Lab 4 - Keypoints

The keypoints in an image are pixels that are more “important” than other pixels, their neighborhood contains more information. These points in an image should not depend on illumination, scale or geometrical, affine distortions (rotations, translations, …). There are many methods that compute keypoints, such that: Harris corner detection, SURF, SIFT, ORB, BRISK, AKAZE, and so on. Using keypoint extraction was applied in solving problems like human action recognition, human pose estimation, object detection and recognition, panorama stitching, video tracking, and this list can continue. Keypoints are employed in the following way: usually in the same time with keypoints computations, descriptors with information around these keypoints are also calculated. These descriptors are feature vectors of fixed length (64 or 128) with information on orientations, magnitude of gradient in a neighborhood around the keypoint. When comparing two images, a keypoint matching procedure is performed to identify the common parts.

A keypoint approach for solving a given task has the following steps:

1. Covert to grayscale (if necessary);
2. Keypoint detection;
3. Descriptors computation;
4. (Keypoints, descriptors) matching;
5. Analyze the results provided by the previous steps in the context of the problem to be solved.

Problems to be approached for this lab:

1. Compute keypoints for an image using different methods (Harris corner detection, SIFT, SURF, …). Use as many keypoints detection methods as the OpenCV library allows (at least 3).
    a. Plot the image with marked keypoints.
    b. Compare the keypoints detection methods analyzing the number of computed keypoints and their positioning in the image. For the positioning comparisons, divide the image in 8 × 8 non-overlapping regions, and count the number of keypoints in each region.
    c. Blur the image. Compute its keypoints. Match the keypoints of the original image and the blurred one using brute force and the kNN based methods. Monitor the number of matched keypoints. Use different types of blurring (with mean filters, with Gaussian filters of different size).
    d. Rotate the original image and then perform the same computations/comparisons as those described in step c.

e. Add Gaussian noise to the original image. Perform the same analysis as that described in step c. Use different amount of noise (different values for the standard deviation, Add different noise to an image | TheAILearner).

2. Consider the following dataset (link here[1]). It contains images from three locations in Iaşi.

   a. For all the images in this dataset compute the keypoints and the corresponding descriptors, using different methods for keypoints detection.
   b. For each image in the test set perform the matching process with all the images from the train set. Use both matching procedures implemented in OpenCV.
   c. Compute the set of images from the training set that have maximum number of matched keypoints with the test image.
   d. If the set computed in step c. has only one image, that image provides the label for the test image. If the set contains more than one image, perform a voting procedure to assign a label (location) to the test image. In case of parity, assign the label randomly or find another way to assign the label. Display/print the name of the image for which the parity situation occurred.
   e. Compute the accuracy of the classification process and the confusion matrix.

   Remarks: 1) Test different keypoints extracting methods and different matching procedures.

   2) If, for an image, the keypoints detection method does not compute any keypoint, modify some parameters of this method until it computes at least one keypoint. If for certain keypoints methods this type of computation isn't possible, display a message that states this fact, and continue with the computations. A test image with no keypoints cannot be classified (or assign a random label) or it cannot be used in the classification process if the image belongs to the training set.

Useful links:

https://www.cs.utah.edu/~srikumar/cv_spring2017_files/Keypoints&Descriptors.pdf

https://www.mdpi.com/2223-7747/10/9/1791/pdf

OpenCV:

https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html

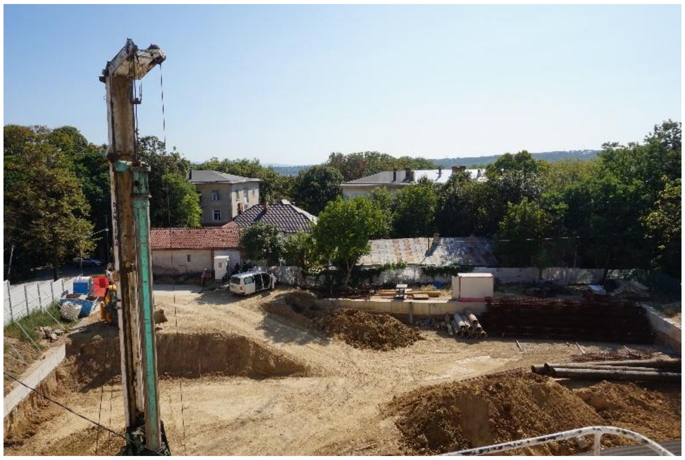
[1] The images from this archive are part of a larger dataset created by the researchers of the Institute of Computer Science, Romanian Academy, Iaşi branch.

