SIFT Object Recognition Algorithm

Outline

- Overview of SIFT
- Applications
- Concepts
- Implementation
- Summary

Overview of SIFT

- SIFT – Scale Invariant Feature Transform
- Robust object detection algorithm
- Works by identifying sets of “features” with specific geometric properties
- These features are stored and compared with known feature sets
- A probability that the objects are identical is returned

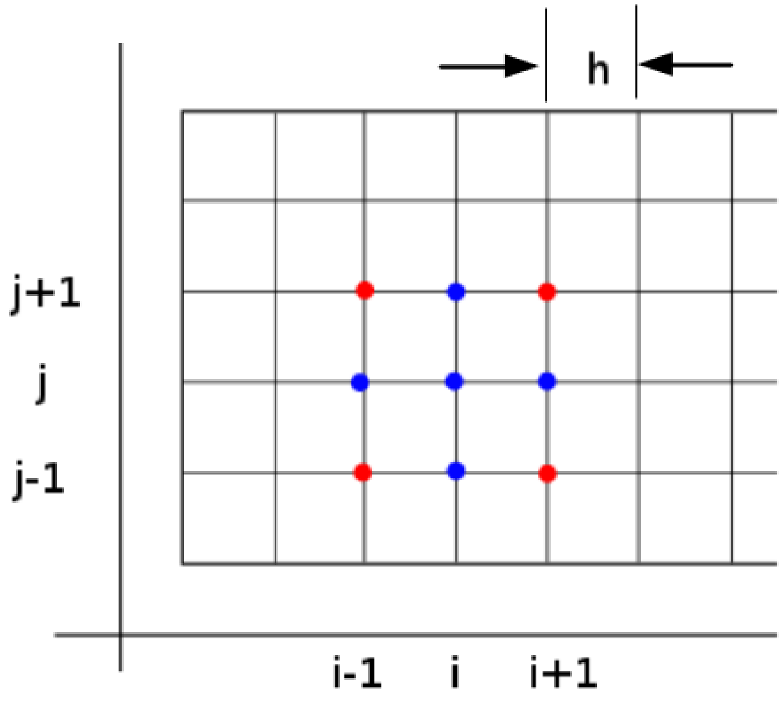
Applications

- Robust detection of objects provided by a feature set which remains relatively invariant to:
  - Lighting conditions
  - Object orientation
  - Distance
  - Partial obscuration
- To achieve these goals, a robust feature set must be determined for the object
- This must be found by trial-and-error (which can be encapsulated well by training procedures including machine learning)

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):

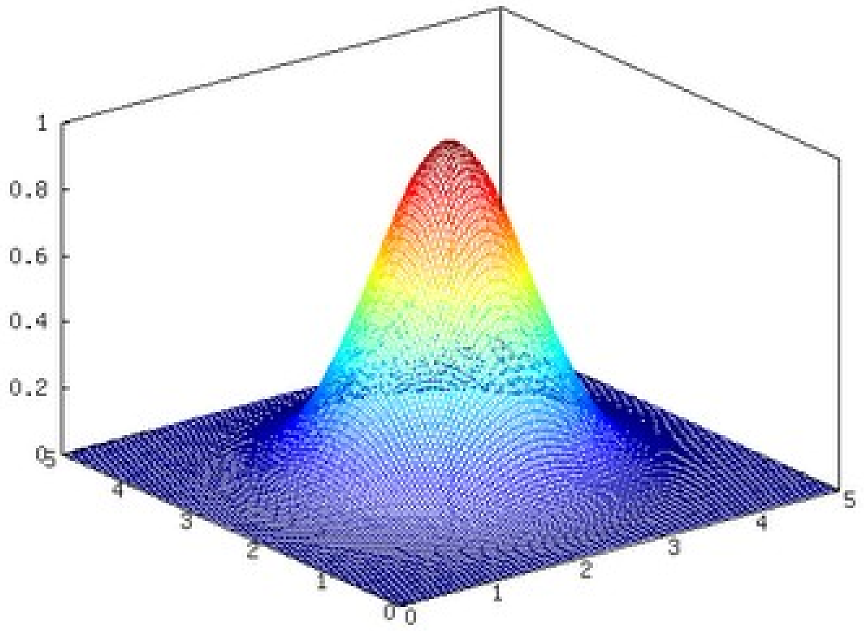
$$\Delta_h^n[f](x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x + (n-i)h),$$

