Exercise 11 — Image Processing

Michael Bleier, Andreas Nüchter
Computer Science XVII: Robotics

Summer 2023

Note: This is the last exercise sheet of this semester!

Goals: In this assignment you will perform basic image processing tasks on satellite images.

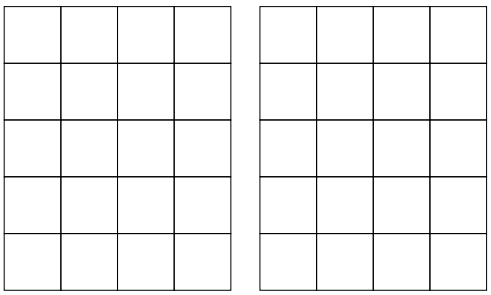
Part 1: Image gradients (*6 points*)

1. Calculate the image gradients (of the 6 pixels in the center) in x und y direction by considering only the 4-neighborhood, i.e., the pixels above, below, left and right of the pixel in question. To this end apply the two filters to the image:
   - $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$
   - $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

1	1	2	2
1	1	3	2
1	1	3	2
1	1	3	2
1	1	3	2

2. Calculate the length of the gradient vector for each pixel and draw the resulting gradient vectors as arrows into the image.

3. Consider all pixels with a gradient length ≥ 1 as edge candidates and mark them in the image. Explain how non-maxima suppression is used to thin out the edges and mark the final edge pixels. You may use nearest neighbor interpolation to determine the value in gradient direction.



Part 2: SIFT features (*9 points*)

In this task we want to compute correspondences between images using SIFT features.

The following links will help you to get started:

- `https://docs.opencv.org/4.5.4/d7/d60/classcv_1_1SIFT.html`
- `https://docs.opencv.org/4.5.4/db/d39/classcv_1_1DescriptorMatcher.html`
- `https://docs.opencv.org/4.5.4/d7/d66/tutorial_feature_detection.html`
- `https://docs.opencv.org/4.5.4/d5/dde/tutorial_feature_description.html`

(If you use an older version of OpenCV, you will need `xfeatures2d` from `opencv_contrib` for SIFT.
`https://docs.opencv.org/3.4.9/d5/d3c/classcv_1_1xfeatures2d_1_1SIFT.html`
If you did not include that in your installation, you can either install it now, or try one of the other features provided in OpenCV. Clearly state in your submission, if you used a different feature type.)

Todo list:

1. Download two satellite images of the Earth, that show the same area at different times. Images from the Landsat and Sentinel missions are available for free. For sources on Landsat data you may refer to (`https://landsat.gsfc.nasa.gov/data/where-to-get-data/`). Most tools listed there also provide access to Sentinel data or you can access it via the Copernicus Open Access Hub (`https://scihub.copernicus.eu/dhus`). Alternatively, you may find suitable images in the Landsat Image Gallery (`https://landsat.visibleearth.nasa.gov/`).
2. Load the two images as grayscale images.
3. Create a SIFT feature detector and detect keypoints in both images.
4. Create a SIFT descriptor for both images and compute the feature description of the keypoints.
5. Create a DescriptorMatcher and compute correspondences between the features of both images.
6. Try different parameters to find the best for your images.
7. Generate an image that shows both images with the detected features and matches. Show only the best k matches. Choose k such that the images and feature matches are still recognizable.
8. Briefly describe your results.

Submission: Please submit your solutions until **Thursday, 20 July 2023, 10:15**. You may work in groups of up to three. Please note the full names of you and all your exercise partners on the submission. Only one submission per group is necessary. Make sure to provide your code with sufficient comments. Create a PDF-Document containing the images and the written answers/statements. Add your source code to a zip-file and upload the two files to WueCampus.

WueCampus course: `https://wuecampus.uni-wuerzburg.de/moodle/course/view.php?id=59606`