# Extracting Scale & Rotation invariant features for reliable object Matching

(In short: SIFT Implementation)

**Paper Details:**

**Title-** Distinctive Image Features from Scale-Invariant Keypoints

**Author** - David G. Lowe **Publication Year -** November 2004

CV project - Fall 2017

Ajay Ramesh

Chandra Kumar Basavaraju

# Contents

# Problem Statement

# How to detect objects in a image ?

References used in this project.


1. (Paper) Lowe, D.G. International Journal of Computer Vision (2004) 60: 91. https://doi.org/10.1023/B:VISI.0000029664.99615.94
2. Introduction to SIFT (Scale-Invariant Feature Transform)
3. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, Barcelona, 2011, pp. 2564-2571. doi: 10.1109/ICCV.2011.6126544
4. https://www.youtube.com/watch?v=NPcMS49V5hg - UCF Computer Vision Video Lectures 2012
   Instructor: Dr. Mubarak Shah , Subject: Scale-invariant Feature Transform (SIFT)
5. https://en.wikipedia.org/wiki/Lenna - Lena.jpg is used for demo.
6. https://www.learnopencv.com/histogram-of-oriented-gradients/

# How to match an object in image ? - SIFT is one way.

- SIFT Known as Scale invariant feature transform.
  - Corners will look like a curve if you scale the image.
  - Harris Corner Detection fails to detect the objects if Image is scaled.
- Extracted features  are used to match the objects.
  - Property : Invariant to image scale and rotation.
- Used in Reliable object matching.

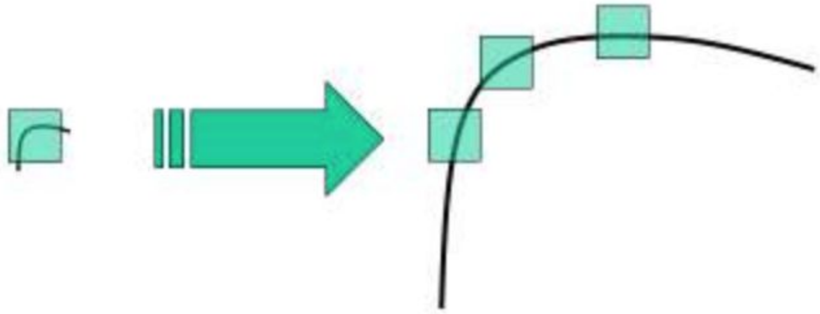


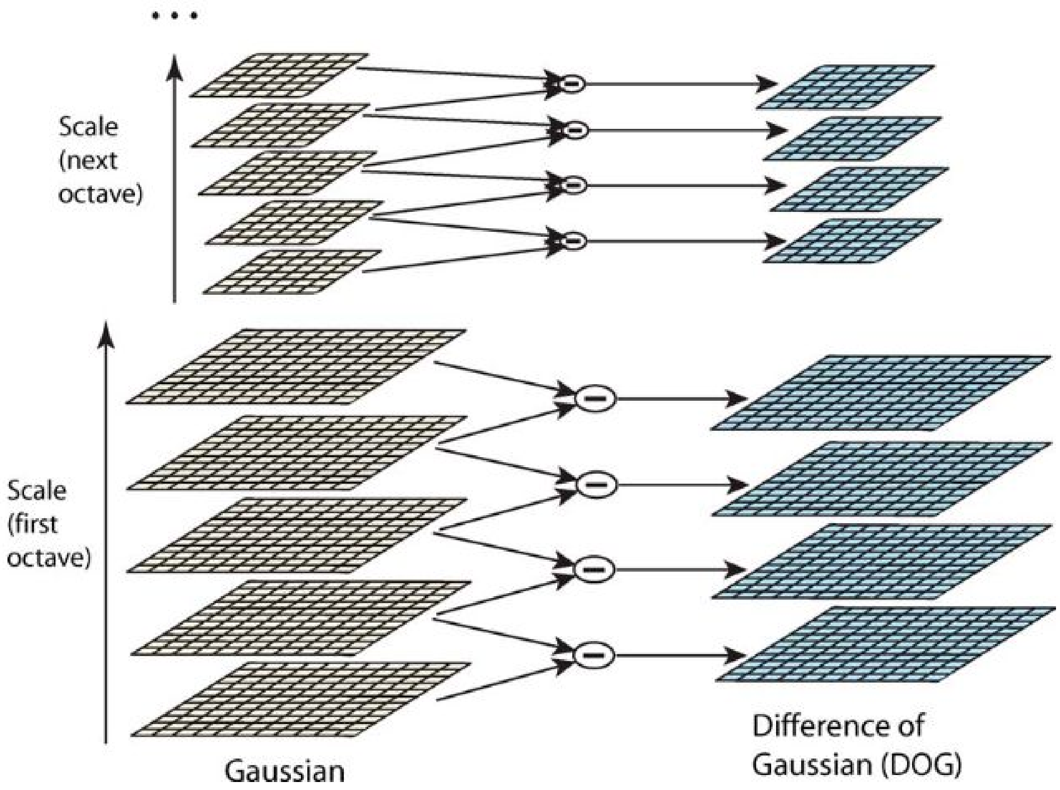
Image source : https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html

