Additional hints

Hi,

Some of you ran into problems, and the solutions might be of general interest.

1. Inlier counting: Hx (and also PX) is a projective quantity (an element in P^2). Since we only have a sensible noise model on the image plane (i.e. Cartesian coordinates), we need to dehomogenize (apply pflat or \pi) elements in P^2 to obtain points on the image plane (R^2). You *cannot* count homography inliers using a residual like ||y-Hx|| directly (since Hx is close to y only up to scale).

2. There is also still some confusion why the parallel RANSAC is necessary, and estimating E is not sufficient. As explained in lecture 8, estimation of E using the 7 (or 8) point method yields an infinite family of solutions when you have a (near) planar scene. If you only estimate E via the 7/8-point method in those cases, you either end up with poor models (if you do not check for degenerate configurations) or RANSAC runs a *very* long time (because most minimal sample sets will be degenerate). That is a very real practical issue (at least until the 5-point method was developed, which you may implement as alternative). You can interpret the parallel RANSAC method as generating several proposals for R and T (some via the 7/8-point method and some via H), and you have to find out if any (and which) of those proposals is leading to a new best model (more inliers, correct physical visibility, no degeneracy). So therefore you need at least to think and explain how to address near-planar scenes (and implement your thoughts) to pass the project. The TAs need at least an attempt for a solution to provide sensible feedback.

3. Regarding the LM method for the camera center / translation vector: since estimating T (or C) is a linear method, you may use all inlier 2d-3d correspondences to obtain a better initial value before running LM. And don't forget to use only inlier 2d-3d correspondences when running LM!

4. Since the rotation part of your projection matrices (e.g. R{i}) are not very accurate, do not expect very many inliers for the 2-point method. Increase the inlier treshold (I used 3 times the inlier threshold used for the epipolar constraint, but you are free to experiment with an even larger value).

Parallel RANSAC hints

The parallel RANSAC routine is probably the trickiest bit in the project. So below some suggestions after receiving some questions and code snippets.

1. Make sure the plain RANSAC for E is working correctly for the non-planar datasets. In particular, make sure you also detect and handle minimal sample sets that are in a degenerate configuration (e.g. by checking the ratio of the 2nd smallest and the largest singular value, which should not be "too small"). So basically you have to think of a criterion when the null space of the system matrix is clearly only 1-dimensional.

2. Parallel RANSAC means estimating both E and H within one iteration of the RANSAC loop. It does not mean to run RANSAC first for E, and then again for H.

3. You cannot directly compare inlier counts for E and H directly. E will almost always win (since inliers for E are based on point-line distances, and H inliers use point-point distances). That is why you have to maintain two inlier fractions (and the corresponding number of required iterations; one for E and one for H). See slide 5 in the project description.

Chirality advice

Hi,

Some of you have issues with the chirality check. My suggestions:

1. Don't forget that the triangulated points should be in front of both cameras! So X(3) > 0 and (R*X+T)(3) > 0 has to be satisfied.

2. Not all inliers using epipolar geometry are in front of both cameras even for the correct configuration. A few epipolar inliers might always end up behind one camera (meaning that they are not physically plausible inliers). One way is to adjust the number of inliers to count only the correspondences that are both inliers w.r.t. the epipolar constraints and satisfy the chirality test. What will get you likely into trouble is using code like "if Xs(3,:) > 0 && (R*Xs + T)(3,:) > 0   best_P = [R T]; end", because it assumes that for the right configuration all epipolar inliers are also in front of both cameras.

2-point method for robust translation vector estimation

Hi,

From the questions I got per email so far there seem to be 3 easy pitfalls, that you should avoid:

1. It is easy to forget to use normalized image coordinates, so multiply your keypoints with the inverse of the calibration matrix (from the left).

2. Another source for errors is to forget to rotate the 3D points X0 into the camera coordinate system via the already determined rotation part (e.g. R{i}).

3. Finally, don't forget to project R*Xj + T onto the image plane (via \pi / pflat) when calculating the inliers.

Best regards,

Christopher