# CSC262: Feature Matching

## Table of Contents

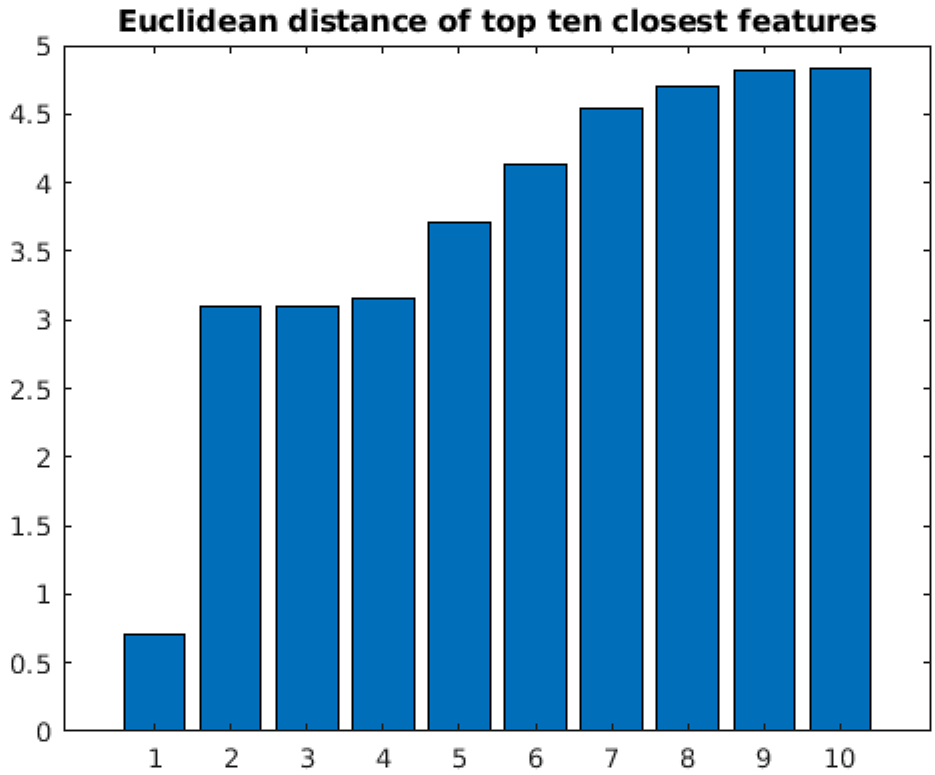


# Overview

In this lab, we will explore the ways in which we could stitch two images taken at a slightly different location (assuming only a translational transformation) by detecting features and matching them together. In this process, we will calculate the Euclidean distances between all feature descriptors, sort them from shortest to longest, then threshold out certain features which have ambiguous neighbor features. Specifically, this lab asks us to compare the nearest-neighbor distance ratio to our own threshold. The purpose of this lab is to learn how to manipulate group of features detected across multiple images to produce the laIn this lab, we will explore the ways in which we could stitch two images taken at a slightly different location (assuming only a translational transformation) by detecting features and matching them together. In this process, we will calculate the Euclidean distances between all feature descriptors, sort them from shortest to longest, then threshold out certain features which have ambiguous neighbor features. Specifically, this lab asks us to compare the nearest-neighbor distance ratio to our own threshold. The purpose of this lab is to learn how to manipulate group of features detected across multiple images to produce the larger frame of the world by diving into a simple panorama technique. rger frame of the world by diving into a simple panorama technique.

