SURF Object Recognition Algorithm

Outline

- Overview of SURF
- Implementation
- Summary
- References
- Concepts

Overview of SURF

- SURF – Speeded Up Robust Features
- Fast, robust object detection algorithm
- Works by identifying sets of feature points for an object and matching these between images
- Differences with SURF
  - Detectors and descriptors modified to improve speed of execution
  - Invariance to affine transforms is assumed to occur as a side effect of the primary efforts
  - In general, lessons learned from other methods are taken into account

Implementation

- Procedure:
  1) Interest point detection
  2) Represent each point by a distinctive feature vector
  3) Match descriptor vectors between images

Implementation

- Interest Point Detection

  Hessian matrix:

$$\mathcal{H}(\mathbf{x},\,\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\,\sigma) & L_{xy}(\mathbf{x},\,\sigma) \\ L_{xy}(\mathbf{x},\,\sigma) & L_{yy}(\mathbf{x},\,\sigma) \end{bmatrix}$$

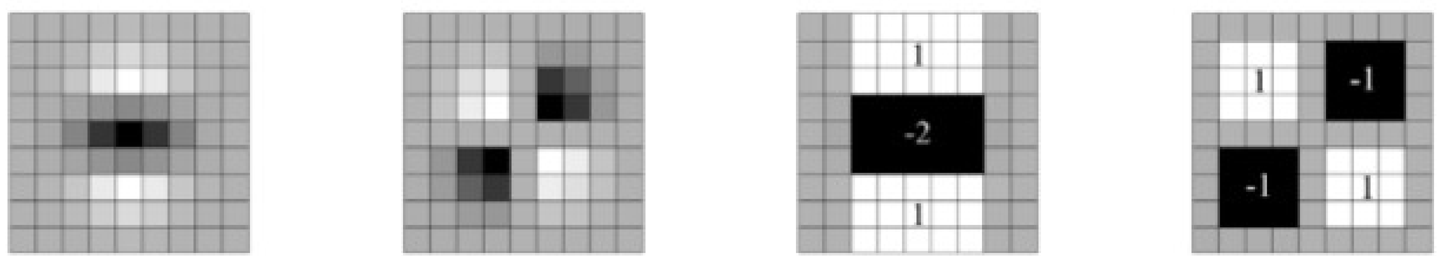
  Based on Gaussian second derivatives:

$$L_{xx}(\mathbf{x},\,\sigma) \qquad \frac{\partial^2}{\partial x^2} g(\sigma)$$

Implementation

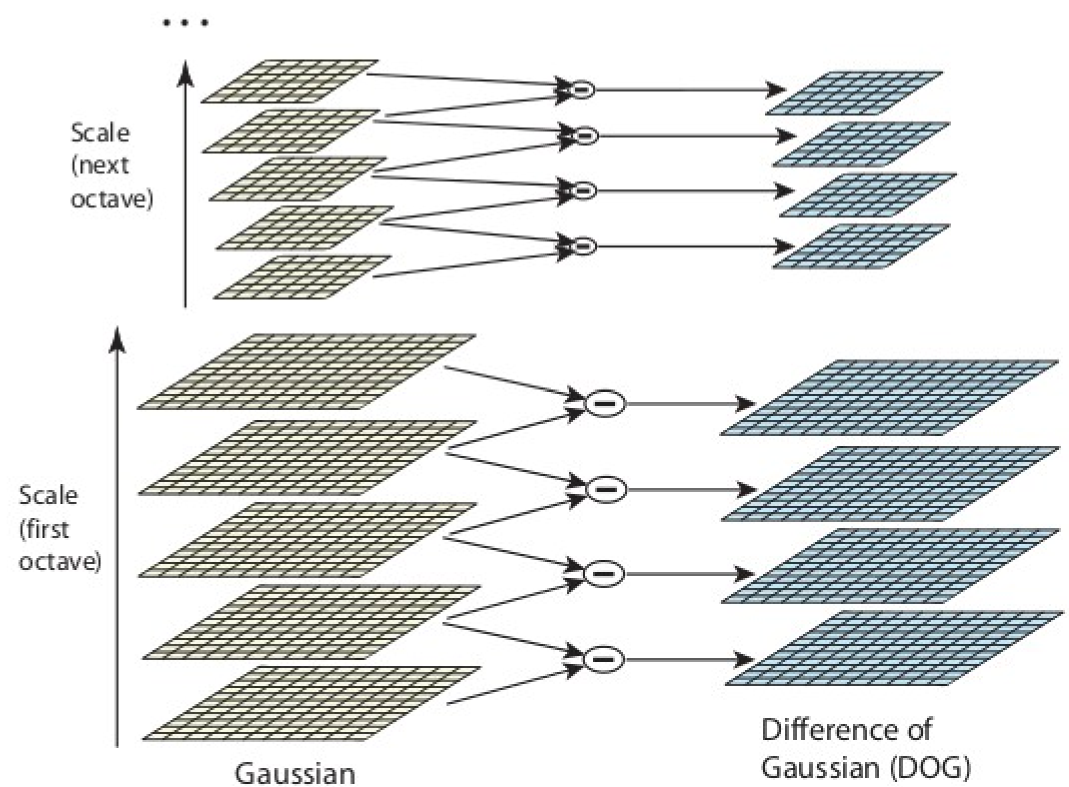
- Interest Point Detection

  Approximation to the Gaussian second order derivatives:



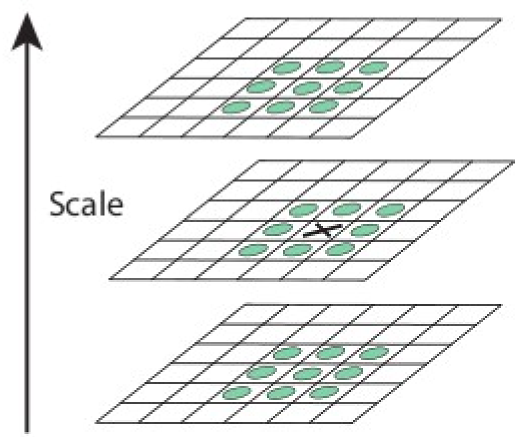
Implementation

- Interest Point Detection
  - Scale space invariance is achieved in a manner similar to SIFT
  - Scale is increased by convolving Gaussian derivatives with different window sizes, to the original image
  - Next octave is reached by downsampling the last image



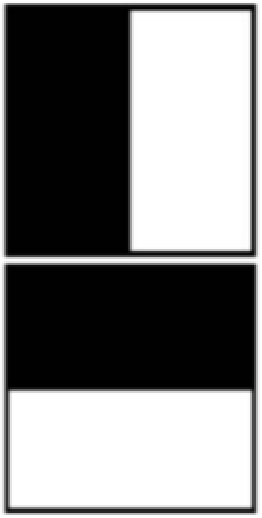
Implementation

- Interest Point Detection
  - Extrema of the determinant of the Hessian are detected
  - Interpolation used in both scale and image space
  - Result is points of “interest”



Implementation

- Descriptor (feature vector) Assignment
  - Orientation Assignment
    - Compute Haar-wavelet responses in a circular region about the interest point
    - Scale = s = sigma
    - Radius 6*s
    - Side length of Haar wavelet = 4*s
    - Dominant orientation estimated by summing responses in sliding orientation window



Implementation

- Descriptor Assignment
  - Descriptor Components
    - Square region aligned with dominant direction
    - Side length = 20*s
    - Break square into 4x4 sub-regions

Implementation

- Descriptor Assignment
  - Descriptor Components
    - Compute features at 5x5 spaced points in sub-regions
      - Each point Gaussian weighted by 3.3*s
      - Haar wavelet respones dx, dy for each point summed for each sub-region
      - Absolute values also summed
      - Invariant to illumination, which acts as offset