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

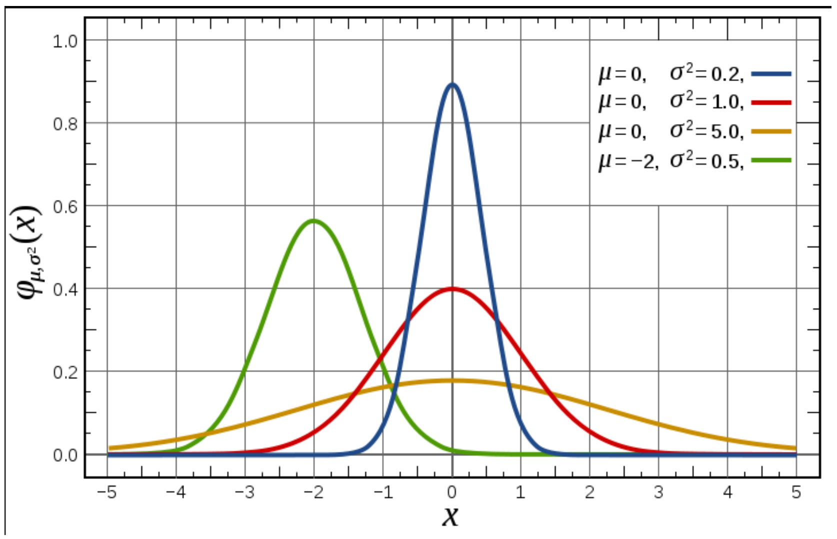
$$\delta_h^n[f](x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f\left(x + \left(\frac{n}{2} - i\right) h\right)$$

Concepts

- Gaussian smoothing



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



Implementation

- Procedure:
  1)Scale-space extrema detection
  2)Keypoint localization
  3)Orientation assignment
  4)Keypoint descriptor
  5)Keypoint matching

