Leopold Gaube

15.01.2017

# 1 Extracting SIFT Features

## 1.1 Selecting Keypoints from Scale-Space Extrema

The main goal of the SIFT algorithmn is to extract locally destinct points from an image so that similar points can be extracted from different images which depict the same object(s). The pixel location of such a locally destinct point is called keypoint and its surrunding pixel neighbourhood is a feature. It would be a bad idea for an image feature-based algorithmn to consider every possible pixel location, because computation for a single image would take a really long time and most features would be unusable for exact localization due to their lack of image information.

That is why a sampling strategy has to be applied in order to find good keypoints. A keypoint is considered good, if its feature exhibits a high recognition value. Therefore it should be invariant to numerous transformations such as rotation, scale, distortion and brightness but also to change in illumination, 3D viewpoint and addition of noise. Of course it would be really hard to satisfy all those properties, but the Scale-Invariant-Feature-Transform turns out to be quite good, which will further be examined in this paper.

Lowe uses for his keypoint samping approch a so-called scale-space representation of the given image. That means the image will be smoothed with different scales $\sigma$ of the Gaussian blur(/kernel). Smoothing/Blurring (synonym?) an image with a low $\sigma$ value will supress fine structures, whereas smoothing with a large $\sigma$ only leaves coarse/rough contours. That is exactly what we need in order to find distintive features no matter how large or small they are depicted in the image. The resulting images stack-up to form a Gaussian pyramid.

Lowe has found that using a initial $\sigma = 1.6$ and gradually adjusting each ensuing scale of the Gaussian blur by a constant factor of $k = 2^{1/3}$ will achieve the best results [reference paper].

After every three blured images - which also means after each doubling of the $\sigma$ scale - the image can be reduced to a quarter of its previous number of pixels by discarding every second row and column of the previous Gaussian image. This procedure does not affect the accuracy of ?? (because the sampling theorem is not violated (reverense/proof)), but results in major computational performance boosts. All images of the same size in the pyramid are considered form an octave.

Subtracting one Gaussian blurred image pixelwise from another results in a Difference of Gaussians. Doing this for every two neighboured images of the same octave in the Gaussians pyramid gives us a Difference of Gaussians (DoG) pyramid with one image less in each octave then before. In order to compensate for this lost image we previously need to compute one more Gaussian blurred image at the end of each octave. Keep in mind that the resampling for the first image of the next octave needs to be done on the same image as before.

In order to finally retrieve the potential keypoints, each pixel of every Difference of Gaussian (except the first and last of each octave –¿ reason + 2 more Gaussian images needed!!!) is compared to its eight pixel neighbours as well as its 18 neighbours from the DoGs laying directly above and below it in the pyramid (see Figure 2.x). A pixel's location is taken into the set of our potential keypoints only if it's value is either a minimum or a maximum in it's surrounding neighbourhood.

As mentioned before smoothing/blurring an image with different strengths results in images with a different amount of detail. A miniumum or maximum in scale-space means that a structure is still visible in one Gaussian image, but has been "smoothed away" by the stronger Gaussian kernel of the next. This leads to a large pixel value difference along the structure between these two

gaussians. A Difference of Gaussians has therefore a really strong response to edges, as explicated in "Gaussian-based edge-detection methods-a survey" by M. Basu.