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right)$$

Implementation

- Descriptor Assignment
  - Descriptor Components
    - Compute features at 5x5 spaced points in sub-regions
      - This vector is 4-dimensional
      - Each sub-region region contains one vector
      - The point's Laplacian is also stored: This indicates “bright” or “dark” region

Implementation

- Match descriptor vectors between images
  - Nearest-neighbor algorithm
    - Euclidean distance between feature vectors in feature space
  - Similarity-threshold based algorithm
    - Some approximation to the distance found without checking all point sets

Summary

- SURF is an object detection algorithm designed to be robust to scale and rotation
- More simplistic implementation than SIFT
- Shown experimentally to be invariant to small-magnitude affine transforms
- 3 times faster than SIFT
- 5 times faster than Hessian-Laplace

Summary

- Performance summary

detector	threshold	nb of points	comp. time (msec)
Fast-Hessian	600	1418	120
Hessian-Laplace	1000	1979	650
Harris-Laplace	2500	1664	1800
DoG	default	1520	400

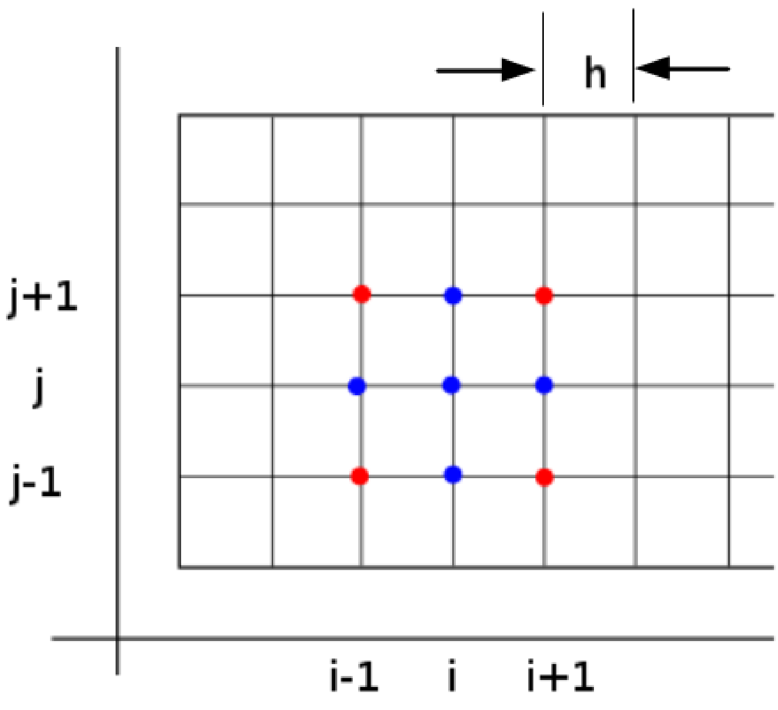
References

1. “SURF: Speeded Up Robust Features,” Herbert Bay, Tinne Tuytelaars, Luc Van Gool, Katholieke Universiteit Leuven

Concepts

- Finite differences

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



$$\phi^n_{ij,x} = \frac{\phi^n_{i+1j} - \phi^n_{i-1j}}{2h}$$

$$\phi^n_{ij,y} = \frac{\phi^n_{ij+1} - \phi^n_{ij-1}}{2h}$$

$$\phi^n_{ij,xx} = \frac{\phi^n_{i+1j} - 2\phi^n_{ij} + 2\phi^n_{i-1j}}{h^2}$$

$$\phi^n_{ij,yy} = \frac{\phi^n_{ij+1} - 2\phi^n_{ij} + 2\phi^n_{ij-1}}{h^2}$$

$$\phi^n_{ij,xy} = \frac{\frac{\phi^n_{i+1j+1} - \phi^n_{i-1j+1}}{2h} - \frac{\phi^n_{i+1j-1} - \phi^n_{i-1j-1}}{2h}}{2h}$$