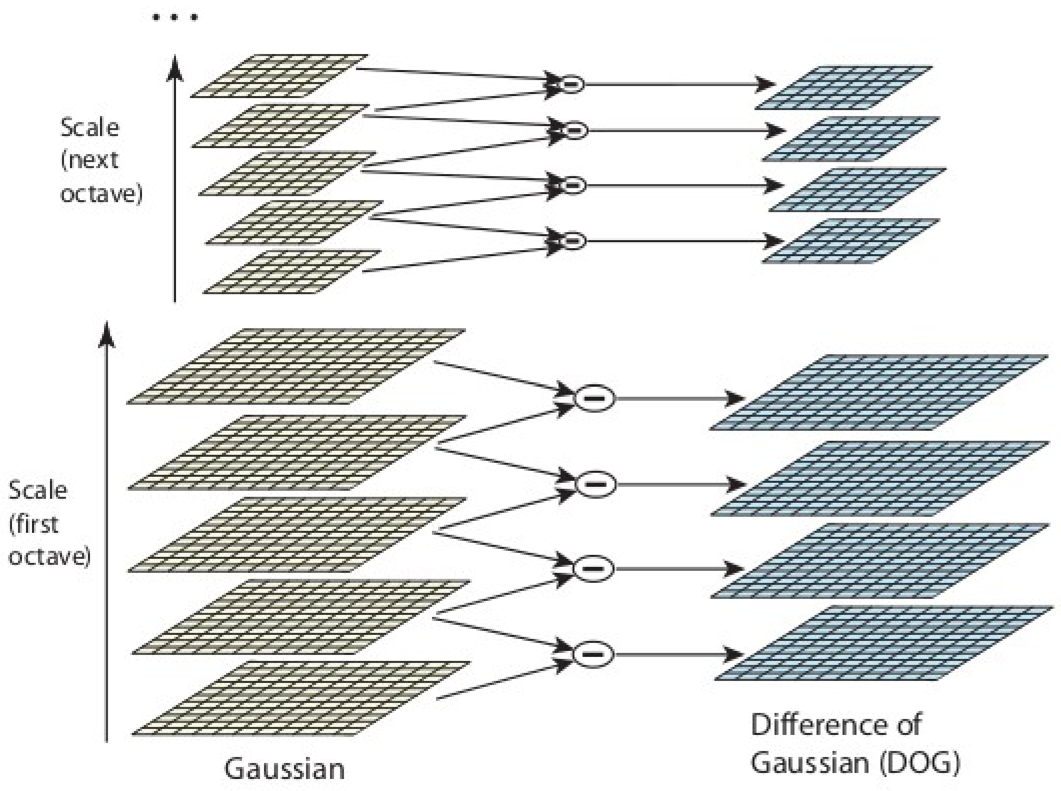
Implementation

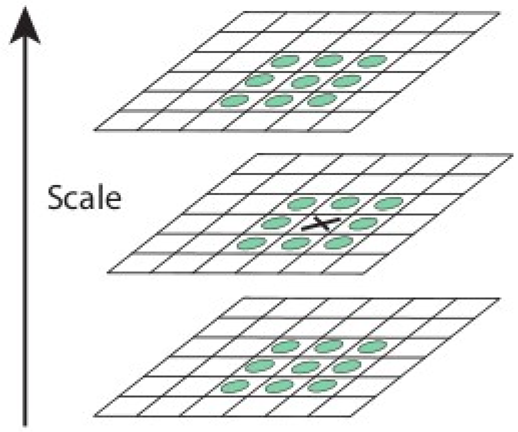
- Scale-space extrema detection
  - Image pre-smoothing (with factor sigma)
  - Image doubling
  - Generate octave
    - Set of images convolved with equivalently-sized Gaussian kernel
    - Gaussian kernels are scaled by k
    - “Gaussian smoothed images”
  - Difference-of-Gaussian
    - Difference between each successive smoothed image
  - Detect extrema
    - Compare each pixel in each DoG with its 26 surrounding neighbors
    - Iff the pixel is greater than or less than all neighbors, then it is an extrema

Implementation

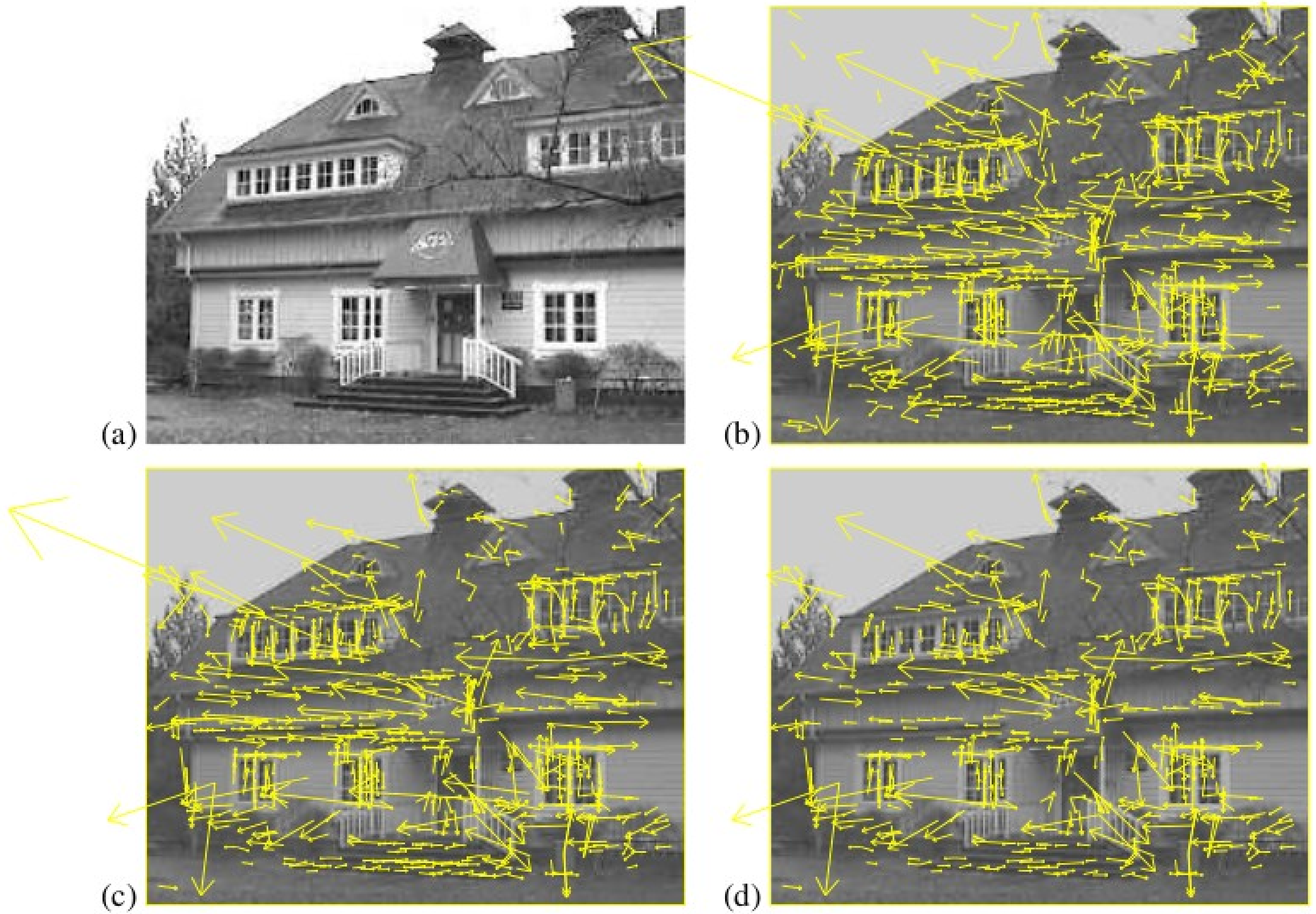
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

