Computer Vison Project 3

120210211 Lee Jihwan

- Problem: Analyze computation times of the functions in computeKeypointsAndDescriptors() and explain how it can speed up.

The template_matching_demo.py compute SIFT keypoints and descriptors. There are two couples of keypoint and descriptor. I check the computation times in computeKeypointsAndDescriptors() functions through the time library. The below chart shows about the computation times for the each function.

Function	Keypoint1, descriptor1 time	Keypoint2, descriptor2 time
Generatebase Image	0.004000ms	0.00099ms
GenerateGaussian kernels	0.0ms	0.0ms
GernerateGaussian Images	0.00701ms	0.020005ms
GenerateDoGImages	0.001991ms	0.005001ms
FindScaleSpaceExtrema	19.56640 ms	37.20035ms
RemoveDuplicateKeypoints	0.003000 ms	0.006012ms
ConvertKeypointsToInputImageSize	0.003000 ms	0.004001ms
GenerateDescriptors	37.80250ms	60.35175ms

The total time is 154.984 ms. In the total computational time, FindScaleSpaceExtremea and GenerateDescriptors occupy the most of the time. The findScaleSpaceExtrema() function performs to identify keypoints. The findScaleSpaceExtrema() function's parameter is gaussian_images, dog_images,num_intervals,sigma,image_border_width, and contrast_threshold.

This step takes long time because it finds the extremum through the loop and the list array has the O(N) time complexity. And for each extremum, it has to localize its position at the subpixel level along all three dimensions (width,height, and scale) using the localizeExtremumViaQuadraticFit() function. So how can we improve the computation time? The first, I try to reduce num_attempts_until_convergence 5 to 2. So the execute time reduces 154.984ms to 138.8856ms. Then I expand the contrast threshold 0.04ms to 0.08ms. This variable decides about the pixel extremum. So the execute time considerably reduces 138.8856 ms to 93.4814 ms.

And I try to change the generateDescriptors parameter. generateDescriptors() function implements to generate descriptors for each keypoint. I try to reduce the window size 4 to 3, which decides about the descriptor window size. The result shows that the time reduce 93.4814 to 75.1077 ms.

And look at the computeKeypointsWithOrientations, there are radius_factor parameter. The radius_factor default value is 3, which means the neighborhood will cover pixels within 3*scale of each point, this scale is the standard deviation associated with the Gaussian Weighting. So I change this value 3 to 1, so the execute time reduces 75.1077 to 65.4245 ms.