# Feature Matching

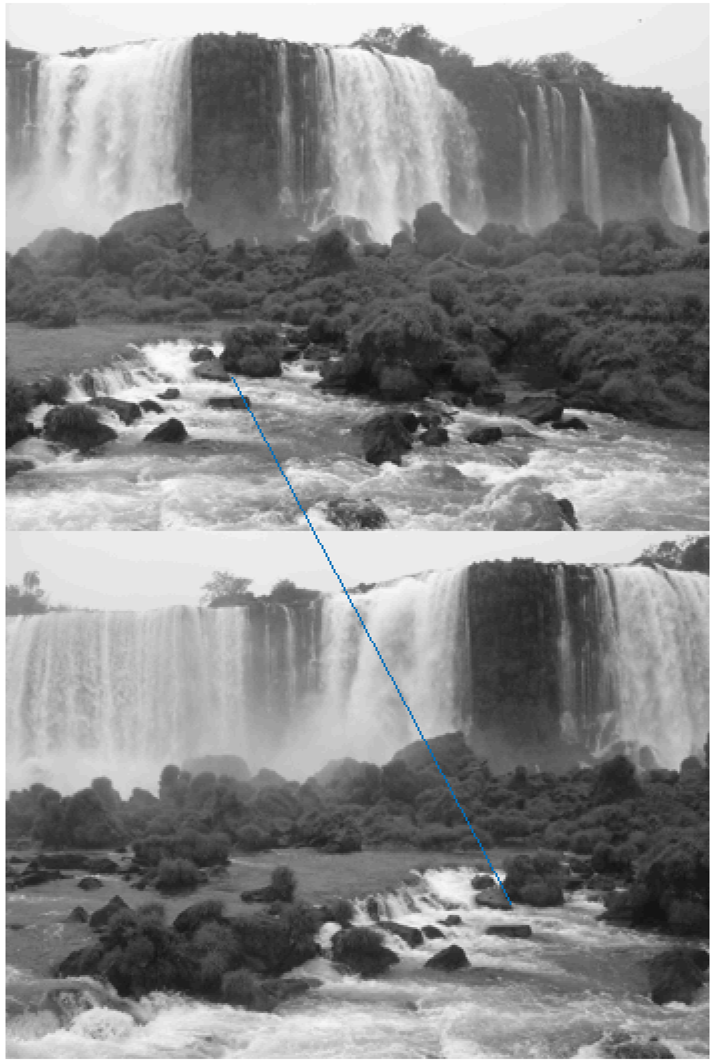
In this section, we chose one of the two images loaded as the reference image. We chose one feature from this reference image, and look for it's match in the other image. We attempt this by calculate and compare the Euclidean distances between our chosen feature and every features in the other image.




# Visualizing Matching

We create a visualization of the chosen feature that we match in the two images. We draw a line from our chosen feature in the first image to it's match that we found in the second image. This visualization, displayed below, indicated that we performed our calculation correctly, as the two points are indeed of the same object in the images.

Visualization of feature matching



# Alignment

In this final section, we calculate a single optimal global alignment by finding the least squared solution to the transformation problem. Then, we utilize this transformation to stitch the two images, creating a single rudimentary panoramic image. We display this image below.

Stiched image from the optimal translation

At first viewing of the image above, the stitched result does appear to merge the two images well. The overlaping waterfall and the rock formation below it are matched by our transformation. However, we noticed that the transformation does not perform well in the ege of the overlaping area. For example, along the edge of the waterfall and the background, we can clearly see mismatched region on the left and right side. The middle of this waterfall edge does appear to be much more well matched. Other clear artifacts in the stiched image are the lines dividing the regions of the overlaping field of view. Along these lines, we can clearly recognize the discontinuity of the stitch. We theorize that these visible lines could be a direct result of the two images having different level of illumination. Another source of discontinuity is obviously the continous motion of the waterfall. Since the water cannot remain constant between the time interval the two images are taken, there must be discontinuity in the stiched image. By this same logic, we recognized that the rocky formation is matched well by our transformation, since they remain constant between the time the images are recorded.

# Conclusion

In this lab, we utilized kpdet.m and kpfeat.m functions to detect features and derive their feature descriptors. Then, we processed the data so that we filter out the invalid features and set them as NaN. We then computed the Euclidean distance between one feature and all the other features of the other image. After we sorted the distances, we used the nearest-neighbor ratio to filter out features which inadequately represent a single point on the other image. This means that there might be an incorrect point on the other image that has a very similar characteristics to the one feature. Then, we calculated the translation estimate for all the features in our reference image. We put this into a loop, then computed the same for all features. At the end, we produced a panorama image that stitched the two images together.

# Acknowledgement

We, as partners, were the only contributors to this lab. We referred to Piazza to figure out if the translation estimate in rows and columns should be rounded. The kpdet and kpfeat functions used in this lab was built in the previous lab by a group member. We also utilized directions, information, and code snippets from the Feature Matching Lab text written by Jerod Weinman for CSC 262: Computer Vision this semester. Finally, we also consulted lab write up guidelines published on the course website by Jerod Weinman.

*Published with MATLAB® R2018b*