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





Implementation



Implementation

- Keypoint localization
  - Apply contrast threshold

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

$$\mathbf{x} = (x, y, \sigma)^T$$

  - Apply threshold based on ratio of principal curvatures

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

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

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Implementation

- Orientation assignment
  - For a given keypoint scale, select the Gaussian smoothed image L with the closest scale
  - Select a region of pixels around (x, y)
  - Populate an orientation histogram with weighted orientations
    - 36 bins (360 degrees)
    - Orientation: $\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$
    - Gradient weight: $m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))}$
    - Gaussian weight:

      Gaussian kernel with sigma = 1.5*(sigma of L)

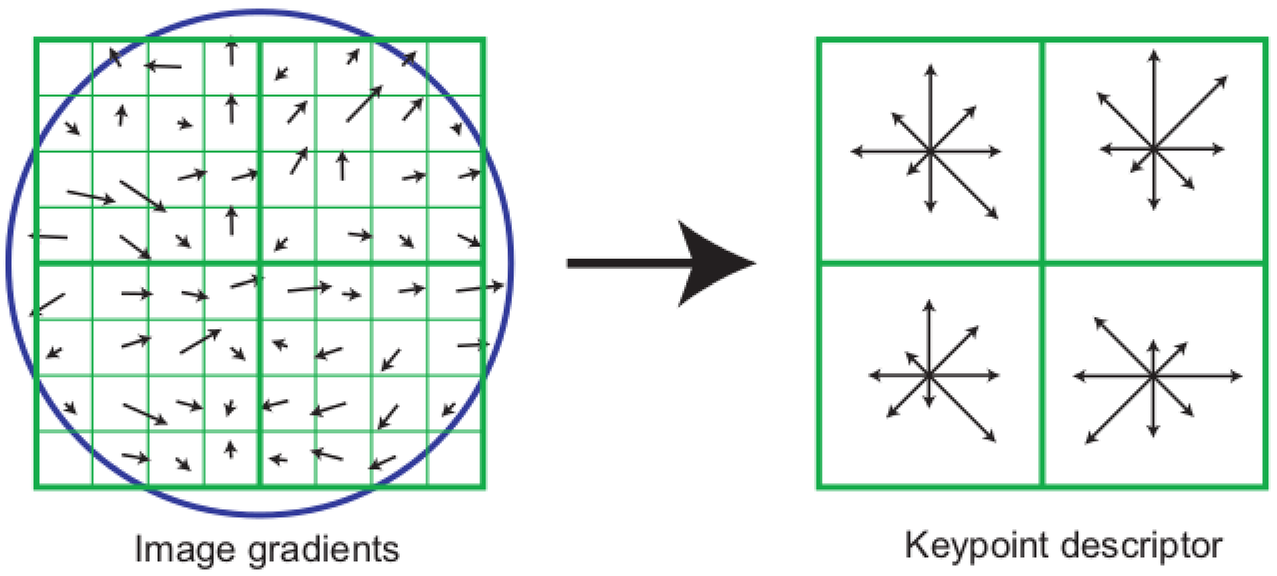
Implementation

- Orientation assignment
  - Voting threshold returns the first and second most popular orientations
  - Cubic interpolation used near each peak to more precisely determine the orientation