Concepts

- Finite differences

Generalized definition (forward, central, backward):

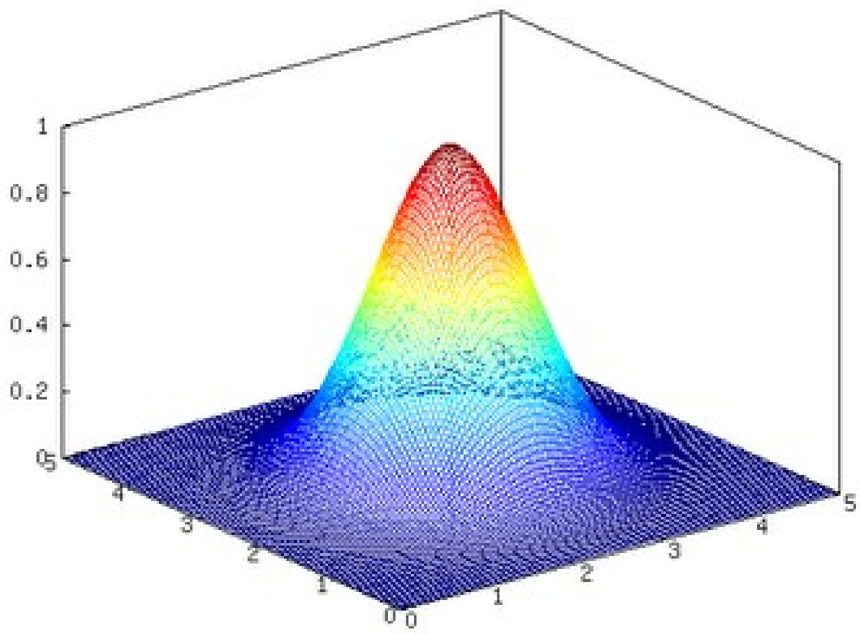
$$\Delta_h^n[f](x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x + (n-i)h),$$

$$\nabla_h^n[f](x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x - ih),$$

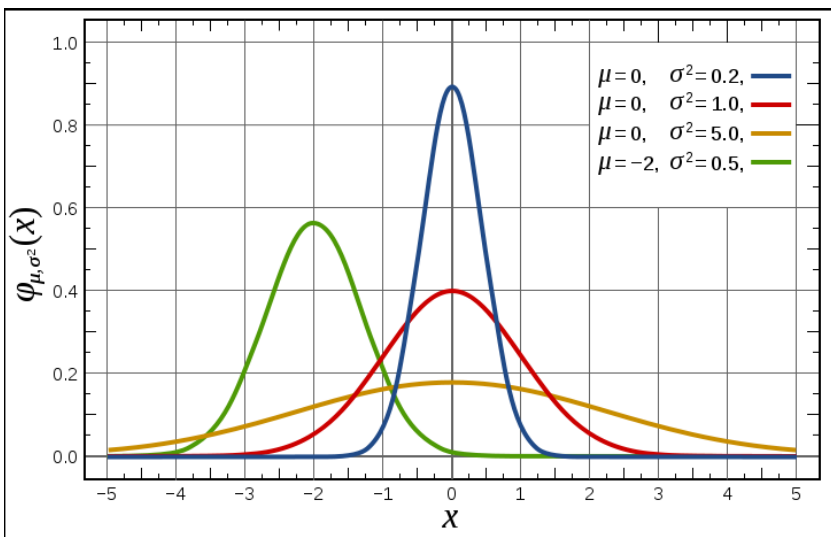
$$\delta_h^n[f](x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f\left(x + \left(\frac{n}{2} - i\right) h\right)$$

Concepts

- Gaussian smoothing



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



Concepts

- Laplacian
  - Sum of the second-order, non-mixed partial derivative of a scalar function
  - Denoted by the “Laplacian operator”

$$\nabla \cdot \nabla f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$