2 The SIFT Method

In this session we will test how good the SIFT method is in exstract features from images robustly. First read the article "Object Recognition from Local Scale-Invariant Features" of David Lowe, which can be downloaded from Nestor. In all of the following, try to explain your findings in terms of the properties of SIFT as claimed in this paper.

2.1 Starting the session

For this practical you must first download file "siftprakt.zip" from Nestor to your work directory, and type

```
unzip siftprakt.zip
```

Then type:

```
cd siftDemoV4
```

and start MatLab. You can now compute SIFT features by typing

```
[I,keys,loc] = sift("scene.pgm");
```

The file name must specify a grey scale image (any type supported by MatLab). After this call, `I` contains the image, `keys` contains the key point features, and `loc` contains the location and orientation of the key points. The method also reports the number of key points found.

You can display the key points using

```
showkeys(I,loc);
```

This displays the key points as arrows of different sizes.

Matching can be performed using

```
matches = match('scene.pgm','book.pgm');
```

The number returned is the number of matches found. This is simply the total number, not the number of correct matches. To inspect the correctness of the matches, an image is displayed, in which the two images are shown side by side.

The directory contains a number of images. Files "scene.pgm", "book.pgm", and "basmati.pgm" are from David Lowe, and contain an indoor scene and two images of objects which appear in that scene respectively. Files denoted "bookrXX.pgm" contain the book rotated at X degrees, whereas files denoted "booksXX.pgm" show the book at different levels of shearing.

File "street.png" and "streetlarge.png" contain an outdoor scene at different resolutions. Files "detail*.png" details within these scenes, at higher resolution.

2.2 Excersises

Exercise 1.

Type in the command

```
[I,keys,loc] = sift("scene.pgm");
```

How many key points are found? Check the dimensions of the array keys, using the command

```
whos
```

Does the result match the description in the paper? If not try to explain the difference. Inspect the key points using `showkeys`. Describe qualitatively how they are distributed (include the figure in your report to make dscribing it easier).

Now match book to scene using

```
matches = match('scene.pgm','book.pgm');
```

What is the number of features matched? What is the percentage with respect to the number of key points in each of the images? Are all matches correct? Are the mismatches reasonable (i.e. can you understand why the algorithm matches the unrelated features)? Do the same for "basmati.pgm".

Now move to the outdoor scene image "street.png", and try to match each of the detail images, as above. Compare this to "streetlarge.png", which is considerably larger. How does the number of key points differ between the two different resolutions. Does higher resolution always improve matches? (NOTE: look very carefully, especially in the case of "detail5.png")

Exercise 2.

Now try to match the detail images of the outdoor scene to the indoor scene and vice-versa. What are the differences between the two situations. Do you think you could set a threshold on either number of matches or percentage of matches to robustly claim a match between images? Motivate your answer.

Exercise 3.

Returning with the indoor scene, study how the number of matches varies as you rotate the book, using the pre-rotated images provided. What does this tell you about the rotation invariance of SIFT?

Exercise 4.

Do the same with the sheared images. Plot your results as a function of the number of pixels shear applied (indicated by the number "XX" in the "booksXX.pgm" file names).