# SIFT :: Algorithm:: Outline

Stages in generating the features are as below:

1. **Scale space extrema detection**
   a. Output << different scaled convolved images
2. **Keypoint localization**
   a. Output << Interest points
3. **Orientation assignment**
   a. Output << Orientation and Magnitude
4. **Keypoint description**
   a. Output << feature descriptors.

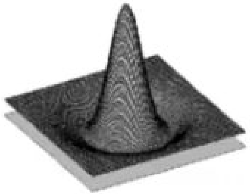
# Scale space extrema detection

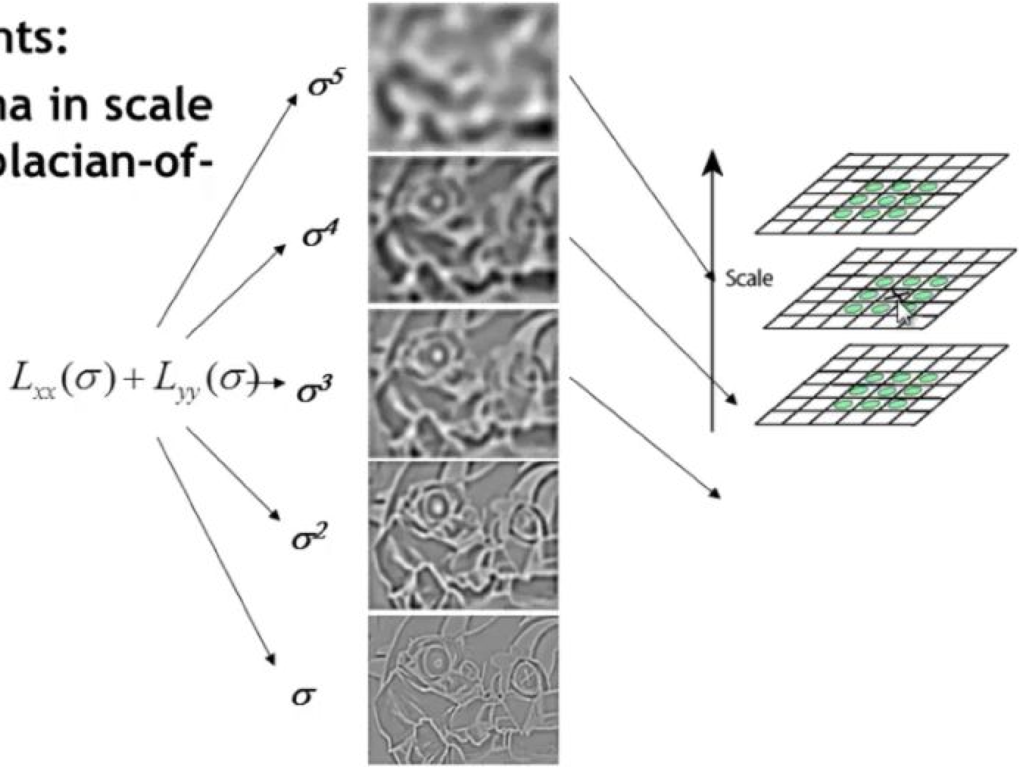


# Laplacian-of-Gaussian (LoG)

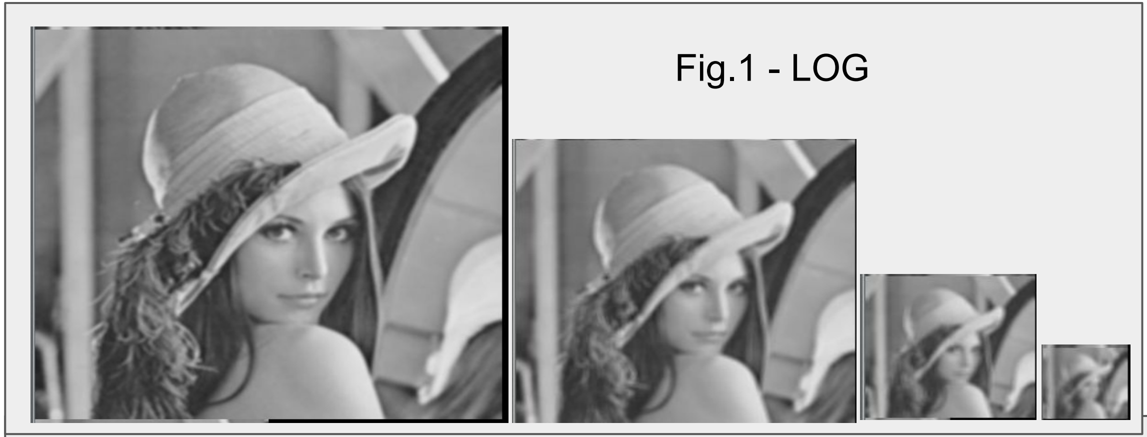
- **Interest points:**

  **Local maxima in scale space of Laplacian-of-Gaussian**





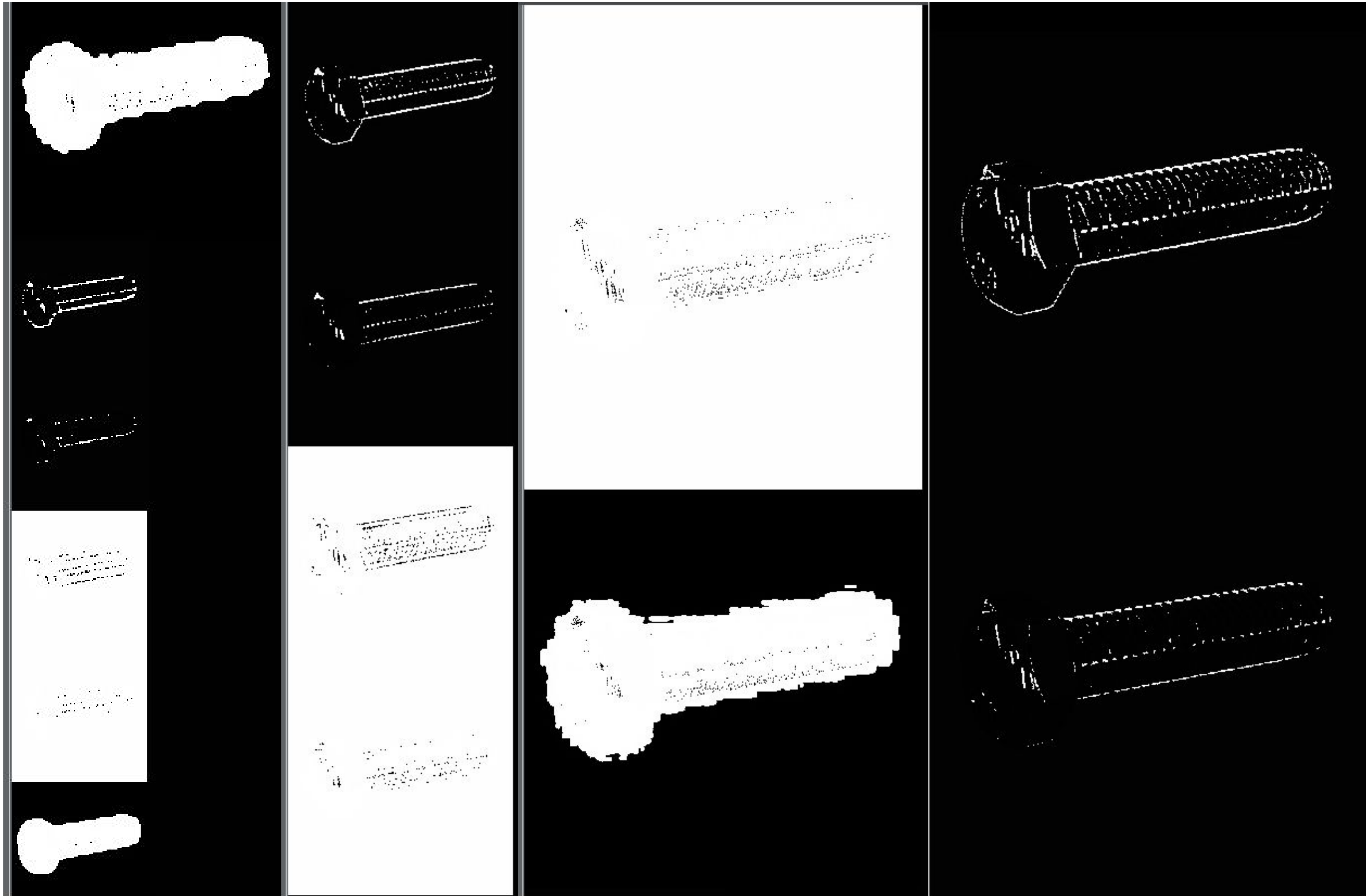
# Output << LOG & Extrema of the image

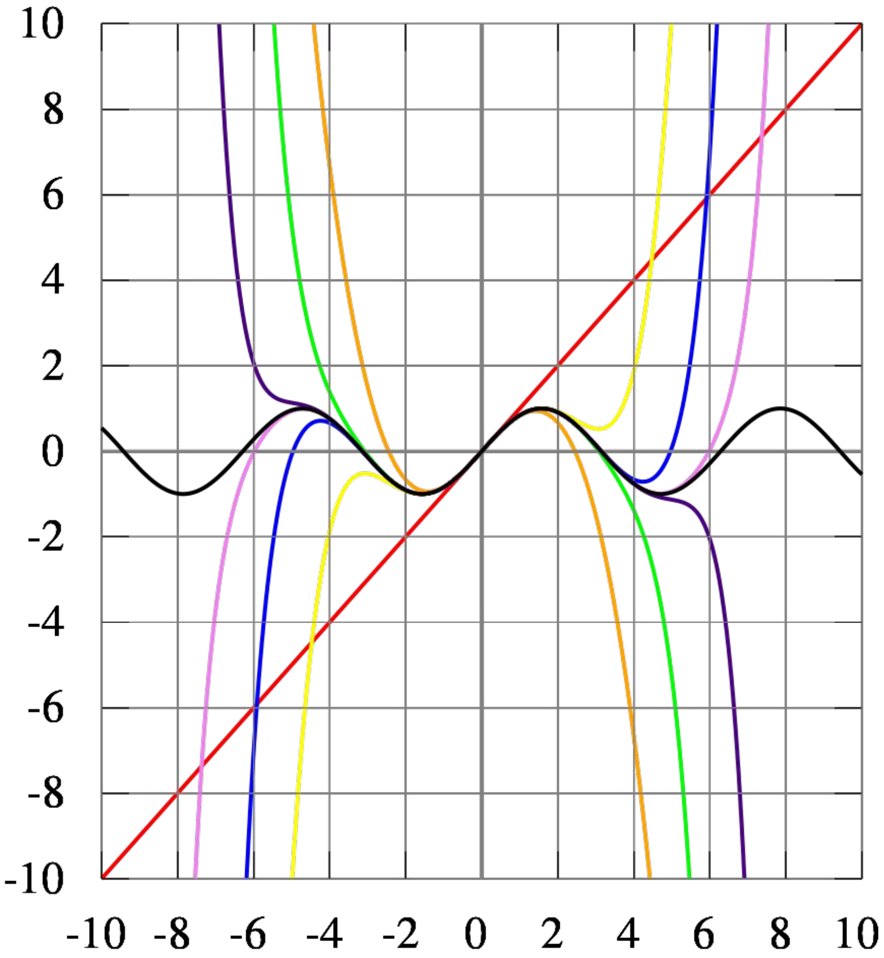




