Keypoint Detection with SIFT

Daniel Amirtharaj
damirtha@buffalo.edu

1 Objective

To detect keypoints in the given image by implementing the Scale Invariant Feature Transform (SIFT) algorithm (in Python).

2 Analysis

2.1 SIFT

SIFT is a robust feature detection algorithm for feature detection and matching. It is not susceptible to sclae, intensity, rotation or affine changes in 2 corresponding images, and hence is a great tool to detect features in real-world applications.
The following steps are followed in this algorithm,

1. Scale space generation, by creating 4 octaves by halving sizes of original image and progressively blurring each image in the octave to create 5 blurred images in each octave.
2. LoG approximation using Difference of Gaussian, generating 4 DoGs per octave.
3. Identification of extrema points, by comparing a pixel with neighbouring pixels in adjacent scales as well.
4. Removal of bad extrema points, such as points with low contrast using a corner detector algorithm.
5. Identification of keypoint orientations, by finding gradient orientations of neighbouring pixels and fixing an orientation based on the histogram of rotations.
6. Generation of feature descriptors, encodes the orientation observed into a 128 bit vector. This can be used to match features in different images.

2.2 Gaussian Blur and DoG

The Gaussian kernel is used to blur an image by calculating an average of pixels and its neighbours within a particular window size. It is usefule here since LoGs can be approximated as a difference of gaussians. The gaussian distribution in 2 dimensions is given by,

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

The Gaussian kernel is obtained by sampling the above distribution along discrete intervals and Gaussian blur of the image is obtained by convolving the gaussian kernel with the image,

$$O(x,y) = \sum_{(x',y')} G(x',y') * I(x-x', y-y'),\ x'\epsilon(-k,k)\ and\ y'\epsilon(-k,k)$$

Here, O is the output image, S the Sobel operator and I the input image.
The difference of gaussian blurs approximate to the Laplacian of gaussians, and hence is used here to approximate LoGs required to be found in SIFT.

3 Method

The following steps were used to obtain keypoints from the first 3 steps of SIFT.

1. Generate 4 octaves by progressively halving the height and width of the image.
2. For each image in the 4 octaves, progressively blur the image to form 5 blurred images for each octave, according to the sigma values given (which differ by a constant factor). This is done in the following steps,
    1. Zero pad the input image (in grayscale) with one row/column of zeroes, on all 4 edges, such that the resulting resolution will be $(h+2)$x$(w+2)$, where hxw is the dimensions of the input image.
    2. Compute the gaussian kernel, given the size of the kernel and sigma and normalize its values.
    3. Convolve the gaussian kernel with the corresponding window of the input image, for each input pixel, to get the value of the output pixel, resulting in a blurred image.
4. Generate the difference of gaussians, by taking the difference of each adjacent image in the scale space. This will give 4 DoGs per octave.
5. Find extrema points by comparing the value of each pixel with its neighbours, both in the same image and the adjacent scale spaces. This will give 2 sets of keypoints for each octave.
6. Overlay the keypoints detected in each octave to finally obtain 4 different sets of keypoints, one for each octave.

4 Results

The following images, were obtained at various steps of the program, and were required to be presented in the report.

4.1 Images of the 2nd and 3rd octaves

These images were obtained once the gaussian blur was applied and 5 scales were generated for each of the 4 octaves. These are given in figure 1. Octave 2 has images with a resolution of 375x229 and Octave 3 has images with a resolution of 188x115.

4.2 DoG Images of the 2nd and 3rd Octave

These images were obtained after the difference of gaussians was generated for each adjacent scale in the scale space of each octave. These are given by figure 2.

4.3 Keypoints detected

These images, give the keypoints detected for each octave, and have the keypoints overlayed on the original image. These are shown in figure 3.

Figure 1: The scale space of octaves 2 and 3 respectively.

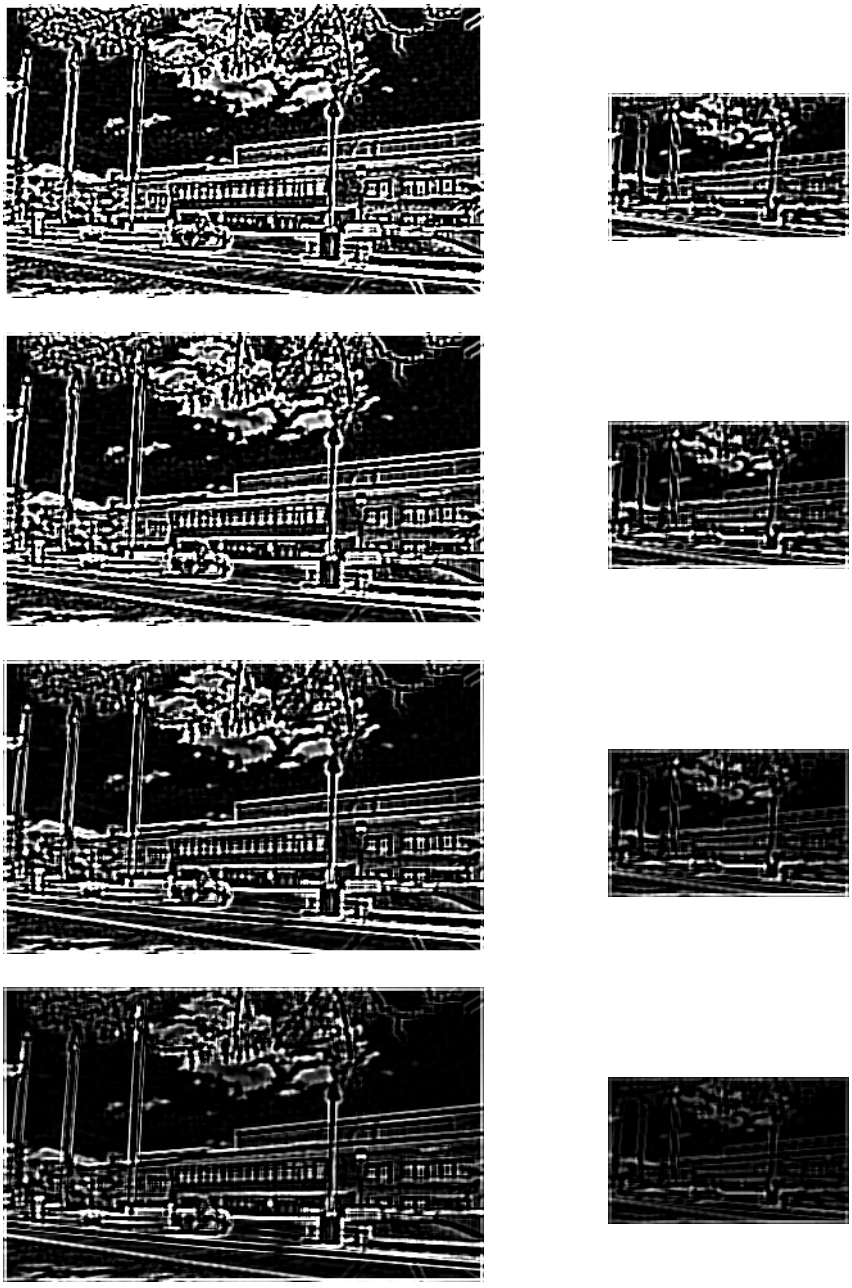

Figure 2: Difference of Gaussian images obtained from octaves 2 and 3.

Figure 3: Keypoints for 4 octaves, overlayed on the original image. In this fig. they are octaves 1 through 4 from top to bottom.