Implementation

- Keypoint descriptor
  - At each keypoint, a descriptor array is generated
  - Rectangular region selected around the keypoint pixel
  - Image gradients taken at each pixel in ROI
  - Gradients are weighted by a Gaussian kernel
  - Samples are accumulated in histograms, one each for a 4x4 set of sub-regions
    - Additionally, trilinear interpolation is used to weight samples
    - Weight (1 – d), where d is distance from bin center
  - Each histogram stores 8 gradients
  - Accumulators are thresholded to 0.2 (experimentally determined)
    - This helps reduce the effect of large gradients, and increases importance of gradient orientation distribution
    - The purpose is to overcome nonlinear illumination differences

Implementation



Implementation

- Keypoint descriptor
  - Descriptor consists of the set of accumulator values stored systematically for all bins, in each accumulator
  - In the above image, the descriptor is 2x2x8
  - A better, experimentally-determined descriptor size is 4x4x8

Implementation

- Keypoint descriptor
  - Advantage of this technique includes invariance to shifting gradients
  - 3D rotation of an object would correspond to relative translation of image gradients
  - At a high level, this technique has been shown to accurately identify objects which have been rotated ~20 degrees

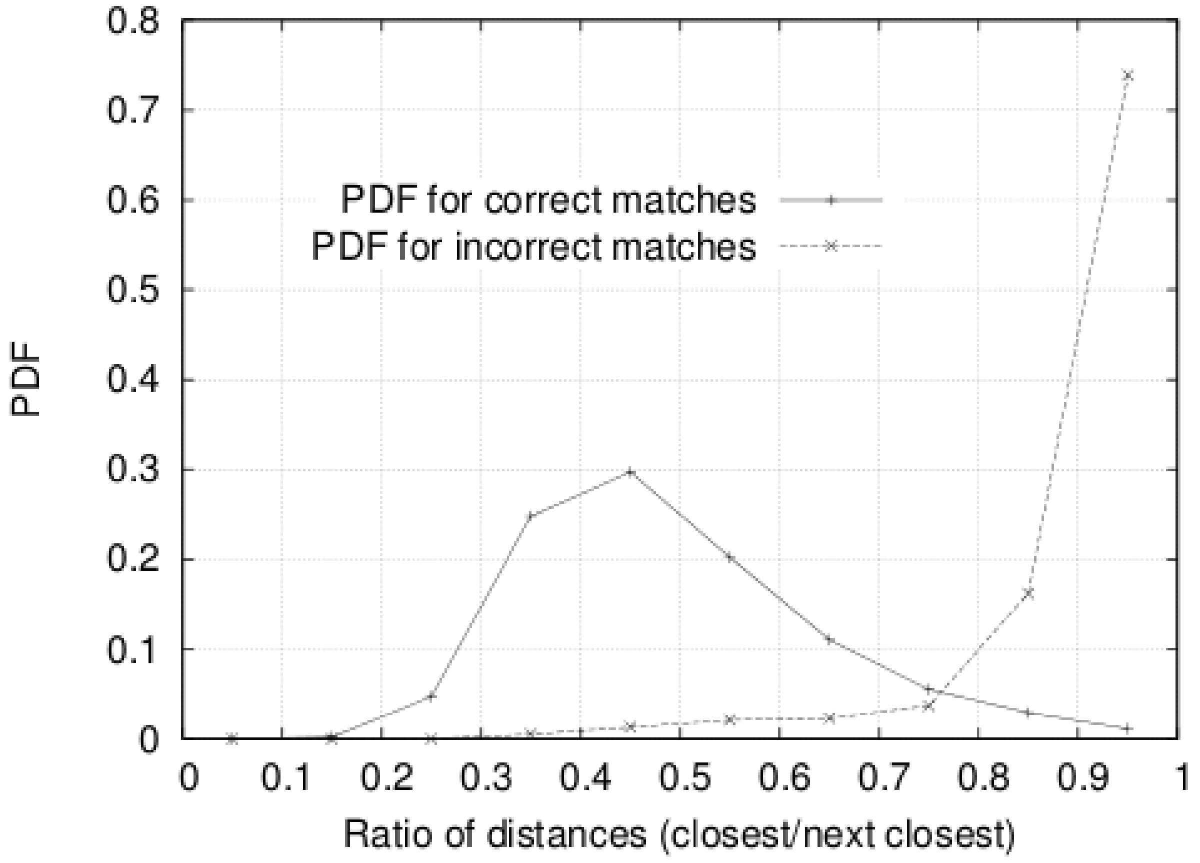
Implementation

- Keypoint matching
  - Begin with one reference image containing an object
  - Place several (> 2) keypoints on the object
    - Ideally, these will be “stable” for the intended application
  - Repeat procedure for each a set of objects
  - This database contains sets of keypoints for each object

