Exercise (4th Day)

(Bag-of-Features)

On the last days, we already had a look at local image descriptors (LBPs and HoGs) and even derived a global histogram representation from LBP features. Today, we will build upon a set of gradient based local image descriptors in order to create a global image representation: the so-called Bag-of-Features representation.

Here, we will use SIFT descriptors – basically a histogram of oriented gradients is computed for $4 \times 4$ cells of a fixed size – that are computed on image patches in order to derive a global representation. Typically, the Bag-of-Features representation is used for object recognition by extracting SIFT descriptors on a dense grid with highly overlapping descriptors.

**Note:** For simplification and runtime purposes our grid will be non-overlapping.

**Where is the detector?** In the lecture you learned that SIFT descriptors are often computed with a specific orientation at points of interest that are determined by a special detector. This method was originally developed to be used in image registration and object detection. For object recognition with the Bag-of-Features approach this step is often skipped and replaced by the dense grid. The dense grid has the advantage that more descriptors are computed which can then describe the object more accurately and also allow for covering quite uniform areas that would be discarded by the detector. It however implicitly assumes that the object is at least to some extent the focus of the image.

**Exercise 1:** Load the image ”/data/geometry.png” and use the `compute_sift_descriptors` method to extract SIFT descriptors on a dense grid.\(^1\) The descriptors are then clustered in order to derive a set of representatives, the so-called visual vocabulary. Each of the descriptors can then be described by its closest representatives, which is basically a quantization. The results shall be visualized by plotting the center points of each descriptor and their surrounding cells. Each center point will be plotted in a color according to the closest representative.

Have a look at finer and coarser grids as well as larger and smaller vocabularies. What do the colors tell you?

**Optional:** Try to replace the given call to k-Means clustering with your own implementation of Lloyd's k-Means algorithm.

**Exercise 2:**

In ”/data/objects/” you will find a set of different objects. Implement an object recognizer that tries to identify the objects in the ”test” directory based on the ones from the ”training” directory. Therefore, you need to compute a Bag-of-Features representation for both training and testing images. The visual vocabulary should be computed based on the SIFT descriptors from all training images. For classification, use a simple Nearest Neighbor classifier.

**Optional:** The Bag-of-Features concept discards all spatial information within an image. A remedy for this is the so-called Spatial Pyramid. It computes a Bag-of-Features representation for sub-regions of an image (i.e. upper left, upper right, lower left, lower right) as well as the complete image and then concatenates the resulting representations for each region. Replace the simple counting without any spatial information by a useful spatial scheme.

**Even more Optional:** If you are familiar with the concepts of Pattern Recognition have a look into the `sklearn` package ([http://scikit-learn.org/](http://scikit-learn.org/)) and train a classifier.

---

\(^1\)For those of you working on a laptop without the non-free opencv modules installed, you can simple load the precomputed descriptors in `ex1_descriptors`.