SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

Introduction to SIFT

This methods innovated by Lowe in 2004. That include verycomprhensive and complicated because of it includes variety of object recegnotion methods and implementations. So obviously it has very complex algorithm. Main goal is image matching from big data image or small image the size it is not important because SIFT always extract fetures from data with localize variable and more than one descriptions. So Firstly *we will extracting distinctive invariant feature and correctly matching against a large database of features from many image.[1]* Secondly, some method in object recognition care about the invariance to image scale and rotation for example Harris Detector, but SIFT don't care rotation or scale magnitude abour the image. So this ability is many important for us. Because *the features are invariant image scaling and rotation, partially invariant to change in illumination and 3D camera point.[2]* Actually SIFT represents a very expensive and cost method but the other side it represent a very highly propability object recognition and image matching method. Also it has some advantages from to oother methods, -robutness to affine transformatin and image in 3D viewpoint.

--Advanteges of SIFT[3]
1-Locality: features are local, so robust to occlusion and clutter
2-Distincveness: individual features can be matched to a large database of objects
3-Quantity: many features generated for small objects

--Algorithm Steps
1- Scale Space Extrema Detection
2- Keypoint Localization
3- Orientation Assignment
4- Keypoint Descriptor

This method mainly focus on 4 step but each step has 3-4 algortihm operation and we explain each of them sequentially. This steps has a lot of mathematical operation and complex schema, but I think Sıft totally 15-20 algorithm step, yes it is actually expensive and cost method but the result is very satisfy our purpose.

1- Scale Space Extrema Detection

Scale Space

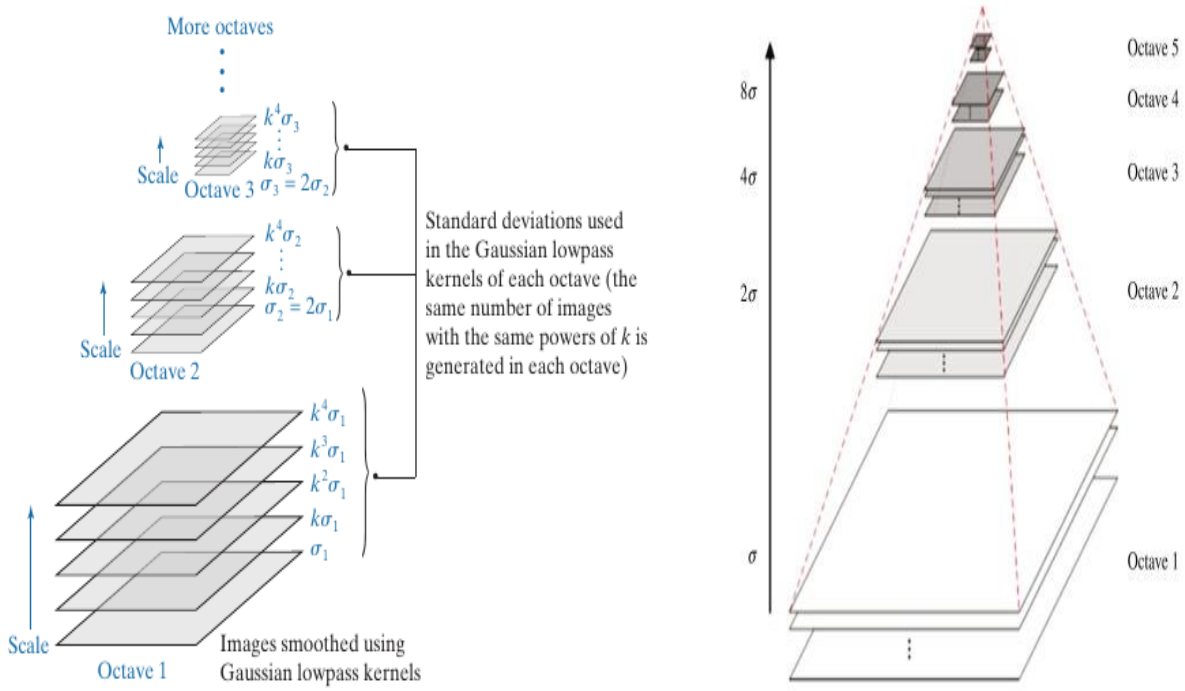
Scale-space firstly used in 1983 by Witkin[4] and he saw some important results for image prcessing. Witkin firstly apply whole spectrum of scales, then plot to recrossing areas versus scales. Then interpret scale space counters and draw a interval tree, and used the many many scale with sigma of Gaussion filters.