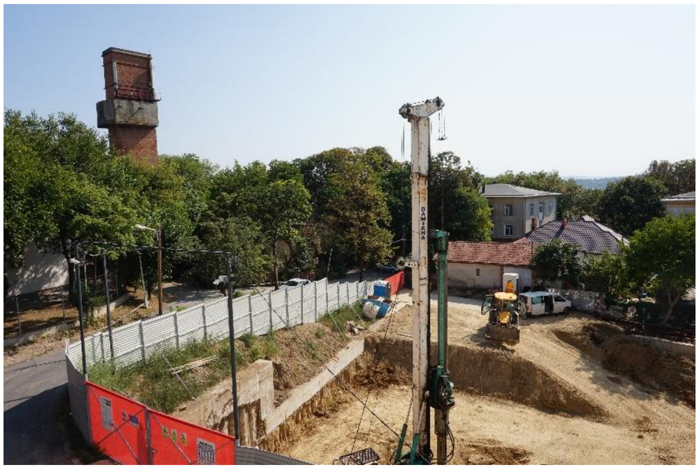
https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html

https://machinelearningmastery.com/opencv_sift_surf_orb_keypoints/

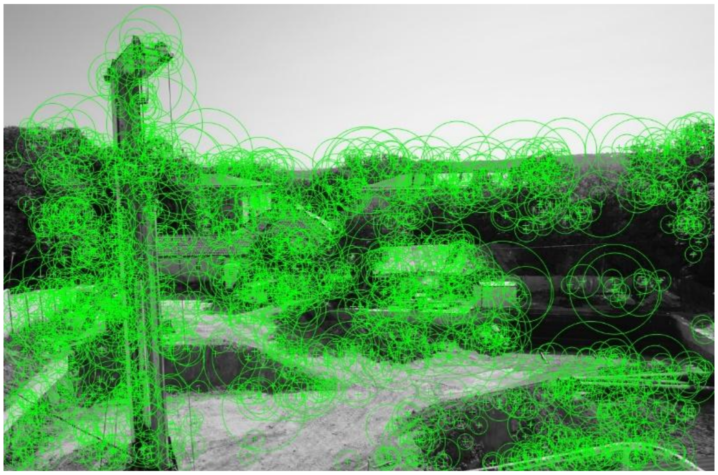
https://learnopencv.com/object-keypoint-similarity/