**Paper suggested parameters:**
No of octaves = 4
Scale in each octave = 5
$\sigma = 1.6$

# DOG Samples





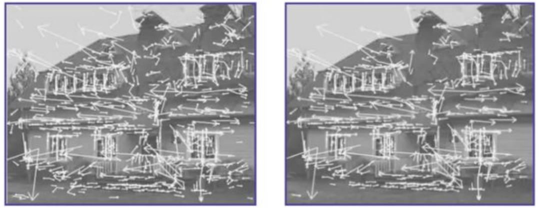

# Introduction :: Taylor Series

As the degree of the Taylor polynomial rises, it approaches the correct function. This image shows $\sin x$ and its Taylor approximations, polynomials of degree 1, 3, 5, 7, 9, 11 and 13.

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}.$$

https://en.wikipedia.org/wiki/Taylor_series

# 2. Keypoint localization



from 832 key points to 729 key points, th=0.03.

1. Initial outlier rejection.
   a. Reject low contract candidates.
   b. Poorly localized candidates along the edge.
2. How?

   A. use Taylor series of DOG, D

   a. A **Taylor series** is a power series representation of an infinitely differentiable function.

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

3. Maxima or Minima located in taylor series at

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

4. Value of D(x) at maxima/minima must be large |D(x)| > th.

# Outlier rejections.

1. DOG is having too good for edges.
2. Similar to Harris-Corner Detection, we can use Hessian of D.

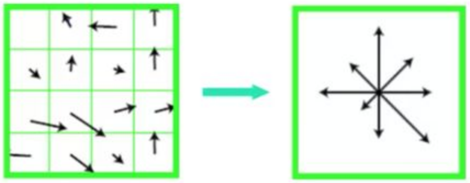
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

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

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

3. Remove outliers if r > 10.

# 3. Orientation assignment



1. To achieve rotation invariance. We need to compute orientation assignment.