We should know SIFT method's input parameter is an image and its output is an n-dimensional feature vector whose elements has invariant feature descriptors. If you want to find which image locations has the invariant to scale change, you should search for stable features across all the possible scales with using function of scale known as Scale-Space or multiscale. Scale-Space represent how to combine different many image scales. And it has one parameter type of smoothed image. The parameter controlling the smoothness for image and all the other scales. That name is the scale parameter. So;

$$L(x, y, \sigma) = G(x, y, \sigma) \star f(x, y)$$

On the function L represent scale of image(I) and G represent the Gaussion kernel. Our input image is a Gray-Scale format. We know the Gaussion kernel from our courses. It provides low-pass smoothed image for us. In this step we concolve our image I(x,y) with Gaussion kernel. We know the Convolve operation start from the kernel's bottom-right part and we use this function for all the pixels and obtain a scale space for our image. *But this is not suffiecient for SIFT. We also should convolve with G that standart deviation parameter is equally and sequentially with constant "k", and k has a power of k=0,1,2,3,...., and so we obtain many different and separeted scale space for one octave each has ((k power of i) x (std)).*[5]

$$\sigma, k\sigma, k^2\sigma, k^3\sigma, \ldots$$



SIFT method divide scale-space to octaves. Each octaves uses for calculating and obtain scale image(L) a doubling of standart deviation form previous space. This octaves behaviour like the music frequency and denoted. Also we use some interval value "s", and s=1 meaning is consist 2 images and s=2 meaning is consist 3 images. We conclude image number is equally likely (s+1) for an octave.

We should DoG(Differences of Gaussion), also octave is known (s+1)th image in the stack. Then second octave first image is formed by downsampling the original image then smoothing it using a kernel with twice standart deviation.

Another important point for octave level, for each octave level image size is halving and it appear like the pyramid formation. The third octave image from the top of any octave is known "Octave Image" because the standart deviation used to smooth it is twice. We should observe octave grow up, then the image lose more fine detail. So the third octave image has fewer details, but their apperance is unmistakebenly has the same structure.

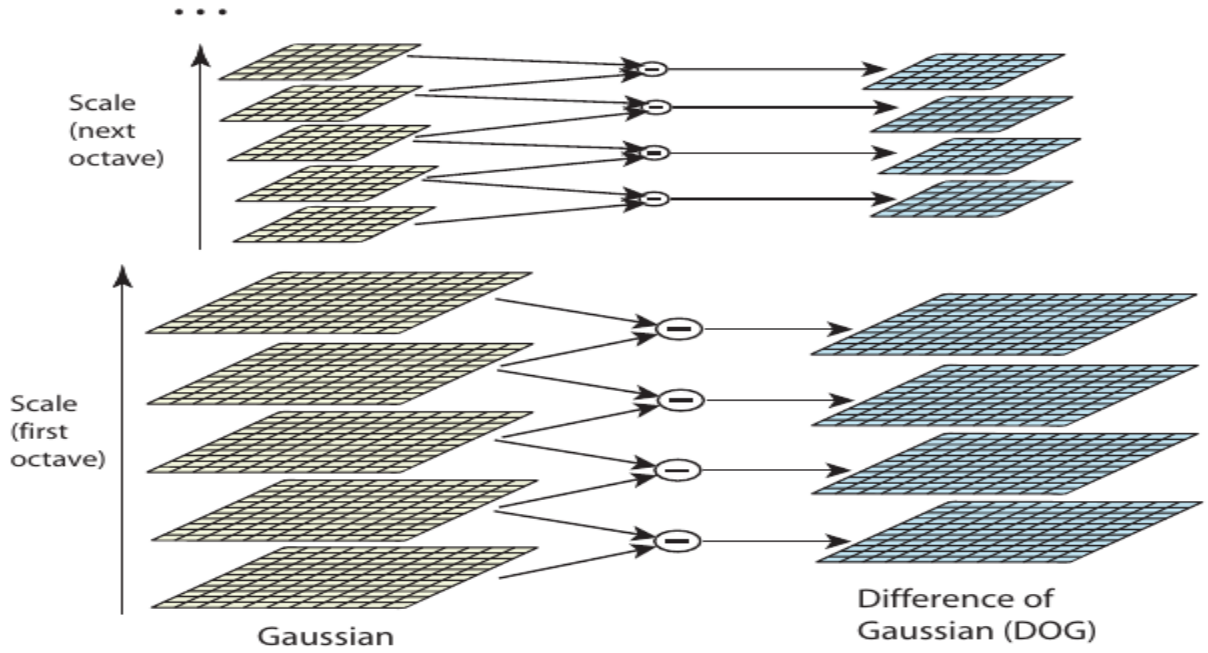
Detecting Local Extrema

This level has 2 steps; First, to find location of the keypoint with using Gaussian kernel. Second, to refine the locations and validity of those keypoints using two processings.

