Visual Understanding 1 Coursework

Garoe Dorta-Perez

CM50248: Visual Understanding 1

Unit Leader: Peter Hall

Bath University

January 13, 2015

1 Introduction

The main objectives of this unit are to acquire the fundamentals of Computer Vision. In order to gather this skills, we will attempt to build a stereo reconstruction system. Using a bottom-up approach the following steps will be taken:

1. Image convolution, transforming using kernel matrices.

2. Feature construction, detecting scale invariant interest points in images.

3. Matching, using the previous points to match two images.

4. Stereo reconstruction, rebuilding 3D points from 2D matched points in stereo images.

2 Literature review

3 Convolution

Transforming images has artistic and practical applications, for instance blurring certain parts in a person face to hide defects or detecting borders with differential operators. For simplicity we will restrict ourselves to gray images, defined as two-dimensional $n \times m$ matrices where each element represents the intensity, and is in the range $\{0, 1\}$ where 0 is black and 1 is white. A convolution kernel k, is defined as a $k_1 \times k_2$ matrix, such that $k_1 \leqslant n$ and $k_2 \leqslant m$. When an image I_0 is convoluted with a kernel, a new image I_1 is generated, such that each element in the second image is calculated as follows WRITE IT AS EQUATION OR EXPLAIN IT OTHERWISE. The advantage in this definition is that, it gives a common framework for every possible transformation; so we only need to change the kernel to fit our needs. However, we need to specify what will happen to the borders of the matrix, leave them as they are, set them to zero, compute circularly, etc; and where does the convolution start with kernel with an even number of columns or rows.

An interesting property of convolutions is the *convolution theorem*, which states that $g * h = F^{-1}(F(g) \cdot F(h))$. This property allows to transform the images into the Fourier domain, apply the convolution, transform back and obtain the same results with less computational cost. Below we show some examples of our convolution code, as well as a table comparing the performance of our initial code, our code with the Fourier transform and Matlab's implementation.

| (a) myconv. | (b) myconvFFT. | (c) Matlab's conv2. |

Figure 1: Comparative of the different convolution codes with a horizontal sovel kernel $k = [-1, 1]$.

	myconv	myconvFFT	Matlab's conv2
Time	0.658s	0.04s	0.002s

Table 1: Performance comparative of the convolution methods.

In Figure 1 we see that the convolution theorem holds, and in Table 1 that performing the operations in the Fourier domain and transforming back is more efficient.

CITE FOURIER TRANSFORM AND CONVOLUTION THEOREM

4 Features

Being able to detect features in images that are invariant to rotations, translations, scale and changes in illumination is an useful tool, with many applications such as image matching. SIFT features CITE LOWE PAPER provide a fairly robust method to detect points in an image with the aforementioned restrictions. A broad overview of the algorithm is:

1. **Extrema detection:** maximum and minimum intensity points are detected in a difference-of-Gaussian pyramid built with the input image.

2. **Keypoints refinement:** point centrer in interpolated and weak points are rejected.

3. **Orientation assignment:** orientation and strength of the keypoint are calculated.

4. **Descriptor calculation:** a 128 feature vector is computed for each element.
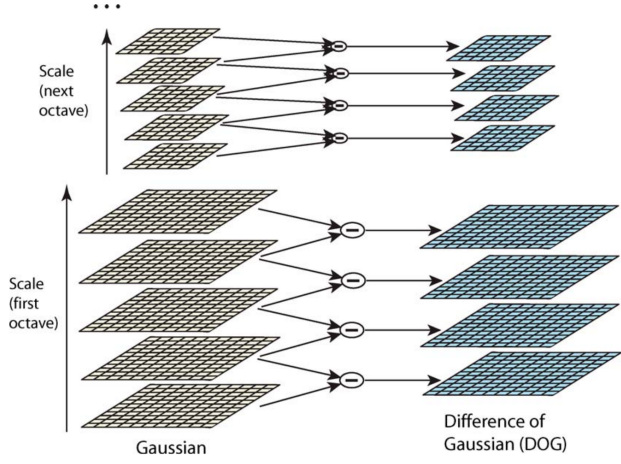
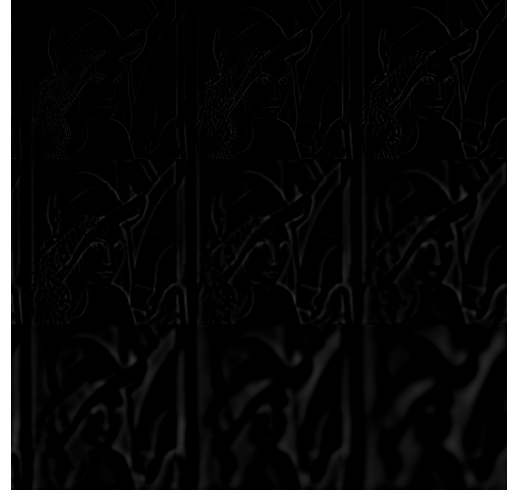Figure 2: Pyramid of difference-of-Gaussian CITE LOWE.

Figure 3: Example of a scale of a difference-of-Gaussian.

To construct the difference-of-Gaussian (DOG) structure, as shown in Figure 2, first a scale of blurred images is constructed. The input image I is convoluted with a Gaussian kernel G:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{1}$$

where L is the output image, $*$ the convolution operator, x and y the pixel indices, and G and Gaussian kernel defined as:

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

where σ is the standard deviation. Next, the DOG structure is computed as a difference of several L:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{3}$$

where D is the output image and k is a constant multiplicative factor. Note that to provide an efficient implementation, the convolution and difference can be computed in the Fourier domain using the convolution theorem explained in Section 3.