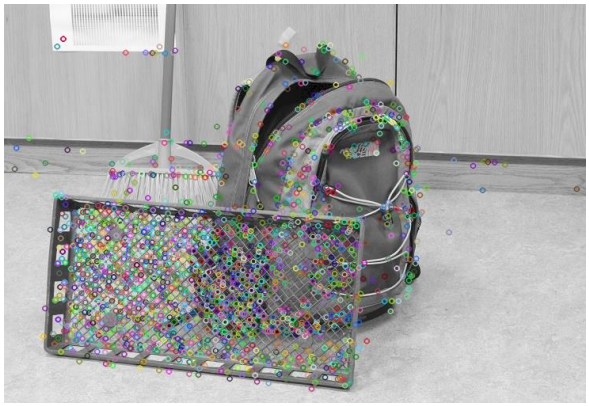


ELEC 474

Prelab 4 – Feature Detection

SIFT Feature Detection

Kevin Huang and Michael Greenspan

Contents

1. Scale-Invariant Feature Transform (SIFT)

For this prelab you will make use of OpenCV's implementation of Scale-Invariant Feature Transform (SIFT) algorithm. SIFT is used in a variety Computer Vision of applications, including object recognition, robotic mapping, 3D modelling, and gesture recognition.

To use OpenCV SIFT you need to install the external ***opencv_contrib*** library, as SIFT is a patented algorithm not included in OpenCV's standard codebase. You can install it by typing the following command into the Anaconda Command Prompt:

Default Terminal (e.g. CMD.exe):

```
python3 -m pip install opencv-contrib-python
```

Python Terminal:

```
pip install opencv-contrib-python
```

Once installed, you can use the SIFT algorithm in Python.

```
# Initialize SIFT Module
my_SIFT_instance = cv2.SIFT_create()
```

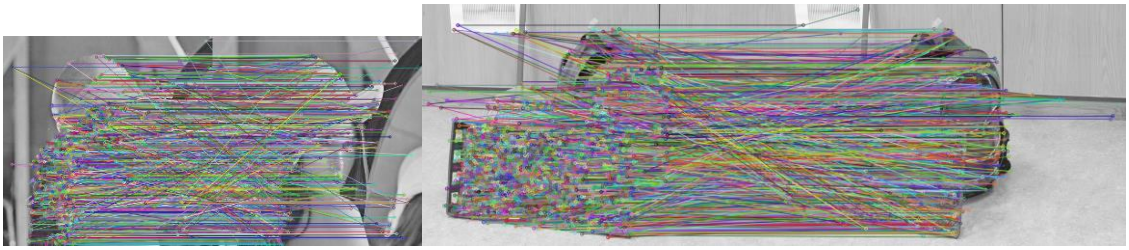
1.1 SIFT Matching

For this portion of the prelab you will use SIFT to match features between a pair of identical images and a pair of stereo images. For each pair of images, follow these steps for SIFT Matching:

1) Load the two images for matching.
2) Initialize the SIFT module in Python using **my_SIFT_instance = cv2.SIFT_create()**.
3) For each image, call **my_SIFT_instance.detectAndCompute(…)** to generate the image's **keypoints** and **descriptors** for possible matches.
4) Draw your detected **keypoints** for your images using **cv2.drawKeypoints(…)** function.

The detected key points displayed on "lena.png", "backpack_left.png", and "backpack_right.png" key points images are:

5) Using any **cv2** matcher (**cv2.BFMatcher()** is used in this example), find matches with your **descriptors**. When computing, set your matcher output (**k=2**) to output **two** possible matches (important for next part of lab).
6) Display all of your matches across the images using **cv2.drawMatchesKnn(…)**. The matches for "lena.png" (same comparison) and the backpack stereo images are:



1.2 Lowe's Ratio Test

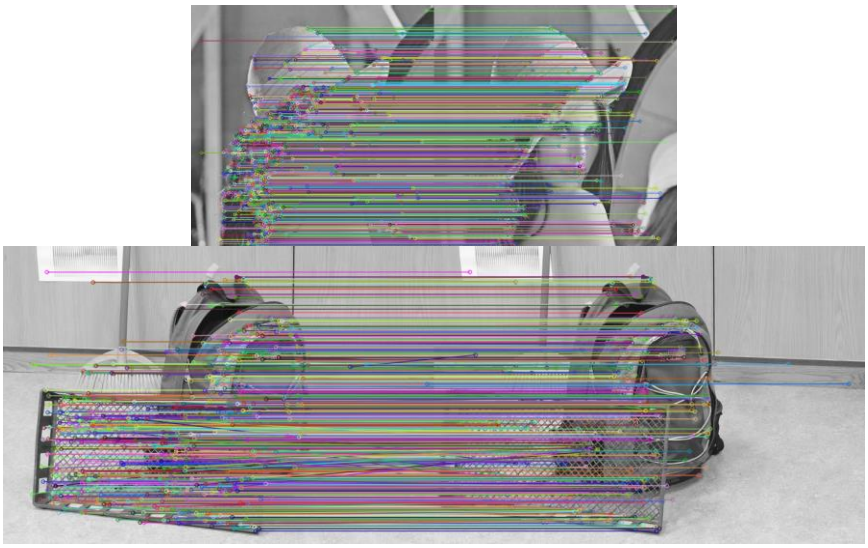
Some of the matches drawn in the previous figure are correct, and some are incorrect. Non-horizontal matches in the Lena and stereo backpack image pairs are incorrect as the images differ purely or almost purely in horizontal translation.

To filter keypoint matches, David Lowe proposed a simple method using Euclidean distances to eliminate matches when the **second-best** match is found to be nearly as good as the best. Working on the assumption that a keypoint cannot have more than one match in an image pair, Lowe's Ratio Test checks if the two distances **(in descriptor space)** are **sufficiently different.**

For keypoint K1, let D1 be the distance to the best (i.e. closest) match, and let D2 be the distance to the next best match. The ratio test looks at the ratio of these two values, and passes the best match as being "good" (i.e. distinct) if this ratio is less than some threshold, i.e. $D1/D2 < \tau$. (In the original paper, Lowe used a value of $\tau=0.8$.)

To implement Lowe's Ratio Test follow these steps:

1) Loop through each of your matches **(keypoint pairs)**.
2) If the Euclidian distance of "good" keypoint ***m*** is **sufficiently different (by some threshold ratio)** to the Euclidian distance of the second-best key point ***n*** then the keypoint is accepted as a ***good*** match.
3) Display Lowe's Ratio Test filtered matches for Lena and stereo backpack image pairs:

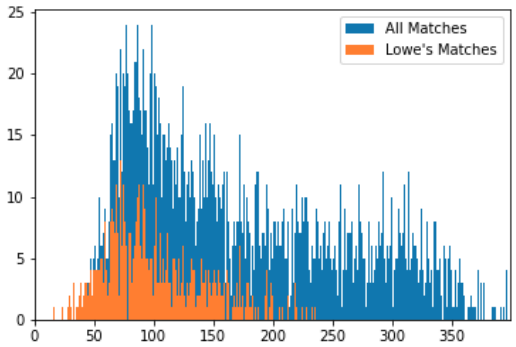


You can see that Lowe's Ratio Test eliminated a number of invalid matches from the initially detected set.

1.3 Histogram

For this portion of the prelab, display your keypoint distances as two histograms, one for **each of the Lena and backpack stereo images.** These histograms should illustrate all matches, distinguishing by color the difference in before and after the application of Lowe's Ratio Test.

For example, the histogram for the backpack stereo image pair follows. Here the x-axis is the distance between the two descriptors (in descriptor space), and the y-axis is the count of the histograms with that distance. All matches are displayed in blue, and only those that pass the ratio test are displayed in gold.



Q1/ What conclusion can be drawn from this histogram?

2. Submission

The submission for this prelab should include a .zip with:

- .ipynb file that includes:
  - Your code for SIFT Feature Detection with Lowe's Ratio Test.
  - Visualization your code's results for "lena.png", "backpack_left.png" and "backpack_right.png" stereo image pairs, including:
    - Each image's keypoints
    - Unfiltered Keypoint matches for each image pair
    - Filtered keypoint matches for each image pair
    - A Histogram illustration of filtered and unfiltered keypoint matches for each image pair
  - A markdown cell with your answer to Q1/
    - For Jupyter Notebook markup see: https://gtribello.github.io/mathNET/assets/notebook-writing.html
- Any necessary supporting files/images

Your code will be run in Jupyter Lab to test for functionality.

	Item	mark
1.	SIFT Matching correctly implemented and displayed	0.5
2.	Lowe's Ratio Test correctly implemented and displayed	0.5
3.	Histogram correct	1
4.	Q1 correctly answered	0.5
5.	Submission format correct	0.5
	Total:	3