EECS 442: Computer Vision, Winter 2016

Homework 4: Image stitching

Due date: March 7 by 11:59PM

The goal of this assignment is to use feature-based homography estimation to stitch together panorama images.

Part 1: Stitching pairs of images

The first step is to write code to stitch together a single pair of images. For this part, you will be working with the following pair (found in the data folder as uttower_left.jpg and uttower_right.jpg):

1. Load both images, convert to double and to grayscale.

2. Detect feature points in both images. You can use the corner detector code provided (`harris.m`) or the blob detector you have developed as part of the last homework.

3. Extract local neighborhoods around every keypoint in both images, and form descriptors simply by "flattening" the pixel values in each neighborhood to one-dimensional vectors. Experiment with different neighborhood sizes to see which one works the best. If you're using your Laplacian detector, use the detected feature scales to define the neighborhood scales.

4. Compute distances between every descriptor in one image and every descriptor in the other image. You can use `dist2.m` for fast computation of Euclidean distance. Alternatively, experiment with computing normalized correlation, or Euclidean distance after normalizing all descriptors to have zero mean and unit standard deviation. Feel free to experiment with SIFT descriptors. We have provided basic code (`find_sift.m`) for computing SIFT descriptors of circular regions, such as the ones returned by your blob detector.

5. Select putative matches based on the matrix of pairwise descriptor distances obtained above. You can select all pairs whose descriptor distances are below a specified threshold, or select the top few hundred descriptor pairs with the smallest pairwise distances.

6. Run RANSAC to estimate a homography mapping one image onto the other. Report the number of

inliers and the average residual for the inliers (squared distance between the point coordinates in one image and the transformed coordinates of the matching point in the other image). Also, display the locations of inlier matches in both images.

7. Warp one image onto the other using the estimated transformation. To do this, you will need to learn about `maketform` and `imtransform` functions.

8. Create a new image big enough to hold the panorama and composite the two images into it. You can composite by simply averaging the pixel values where the two images overlap. Your result might look something like this:

For estimating the transformation it is sufficient just to use grayscale images. You can apply the same compositing step to the color images to get a final result.

Tips and Details

- Refer to the class lecture slides on fitting.

- For RANSAC, a very simple implementation is sufficient. Use four matches to initialize the homography in each iteration. You should output a single transformation that gets the most inliers in the course of all the iterations. For the various RANSAC parameters (number of iterations, inlier threshold), play around with a few "reasonable" values and pick the ones that work best. For randomly sampling matches, you can use the `randperm` or `randsample` functions.

- In MATLAB, the solution to a nonhomogeneous linear least squares system `AX=B` is given by `X = A\B;`

- Homography fitting calls for homogeneous least squares. The solution to the homogeneous least squares system AX=0 is obtained from the SVD of A by the singular vector corresponding to the smallest singular value:
`[U,S,V]=svd(A); X = V(:,end);`

Part 2: Stitching multiple images

Now that you have code for pairwise stitching working, the next step is to extend it to work on multiple images. You will be working with the provided data, consisting of three sequences of three images each. For the "pier" sequence, sample output can look as follows (although yours may be different if you choose a different order of transformations):

Source of data and results: Arun Mallya

For this part, you should experiment with and discover your own strategy for multi-image stitching. For full credit, the order of merging the images should be determined automatically. The basic idea is to first run RANSAC between every pair of images to determine the number of inliers to each transformation, use this information to determine which pair of images should be merged first (and of these, which one should be the "source" and which the "destination"), merge this pair, and proceed recursively.

For extra credit

- Experiment with registering very "difficult" image pairs or sequences -- for instance, try to find a modern and a historical view of the same location to mimic the kinds of composites found here. Or try to find two views of the same location taken at different times of day, different times of year, etc. Another idea is to try to register images with a lot of repetition, or images separated by an extreme transformation (large rotation, scaling, etc.). To make stitching work for such challenging situations, you may need to experiment with alternative feature detectors and/or descriptors, as well as feature space outlier rejection techniques such as Lowe's ratio test.

- Try to implement a more complete version of a system for "Recognizing panoramas" -- i.e., a system that can take as input a "pile" of input images (including possible outliers), figure out the subsets that should be stitched together, and then stitch them together. As data for this, either use images you take yourself or combine all the provided input images into one folder (plus, feel free to add outlier images that do not match any of the provided ones).

- Implement bundle adjustment or global nonlinear optimization to simultaneously refine transformation parameters between all pairs of images.

- Learn about and experiment with image blending techniques and panorama mapping techniques (cylindrical or spherical).

Grading checklist

In the report please include the following:

1. Pairwise stitching:
    a. Describe your solution, including any interesting parameters or implementation choices for feature extraction, putative matching, RANSAC, etc.
    b. For the `uttower` pair provided, report the number of homography inliers and the average residual for the inliers (squared distance between the point coordinates in one image and the transformed coordinates of the matching point in the other image). Also, display the locations of inlier matches in both images.
    c. Display the final result of your stitching.
2. Multiple image stitching:
    a. Describe your solution (outline of algorithm, interesting parameter choices, etc.)
    b. For each input sequence, list the order of image stitching output by your algorithm and show your final output image. Discuss any issues or artifacts that may be evident in your output. If your program did not succeed on some of the sequences, discuss why.

We have provided some initial files to serve as a rough skeleton, but you are free to reorganize the code as you see fit. Please attach all files necessary to run your code in your submission.

Acknowledgements

This homework was developed by Lana Lazebnik at UIUC.