2. Compute the central derivatives and gradient and its magnitude at each keypoint location.

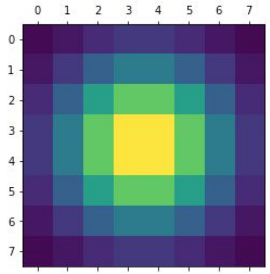
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

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

4. Compute 36 bin histogram.

The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

# 4. Keypoint description



Gauss Weights window.

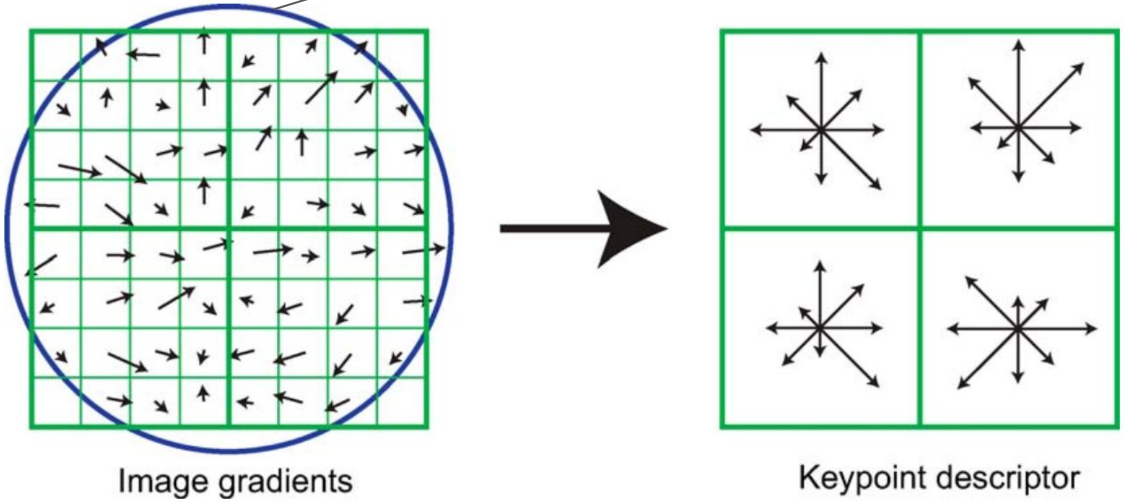
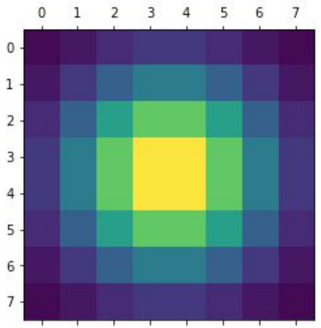



*Figure 7.* A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a $2 \times 2$ descriptor array computed from an $8 \times 8$ set of samples, whereas the experiments in this paper use $4 \times 4$ descriptors computed from a $16 \times 16$ sample array.

