Machine Perception COMP3007

Feature Extraction

Exercise 1 - SIFT Feature Extraction

In this exercise, you will investigate the invariace property of SIFT feature descriptors. First, run the SIFT feature detector on a checkerboard image. Examine how many SIFT keypoints that have been identified. Now, using the method `drawKeypoints` to highlight the locations of the keypoints. Observe any special thing about the locations of the keypoints and try to explain the result. What can you comment on the locations of the SIFT keypoints compared to those of the Harris corners that you have found in Practical 3. You should also examine the main orientation of the SIFT keypoints as well.

Next, pass these SIFT keypoints to SIFT feature extractor to obtain SIFT descriptors for these detected keypoints. Examine the number of descriptors and the dimension of each descriptor. Check with the lecture notes on SIFT features and try to explain. Now export the descriptors as an intensity image and carefully observe this image. What can you comment about the descriptors? Are there any special pattern found in the output intensity image.

Repeat the above steps but this time you use a rotated and slightly scaled version of the checkerboard image. Again, examine the SIFT feature descriptors when exported as an intensity image. Can you verify the scale invariance property of SIFT descriptors? Pick a few pairs of SIFT keypoints of the two images and examine their SIFT descriptors in more detail.

Exercise 2 - Binary Shape Analysis

In this exercise, you will extract blob features using binary shape analysis from a gray-scale input image. This task involves separating individual characters and then extract several binary features.

First, convert the input gray-scale image to a binary image using a suitable threshold. You may use the Otsus thresholding method, which is available in OpenCV (see `threshold`). Examine the binary image and make sure that the threshold works as expected.

Next, implement a connected component labeling (CCL) algorithm to separate the blobs (i.e. the characters of this image). Refer to the lecture notes for this type of algorithm.
Once you have successfully implemented CCL, apply it to the binary image and ensure that you can obtain each blob accurately. For each blob, identify the tightest bounding box and extract the corresponding character. Verify that you have found the bounding boxes correctly by extracting individual characters as image files or drawing the bounding boxes on the original image.

Now, for each character, compute the following features:

- Area, i.e. the total number of foreground pixels

- Height
- Width
- Fraction of foreground pixels, i.e. $\frac{\text{Area}}{\text{Height} \times \text{Width}}$
- Distribution of the foreground pixels in the X-direction
- Distribution of the foreground pixels in the Y-direction

Compare the features that you obtain for different pairs of characters. Identify which features that allow you to distinguish characters that are different, and which features that most consistently describe characters that are the same.

Exercise 3 - Histogram Feature Extraction

Write a program that computes the histogram of a given input gray-scale image patch. You need to decide your choice of the number of bins. Then go back to the example in Exercise 2. Using the bounding boxes found for the characters, compute the histogram for each segmented character. Again, compare the histograms of characters that are different, and characters that are the same and comment on the histogram feature. How does it depend on the resolution of the histogram?

End of Practical.