Bauhaus-Universität Weimar

Exercise 5

Topic: SIFT – Scale Invariant Feature Transform

SIFT is a state-of-the-art approach to describe and identify corresponding points in images from the same real world objects from different viewpoints. We have a Matlab code which performs the computation of SIFT features for two input images as well as the automated matching of these points. Unfortunately, the core routine `ComputeDescriptor` for computing the SIFT feature descriptors for each identified point was erased. Your task: Re-implement this function! The main steps of this function are (See implementation details on the slides for complete description):

Inputs: σ for each scale, gradient magnitudes and orientations for each scale, keypoints
Outputs: Keypoint descriptor array

- Compute a 2D Gaussian filter G of size 49×49 px with $\sigma = 8$
- **For** all existing scales
  - **For** all identified keypoints in current scale
    - Initialize the 16×16 px arrays x_c and y_c with local coordinates
    - Rotate x_c and y_c according the main orientation θ_m of the current point
    - Determine the nearest scale σ_n of the keypoint
    - Sample the Gaussian filter, magnitude and angle images at scale σ_n
      ➔ Result: sampled 16×16 arrays G_s, M_s and θ_s
    - Compute weighted gradient magnitudes (16×16): $M_w = G_s .* M_s$
    - Rotate magnitude angles: $\theta_{s,rot} = \theta_s - \theta_m$
    - Build 16 histograms (one for each (4×4) subcell) with 8 bins (binsize = $360/8 = 45°$)
    - Build a descriptor vector by rearranging histogram values to 128 element vector
    - Normalize, crop and again normalize this vector
    - Store it in a $n_p \times 128$ array, where n_p is the overall number of keypoints
  - **End**
- **End**

Validate your implementation using the images (*Input1.jpg* and *Input2.jpg*).

Matched Points will be connected with a red line in the result plot. So if a lot of these lines intersect in one point you did a good job!

Dr. Jens Kersten
jens.kersten@uni-weimar.de