Assignment 3: Image Alignment and Stitching

November 21, 2012

1 Image Alignment

To align two images, we use the standard procedure involving:

1. Find keypoints in both images
2. Compute descriptors for each keypoint
3. Compute matches between keypoints
4. Estimate the Affine transformation using RANSAC

In the previous assignment we have looked at keypoint detection, descriptors and matching. As requested in the current assignment, we have used the `vl_feat` SIFT implementation for these three stages. This procedure is straightforward, and can be found in `imageAlign.m`, where we use `vl_sift` to detect and describe keypoints, and `vl_ubcmatch` to do the matching. The result is a set of point correspondences, which we pass to our `ransacA.m` function, where we perform robust fitting of an affine transformation.

Next, we perform RANSAC with an affine transformation model. An affine transformation in 2D has 6 parameters, and since each 2D-point correspondence gives us two equations, we need 3 point correspondences. We simply perform the RANSAC algorithm as described in the excercise, and to estimate the transformation we use a simple linear least squares (using the pseudo inverse, as described in the assignment).

To see the result, simply run `AlignmentDemo.m`, or see figure REF. The first image that is plotted shows the two images with the keypoints and the corresponding transformed keypoint linked by a line. Note that the keypoints in the right image are *not* the keypoints found in that image. The second image that is plotted shows the two images, and the transformation of image one to image two, and vice-versa.

1.1 The number of RANSAC iterations

In our basic RANSAC implementation, convergence is usually very fast. For the tram image, we found that only about 8 out of 67 point correspondences appear to be outliers. This means that the probability of picking three inliers is $(59/67)^3 \approx 0.68$. For this reason, the algorithm often finds the correct transformation in the first few iterations on this image.

There is an easy way to estimate the number of iterations that have to be performed to be reasonably sure to have found the true transformation. Let q be the probability of sampling three inliers. If we perform h iterations, the probability that none of them will be 'good' (i.e. consist of three inliers) is given by $(1-q)^h$. We want to choose h so that this quantity is below some acceptable level: $(1-q)^h \leq \epsilon$. If we invert this inequality, we find

$$h \geq \left(\frac{\log}{\log 1 - q} \right) \tag{1}$$

However, since we do not know q, we cannot use this relation directly. The probability of sampling a set of $k = 3$ inliers can be expressed in terms of the number of matches N and the number of inliers N_I:

$$q = \frac{\binom{N_I}{k}}{\binom{N}{k}} \approx \left(\frac{N_I}{N} \right)^k . \tag{2}$$

where the approximate equality holds when $N_I, N \gg k$ (which will almost certainly hold for our case $k = 3$. We still cannot use this formula directly because we do not know N_I. However, the number of inliers of the current best model is a conservative estimate of N_I, so we can use it instead of N_I. We have implemented this method to determine the number of iterations h.

2 Image Stitching

We can use the estimated affine transformation to stitch a series of images.

3 Homography

As a bonus, we implemented homography estimation and used it to stitch two images, as in the previous assignment. We solve for the optimal projective transformation using the SVD.