# 4. Example

```
Orientation angles(in degrees)

         ___________
        |30|33 | 45|
        | .| . | . |
O =     | .|55 |22 |
        |4 |_._|_._|
```

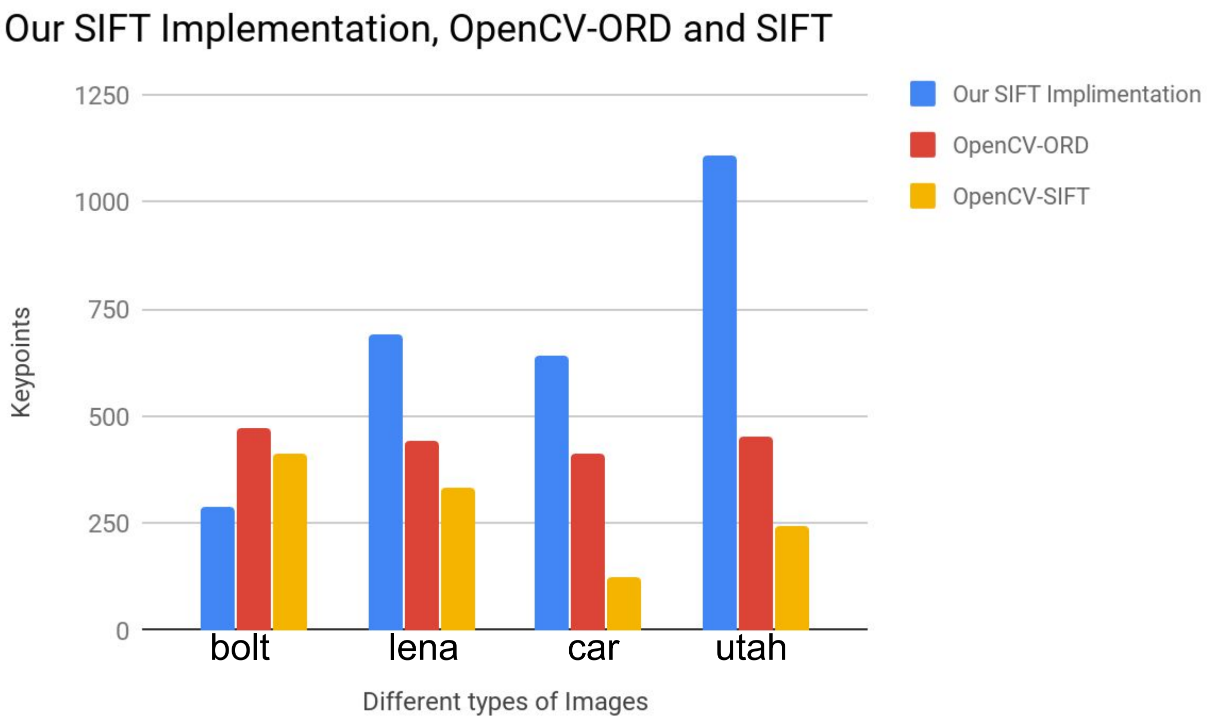
```
Step 1: Create histogram

             |
             |
             |
     4       |     2
    ---------|----------
    0-30         0-60
```



