algorithm – object recognition

## SIFT keypoint feature matching

- For the given keypoint the best candidate for matching is nearest neighbor keypoint in the database from training image with closest vector feature. The nearest neighbor keypoint can be a vector with minimum Euclidean distance calculated to feature vector of the given keypoint.
- How to effectively compare and search for closest feature vectors in space 128D?
- The SIFT algorithm use for database searching an approximate algorithm called Best-Bin-First. This is approximate in the sense that it returns the closest neighbor with high probability.

  For matching purpose only the nearest neighbor and the second nearest neighbor is considered.

# SIFT algorithm – object recognition

## Keypoint database

- Object recognition requires the preparation of a database containing keypoints of the objects from reference images.

## Recognition using SIFT keypoint features

- Compute SIFT keypoint features vector on the input image
- Match these vector features to the trained database. Each matched keypoint is specified by four parameters: 2D location (x,y), scale and orientation
- Hough transform is used to increase recognition robustness to identify clusters of matches that vote for the same object pose. Locations in the Hough accumulator matrix that accumulate at least 3 votes are selected as object candidate and describe keypoint location, scale and orientation
- The last step is geometrical verification procedure in which a least-squares method is used. As a result the best affine projection parameters relating the training image and tested image are calculated.



# Other algorithms based on local features

- SURF: Speeded-Up Robust Features
  (H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In European Conference on Computer Vision, ECCV 2006 )
- FAST: Features from Accelerated Segment Test
  (E. Rosten and T. Drummond. Machine learning for highspeed corner detection. In European Conference on Computer Vision, ECCV 2006)
- BRIEF: Binary robust independent elementary features
  (M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In European Conference on Computer Vision, ECCV 2010)
- BRISK: Binary Robust Invariant Scalable Keypoints
  (S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In IEEE International Conference on Computer Vision, ICCV 2011)
- ORB: Oriented FAST and Rotated BRIEF
  (E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In IEEE International Conference on Computer Vision, ICCV 2011)
- FREAK: Fast Retina Keypoint
  (A. Alahi, R. Ortiz, P. Vandergheynst. FREAK: Fast Retina Keypoint. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012)