Feature detection and matching

11th October 2016

Abstract

This laboratory is devoted to experiment with an implementation of the SIFT descriptor detection and matching.

Contents

1 Introduction

In order to perform this laboratory, the code available at the following link has to be downloaded

`http://www.vlfeat.org/download/vlfeat-0.9.16-bin.tar.gz`

This code is an implementation of the SIFT descriptor detection and matching algorithm described in the work [1] (it is not recommended to download newer versions for Linux). Although it does not correspond to the original author's implementation, the descriptors extracted by this implementation correspond nearly exactly to the ones extracted by the original SIFT.

The code has an API that can be called from MATLAB and C language. For the former case, the library already includes precompiled binaries that can be used under Windows, Linux or Mac.

In order to configure the library, follow the next instructions:

1. Run MATLAB and change directory to the one where the PDF file you are reading resides.

2. In order to configure the `vlfeat` library, run the following command (Avoid copying the text from the PDF file since MATLAB will complain about incorrect characters. You may ignore XML warning messages when running the command):

   ```
   run('vlfeat-0.9.16/toolbox/vl_setup');
   ```

3. To check if installation went right, run

   ```
   vl_version
   ```

2 Feature detection

We are now going to perform some experiments with the SIFT descriptor detection algorithm. For that issue run the following commands (remember! avoid copying the commands from the PDF file since MATLAB will complain; you'll need to introduce them by hand)

```
I = vl_impattern('roofs1');
I = single(rgb2gray(I));
imshow(I);
[f,d] = vl_sift(I);
```

The SIFT keypoints are detected and described by means the `vl_sift` MATLAB command. This command requires a single precision gray scale image. The image range may be in [0,255] or [0,1]. In this example the image is within the range [0,1].

The `vl_sift` command returns the following information:

- The matrix `f` has a column for each keypoint. A keypoint is characterized by a center `f(1:2)`, scale `f(3)` and orientation `f(4)`. The key points are ordered by ascending scale.

- Each column of `d` is the descriptor of the corresponding keypoint in `f`. A descriptor is a 128-dimensional vector of class `uint8`.

We may view the descriptors by using

```
show_keypoints(I,f);
```

Note that a large number of keypoints have been detected in the image. For each keypoint, the application draws a green colored circle indicating its position, orientation and scale at which it was detected. The bigger the scale at which the keypoint was detected the bigger the circle is painted. From an intuitive point of view, the size of the circle is associated to the size of the (circular) patch that is used to compare the patch with the neighboring ones.

One may view only a small subset of the previous keypoints by randomly selecting some of them

```
show_keypoints(I,random_selection(f,50));
```

Question 1*: Take at your own choice several keypoints that have been detected at different scales. Using the theory given in the lectures, comment on the reasons of why do you think that a keypoint has been detected at that position and at that particular scale. You may repeat the experiment with another image (such as 'river1') to understand what a significant keypoints is.*

The `vl_sift` command allows to use some parameters to control how keypoints are detected. The SIFT detector of this lab is controlled mainly by two parameters: the peak threshold and the (non) edge threshold.

- The peak threshold filters peaks of the Difference of Gaussian D scale space (see slides) that are too small in absolute value. That is, for a given keypoint at location p one checks if $|D(p)|$ is above a given threshold.

- The edge threshold eliminates peaks of the Difference of Gaussian scale space whose curvature is too small (such peaks yield badly localized keypoints). Let H be the autocorrelation matrix defined as

$$H = \begin{pmatrix} \sum_{\mathbf{p}\in R} \left(\frac{\partial D}{\partial x}\right)^2 & \sum_{\mathbf{p}\in R} \frac{\partial D}{\partial x}\frac{\partial D}{\partial y} \\ \sum_{\mathbf{p}\in R} \frac{\partial D}{\partial x}\frac{\partial D}{\partial y} & \sum_{\mathbf{p}\in R} \left(\frac{\partial D}{\partial y}\right)^2 \end{pmatrix}$$

  The principals curvatures are estimated as the eigenvalues of the previous matrix. Let r be the ratio between the largest magnitude eigenvalue and the smaller one, so that $\lambda_{max} = r\,\lambda_{min}$. Then

$$\frac{\mathrm{Tr(H)}^2}{\mathrm{Det(H)}} = \frac{(\lambda_{max}+\lambda_{min})^2}{\lambda_{max}\lambda_{min}} = \frac{(r+1)^2}{r}$$

which depends only on the ratio r rather than the individual eigenvalues. The quotient is minimum when the two eigenvalues are equal and it increases with r. Therefore, to check that the ratio of principal curvatures is below a given threshold, r, we only need to check

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

to accept a keypoint. By default a threshold of $r = 10$ is used.

Let us see some examples. First we test the SIFT detector by using different peak threshold

```
[f,d] = vl_sift(I,'PeakThresh', 0.01);
show_keypoints(I,f);
```

Note that the small threshold is due to the fact that the range of gray-values of the input image is within [0,1].

Question 2: *Which is the effect when using the peak threshold=0.01 on the 'roofs1' image? Comment the differences with respect to the previous result (default value of the peak threshold=0).*

Question 3: *Try to slowly increase or decrease the threshold. Comment why the number of detected keypoints decreases when the threshold is increased. Is this the expected behavior according to the way the threshold is defined?*

We now try to experiment with the edge threshold. For that you need to fix the peak threshold, and change the edge threshold for a different value of the default = 10.

```
[f,d] = vl_sift(I,'PeakThresh', 0.04, 'EdgeThresh', 10);
show_keypoints(I,f);
```

Question 4: *Try to slowly increase or decrease the threshold. Comment why the number of detected keypoints decreases when the threshold is decreased. Is this the expected behavior according to the way the threshold is defined?*

3 Practicum submission

You are requested to deliver a PDF document including the experimentation performed during this laboratory and the next one (lab 3 and lab 4). The report should included the answers to the questions proposed in this lab as well as the experimental results for the labs. The PDF document should include all necessary images to fully understand your discussion. There is no need to deliver a long report. I only expect you to answer the questions in such a way that I'm able to see that you have understood what you have been doing during this lab.

References

[1] Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.