Exercise in Geometry Constrained Feature Matching

Henrik Aanæs, haa@imm.dtu.dk

April 2, 2013

In this exercise the curriculum on feature tracking and view geometry are combined, in that you should enhance the feature matcher from "Exercise in Feature Matching", with robust fundamental matrix estimation, which should hereby make for an improved feature matcher.

The images you are expected to work with are:
`House1.bmp`
`House2.bmp`

And later the images in `TwoImageData.mat`, which should be located together with this text.

1 The Eight Point Algorithm

Write a function `Fest_8point` which implements the eight point algorithm for fundamental matrix estimation. This should be a slight modification of the functionality for homography estimation in "Exercise in Estimating View Geometry". As for the homography estimation the data point should also be normalized.

Data to validate your approach on is given in `Qtest.mat`. This file should be located together with this text, and contains two matrices `q1` and `q2`, of corresponding 2D points, which are consistent with the fundamental matrix, of the geometry in `TwoImageData.mat`. This fundamental matrix was computed in "Exercise in the Camera Model", denote it `Ftrue`.

It is recommended that you take the following steps.

1. Confirm that `q1` and `q2` are consistent with `Ftrue`.
2. From `q1` and `q2` construct the linear equations constraining the fundamental matrix, and confirm that they correspond to the elements of `Ftrue`.
3. Set up the functionality to estimate a fundamental matrix from these linear equations. Confirm that this fundamental matrix is equal to `Ftrue` – up to numerical error.
4. Normalize the data, as described in "Exercise in Estimating View Geometry". The estimated fundamental matrix estimated from these normalized points should be modified in the following manner to correspond to the original points:
   `F=T2'*F*T1;`
5. Confirm that this estimate is equal to `Ftrue` – up to numerical error.
6. Wrap the code in a function, e.g. named `function F=Fest_8point(q1,q2).`

2 Feature Matching

Repeat part of the exercise "Exercise in Feature Matching", by matching the two images from `TwoImageData`
I.e. load the appropriate images and initialize `vl_setup`, and execute the code:

```
[fa, da] = vl_sift(single(im1));
[fb, db] = vl_sift(single(im2));
[matches, scores] = vl_ubcmatch(da, db);
nMatch=size(matches,2);
```

Illustrate the result and confirm that it looks reasonable compared to your expectations.

3 F Estimation via Ransac

Modify your Ransac algorithm from "Exercise in Robust Model Fitting" to fit fundamental matrices instead of lines. Hints:

- A possible way of finding eight random matches is via:
  ```
  idx=randperm(nMatch);
  idx=idx(1:8);
  ```
- Use your function `Fest_8point` to estimate the fundamental matrix from the eight point sample.
- Write a function `function dist=FSampDist(F,p1,p2)`, which implements the Sampsons distance, i.e. equation (2.40) in the lecture notes.
- Use the following code to estimate the number of inliers:
  ```
  for cM=1:nMatch,
  if(FSampDist(F,q1(:,cM),q2(:,cM))<3.84*3^ 2)
  nIn=nIn+1;
  end
  end
  ```
- Explain the term `3.84*3^ 2`.

Compare estimated/best fundamental matrix with `Ftrue`. E.g. use the formulae/code:

```
b=Ftrue(:);
a=F(:);
a'*b/(norm(a)*norm(b))
```

Explain this formulae/code

4 Refine F Estimate

Find all the inliers w.r.t. your estimated fundamental matrix. Refine this fundamental matrix estimate by running `Fest_8point`, on all the inliers. Compare estimated/best fundamental matrix with `Ftrue`, again. Comment.

Illustrate the matched inliers, and compare with the result in Section 2, i.e. without use of the fundamental matrix.

5 Repeat on the `House` images

Apply your geometry based image matching algorithm on the `House1.bmp` and `House2.bmp` images, where the fundamental matrix is unknown. Comment on the result, including the effect of applying the fundamental matrix.