Detecting extrema generally shows in the difference of gaussian of two adjacent scale-space image in an octave, convolved with input image that corresponds to that octave

$$D(x, y, \sigma) = \left[G(x, y, k\sigma) - G(x, y, \sigma)\right] \star f(x, y)$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$



Actually we see the DoG is obtained subtracting two Scale-space(L). And we obtain (number of octave image -1) DoG images. Additionally we should know DoG function has close approximation to the scale-normalized Laplacian of gaussian(LoG). [6]

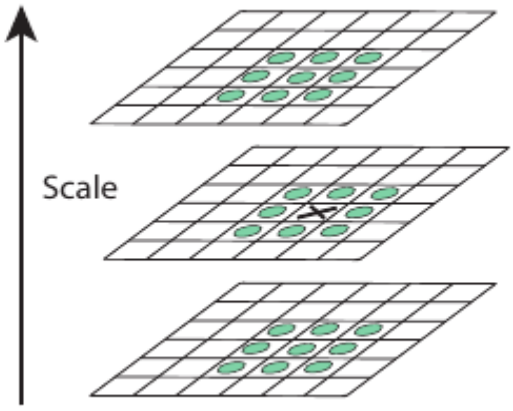
$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2\nabla^2 G.$$

The factor (k-1) in equation is a constant over all scales and therefore does not affected extrema location.[7] The approximation error will go to zero as k goes to 1, but in practice we use the k value of squart root of 2.

2- Keypoint Localization

Improve accuracy of keypoint Locations



How to find extrema locations? For example we have obtained some DoG images in each octave and we have a 3 space for them like the left-side image. And we search extrema point for pixel "X". The maxima and minima of DoG images detectable by comparing pixel of its 26 neighbors. If we obtain an extrema min or max we write the this value for this pixel.

If we think continuous function as an octave image that is true any maximum or minimum point actually be located between sample points. And on the left-side image shows the where the Extrema of the D(x,y,o) images in an octave.

Now we obtained a candidate keypoint by comparing neighboring. We can go to next step, *to perform a detailed fit to nearby data for location, scale and ratio of principal curvatures.*[8] this step rejected 2 arguments one of them low contrast the other is poorly localized along an edge. In this step we use the Taylor expansion of the scale-space function, DoG, shifted so that origin is our sample point.

$$D(\mathbf{x}) = D + \left(\frac{\partial D}{\partial \mathbf{x}}\right)^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial D}{\partial \mathbf{x}}\right)\mathbf{x}$$
$$= D + (\nabla D)^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H}\,\mathbf{x}$$

Where D is the derivatives are evaluated from x = (x,y,o)T
and the triangle symbol represent the gradient vector function. H is also known Hessian matrix.

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

And the X value shows min or max extrema location point.

Eliminating Edge Response

How we find this? We will write step by step solution again;

- *use Taylor expansion of DoG*
- obtain X(min or max location) from above formula
- *value of D(x) at minima or maxima must be large then threshold point 0.03*,* otherwise you must remove them
- DoG has strong response along edge, assume like surface of images. Then compute principal curvatures(PC) and along the edge one of the PC is very low, across the edge is high like the Harris Detector method.

To quantity the difference between edges and corners we should see and local curvatures. Generally edge meaning is high curvature is one direction and low curvature in the orthogonal direction. Estimate the local curvature of the DoG at any level in scale-space,

- *then Compute the Hessian Matrix of determinant*

$$\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 \end{bmatrix} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

- remove outliers by evaluating Tr(H) and Det(H)

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$$

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

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

- eliminate keypoints if r > 10.

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

3- Orientation Assignment

Now we know location of each keypoint. In Scale-space we have achieved scale independency. Our next step is assign a consistent orientation to each keypoint with related to local image properties. For each image sample L(x,y) we should compute m(x,y) -gradient magnitude- and Q(x,y) -orientation angle- with using the pixel differences.