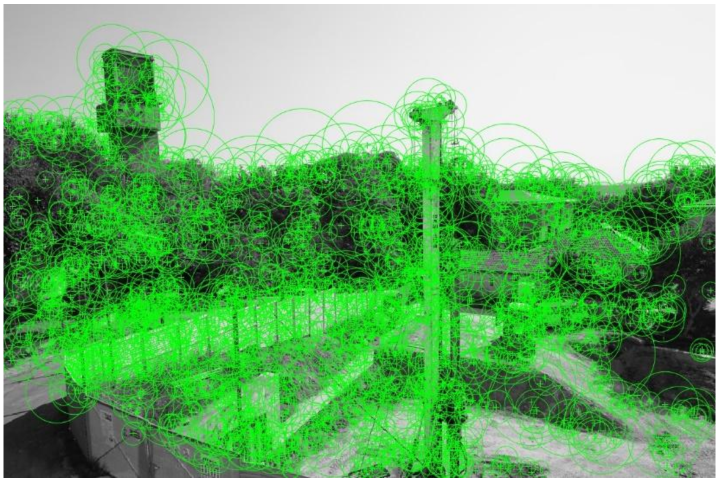
Original images

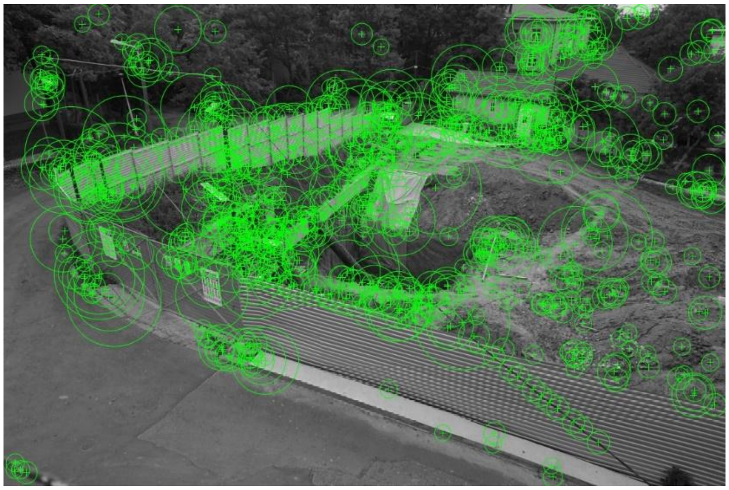




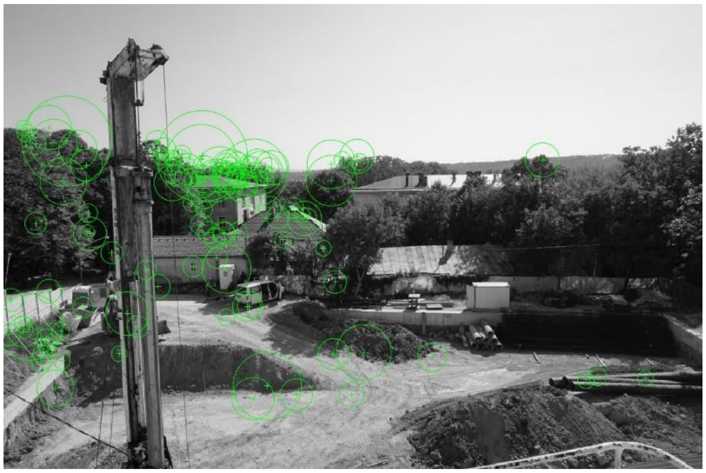
Images with SURF keypoints

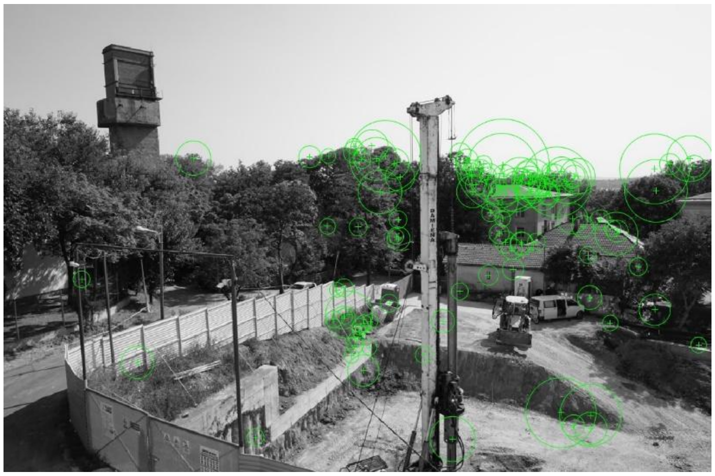


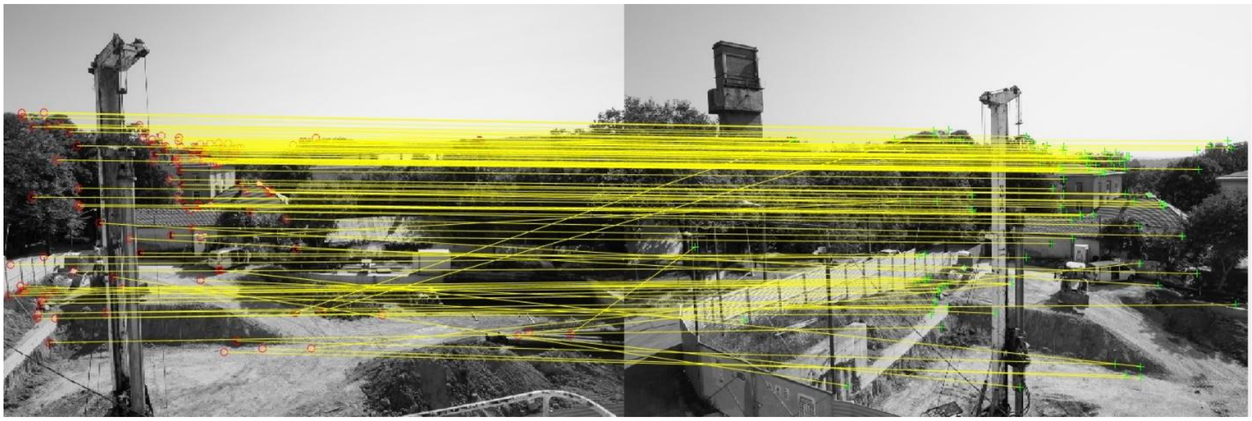




Images with matched keypoints







5 matched keypoints