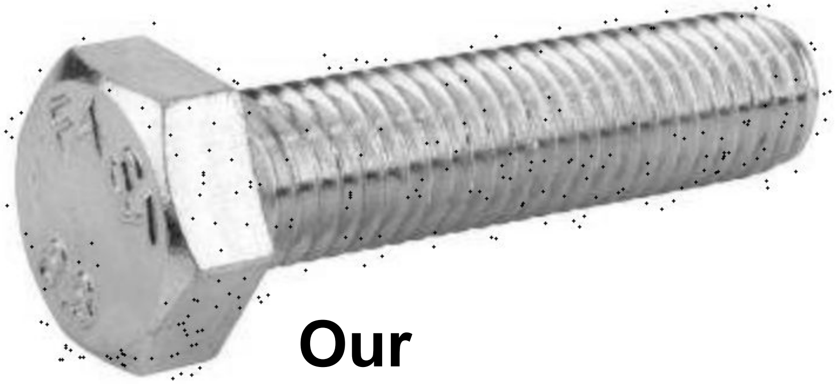

Gaussian weight

```
Step 2: Add weights to above histogram

Weights: add this to above histogram

          _________
         |1 |3  | 1 |
         | .| . | . |
 W =     | .|1  |2  |
         |1.|_._|_._|

Algorithm:
  1. Scan the Orientation matrix(O).
       for each angle in O, add the weight to corresponding bin.

       Example:
           O[0, 0] = 30, so add the weight 1 to bin 0 - 30.
           O[0, 1] = 33, so add the weight 3 to bin 0 - 30.
           O[1, 2] = 22, so add the weight 2 to bin 0 - 30.
           O[3, 1] = 4 , so add the weight 1 to bin 0 - 30.

           Total weights for 0-30 bin = (1 + 3 + 2 + 1 ) = 7

       HOG is below. Histogram which takes
       weights and gradient directions.

           ^    |
           ^    |
           ^    |
           ^    |
           ^    |
           ^    |     ^
           ^    |     ^
       ---------|----------
         0-30     0-60
```

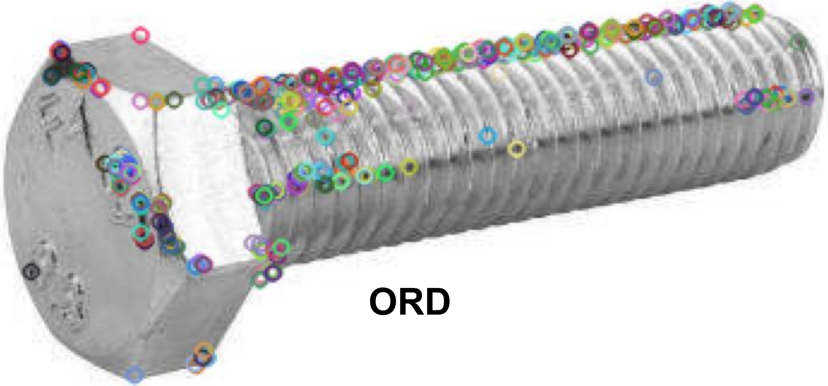
# Results



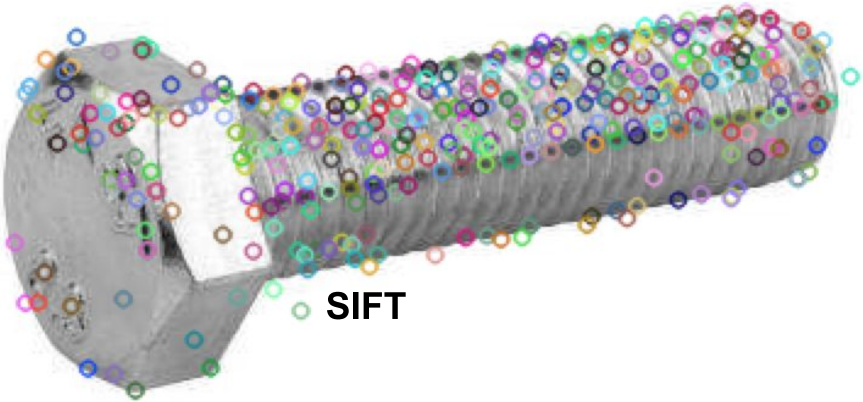
Our SIFT Implementation, OpenCV-ORD and SIFT
Keypoints
1250
1000
750
500
250
0
bolt
lena
car
utah
Different types of Images
Our SIFT Implimentation
OpenCV-ORD
OpenCV-SIFT

**Our Implementation**



**ORD**



**SIFT**

# Final Output

# Questions

Thank you