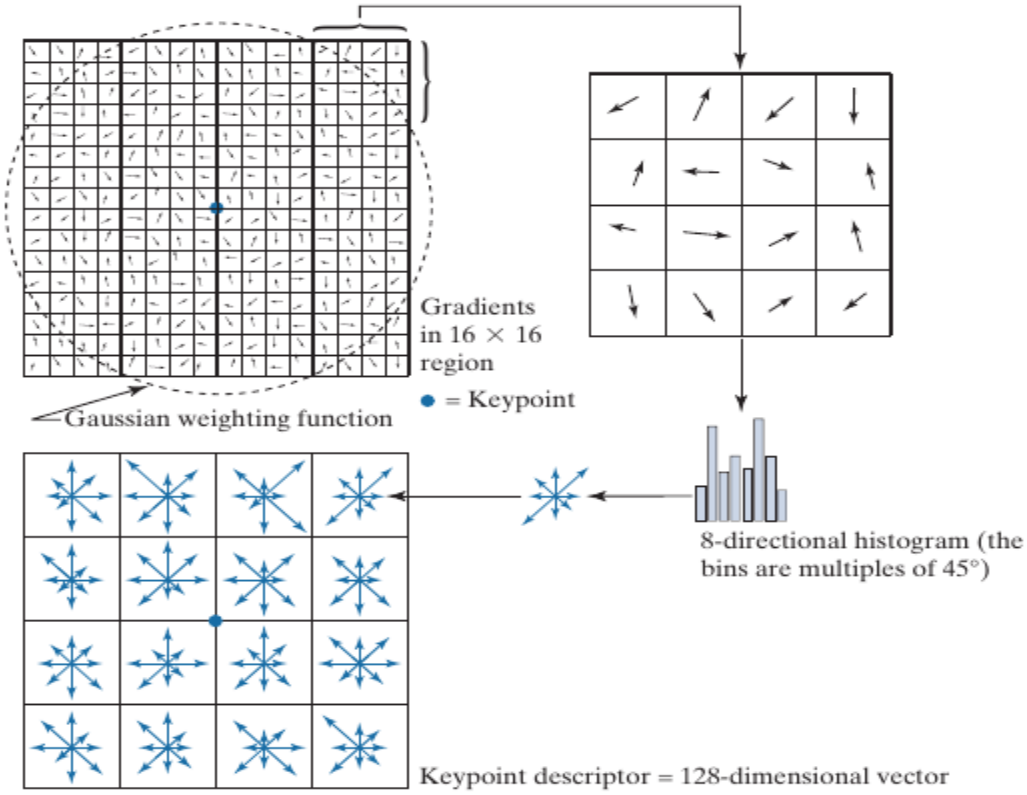
$$M(x,y) = \left[\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2 \right]^{\frac{1}{2}}$$

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

An orientation of histogram is formed from the gradient orientations of sample points in a neighbored of each keypoint. The orientation histogram has 36 bins covering 360 degree range of orientations. Each sample added to our histogram is weighted by its M value and a Gaussion wieghted circular schema with standart deviation with multiple 1.5 times that is the scale of keypoint. SIFT features are dedicted to large databse pixel noise and big error is the starting location and scale detection.

4- Keypoint Descriptor

At this point until now we scaled the image then orientated to each keypoint thus providing invariance these variables. Our next assignment computing a descriptor for our image. We use weighted circular shaped with gaussion kernel and gradient vector table for pixels then we should obtain each 4x4 gradient vector for one keypoint desciptor. Also we could make with using a 8 bins histogram. We can see each descriptor gradient vector have different magnitude and angle maybe some of them very distinctive and huge some of them is very small like the 3d visuluation gradient vectos combination. This keypoints



obtained from our previous step. For a local region around each keypoint that is highly distinctive but at the same time as invariant as possible to change in scale, orientation, illumination and image viewpoint. The idea able to use this descriptors to identify similarity between local regions in two or more images.

Conclusions

SIFT method siad that keypoints described in this paper are particularly useful due to their distinctiveness, which enables the correct match for a keupoint to be selected from a large database of other keypoints.[9] this succesfullnes is depend on a high dimensinoal vector from gradients within a local region of the image. The keypoints shows invariant to image rotation and scale and robust across a substantial range of affine distortion, addition noise and change illuination.

For example above the left image has some affine transformation, scaling, rotation, 3d viewpoint changing some losesses detailing for to right image. But if we found the keypoints and their keypoint descriptor in a large databse image we should obtain a similarities for object regonition perspective via the our SIFT method application.

The fact that keypoints are found on the tange of scales meaning is small local features are available for matching small highly occluded objects, while large keypoints well for images subject to noise and blur.

[1] Dr. Mubarek Shah, Computer Vision, Uni of Central Florida, youtube.
[2] David G. Lowe, Distintive Image Features from Scale-Invariant Keypoints, Uni. Of British Columbia, 2004
[3] Dr. Mubarek Shah, SIFT method representations, Uni of Central Florida, youtube.
[4] Witkin, A. P. "Scale-space filtering", Proc. 8th Int. Joint Conf. Art. Intell., 1983
[5] Yao Song, Weidang Chai, Computer vision for microscopy image analyses, 2021
[6] Lindeberg, Laplacian of Gaussian, 1994
[7] David G. Lowe, Distintive Image Features from Scale-Invariant Keypoints, Uni. Of British Columbia, 2004
[8] David G. Lowe, Implementation of keypoint accuracy, Uni. Of British Columbia, 1999
[9] David G. Lowe, Distintive Image Features from Scale-Invariant Keypoints, Uni. Of British Columbia, 2004