Epipolar Geometry

Computer Vision 600.461

Professor: Greg Hager

Lecturer: Daniel Abretske

Motivation

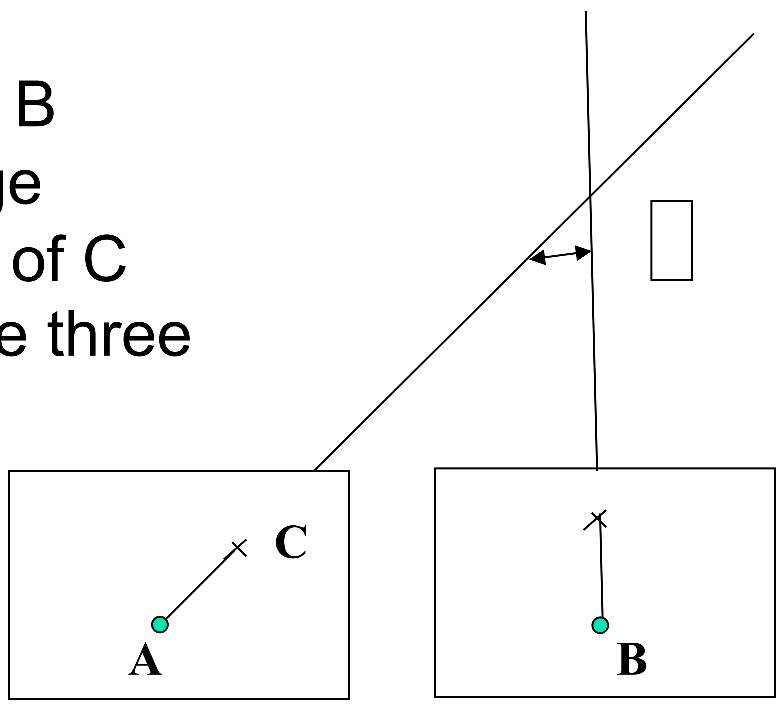
- Last lecture, we saw how to recover the structure of a scene given known poses and orientations of two cameras.
- It is possible to compute camera positions from images.
- Therefore, given two views of a scene it is (in general) possible to compute camera poses and scene structure

A Quick Stereo Review

- Given two images we rectify the images so that scanlines are aligned horizontally.
- We then triangulate image geometry from matched points.
- However, matched points have 4 coordinate values and when working with rectified images we only make use of 3 of these values.
- What information is contained in the 4th value?

Epipolar Plane

Given camera centers A and B and a point C in the left image we define the epipolar plane of C as the plane defined by these three points.



Epipolar Lines

- The epipolar line L (for a point in the left image) is the line of intersection between the epipolar plane and the right image plane.
- Corresponding points must lie along the epipolar lines.
- The point of intersection of all epipolar lines is called the epipole.

A Brief Aside

- Image rectification to make scan lines parallel is in fact the act of making all epipolar lines into scanlines.
- In terms of projective geometry this is also equivalent to making the epipoles into points at infinity(since parallel lines intersect at points at infinity).
- Further the right epipole is in fact the image of the camera center of the left image projected into the right image plane(simalarly for the left epipole)

Epipolar Geometry

- Assume we are given two images of a rigid scene. Then for a pair of matched image points x_1 (left image) and x_2 (right image)

$$\lambda_2 x_2 = R\lambda_1 x_1 + T$$

$$\lambda_i \in \Re^1, R \in SO(3),$$

$$T \in \Re^3, x_i \in P^2$$

- We would like to write a constraint on image matches that we can solve for using only image points.

Algebraic Derivation of E.

$$sk(T)x_2 \bullet \lambda_2 x_2 = sk(T)x_2 \bullet (R\lambda_1 x_1 + T)$$

$$0 = x_2^T sk(T) R x_1 = x_2^T E x_1$$

The matrix E is called the Essential Matrix and is a quadratic constraint on matching image points that we will exploit to recover camera motion.

Properties of E

1. det(E)=0
2. The left and right nullspaces of E are respectively the left and right epipoles
3. An epipolar lines can be expressed in terms of E as

$$l_2 = Ex_1$$

$$l_1 = E^T x_2$$

Properties of E

1. A matrix E is an essential matrix if and only if for [U,D,V]=svd(E), U and V are both rotations and D = diag([k,k,0]).
2. E is also subject to an arbitrary scaling factor.

Estimating E

- R and T have a total of 6 unknowns, however since we can apply an arbitrary scaling of the scene this reduces to a total of 5 unknowns.
- This implies that the minimal number of points we should need to solve for the essential matrix E is 5.
- There are in fact algorithms to find E from 5,6,7 and 8 points but algorithms that use less than 8 points require nonlinear solvers and are much more complex.

Eight Point Algorithm

- If we carry out all the calculations it turns out we can write $x_2^T E x_1 = 0$ as a vector product.

$$a = x_1 \otimes x_2$$

$$= [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2]$$

$$= [x_1x_2, x_1y_2, x_1, y_1x_2, y_1y_2, y_1, x_2, y_2, 1]$$

$$e = vec(E) = [e_1, e_4, e_7, e_2, e_5, e_8, e_3, e_6, e_9]$$

$$a \bullet e = 0$$

Eight Point Algorithm

- Since E is known up to scale we stack at least 8 ‘a’ vectors into a matrix ‘A’ and then apply a least squares solution for homogeneous systems.
- In practice, to find the solution we take [U,D,V]=svd(A) and last column of V to be ‘e’, i.e., the essential matrix can be found from the nullspace of A.
- In the presence of noise it is necessary to take the svd of E and enforce the singularity of the matrix and the equality of the singular values.

Decomposition of E

- Given an E the next logical step is to find the R and T that give rise to the given geometry.
- It turns out that there will be 4 possible solutions as it is necessary to consider both E and -E in the face of noise.

Decomposition of E

- The following formulas give the possible camera motions. It is possible to eliminate 3 of the 4 solutions by requiring that all point reconstructions have positive depth. Again [U,D,V]=svd(E). For the details of the proof you can refer to Yi Ma et al. ‘An invitation to 3-D Vision’

$$R = UR_Z^T(\pm\frac{\pi}{2})V^T$$

$$sk(T) = UR_Z(\pm\frac{\pi}{2})DU^T$$

$$R_Z(\pm\frac{\pi}{2}) = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Fundamental Matrix

- All of the above assumed that we were working with calibrated cameras. If the cameras are not calibrated we can still recover a geometric relation between images. This is known as the Fundamental Matrix, F.

$$F = K^{-T} E K^{-1}$$

- The algorithms described to recover E from image points also apply to recovering F, however it is not possible to recover R and T from F due to the calibration ambiguity.

Homographies

- It’s important to point out that there are certain degenerate configurations of points for which epipolar reconstruction fails.
- The most important case is where all the points lie on a plane. In this case it is necessary to work with homographies instead of the essential matrix.
- There are other degenerate configurations but they are much rarer

Additional Topics

- It is possible to perform similar derivations relating 3 images and the resulting constraint is known as the Trifocal Tensor.
- The previous discussion also assumed scenes were rigid with only the camera moving, there are constraints for multiple rigid motions in a scene and work has been done for non-rigid scenes.

Conclusions

- It is possible to start with two images (both taken with the same camera even) and create a reconstruction of a scene using only image points by combining epipolar geometry and stereo reconstruction.
- It is in fact possible to combine these steps into a single reconstruction step which will be the topic of the next lecture.