Implementation

- Keypoint matching
  - Analyze an image, identifying the keypoints
  - Compare each keypoint with those in the database
  - Simple approach: Call the keypoint with the smallest Euclidean distance a "match"
    - 4x4x8 == 128 == vector degrees of freedom
  - More reliable approach: Identify as a match the keypoint whose second-closest keypoint is very far
    - Distance1 = Euclidean norm between (keypoint) and (database keypoint 1)
    - Distance2 = Euclidean norm between (keypoint) and (database keypoint 2)
    - If |Distance1/Distance2| > threshold, then a match exists
  - Exhaustive search is required, but Lowe[1] uses the approximate Best-Bin-First algorithm

Implementation



Implementation

- Keypoint matching
  - After finding keypoints in the database, spatial relations between the keypoints must be matched
  - Sets of points are used to populate a Hough table in pose space
  - Pose space is 6-dimensional, and refers to the degrees of freedom in which a real-world object can move

Implementation

- Keypoint matching
  - An affine transformation equation is solved with points from the image and database
  - If 3 or more points solve this transormation, then the object is considered a match

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = [\mathbf{A^T A}]^{-1} \mathbf{A^T b}$$

Implementation

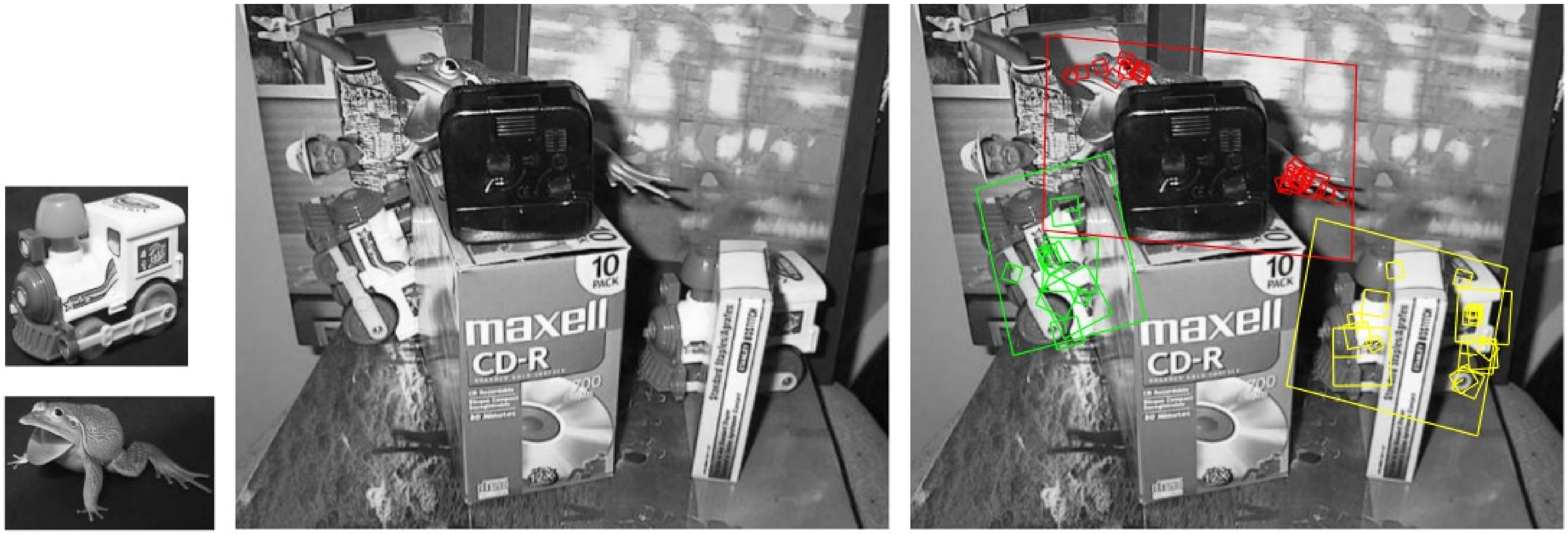


Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Summary

- SIFT is an algorithm which can provide reliable object recognition capability
- Adequate training (or trial-and-error) is required to accomplish robust behavior

References

1. “Distinctive Image Features from Scale-Invariant Keypoints,” David G. Lowe, University of British Columbia