ECE 661
Purdue Univ.
Avi Kak
Lecture 24

Binocular Stereo: Image Rectification and Scene Reconstruction

- Our goal is to construct a 3D model of a scene from its two views. For that we need to first solve the correspondence problem — which means that for every pixel in the first view we must try to find the corresponding pixel in the second view. Solving the correspondence problem is made easier if we first rectify the images. Rectification causes the epipolar lines to coincide with image rows. After rectification, when we search for the corresponding pixel in the second image, we only need to look at the same row (in the second image) as for the pixel in the first image. To do a good job of image rectification, we need a high-quality estimate for the fundamental matrix F.

- Do you see a chicken-and-egg situation here: We need some correspondences in order to estimate F so that we can rectify the images and find even more correspondences.

- In general, unless you get lucky, the estimate of F produced by the linear least-squares algorithm — even with the conditioning mentioned on page 23-6 of Lecture 23 — is not sufficiently accurate for good image rectification. (You are not likely to find the roughly 40 correspondences that you can trust for your linear least-squares estimate. You may have to make do with far fewer — maybe just the minimum number of 8 correspondences.)

- So our first order of business is to refine F by using RANSAC and by nonlinear least-squares minimization (Lecture 13) of a geometric cost function — just as we did for refining our homography estimates and our estimates of camera calibration parameters.

- While it would be relatively easy to combine RANSAC with linear least-squares estimation of F as described in Lecture 23, how would one define a geometric cost function that can be expressed as $d^2_{geom} = \|\vec{X} - \vec{f}(\vec{p})\|^2$, where $\vec{X}$ represents the measurements and $\vec{f}$ the predictions for those measurements using the parameter vector $\vec{p}$? [The geometric cost functions for both homography estimation and the estimation of camera calibration parameters were based on the following idea: Use the currently known values for the parameters $\vec{p}$ to project a a set of points into the image where our measurements $\vec{X}$ reside. Let the squares of the discrepancies between the projected points and the measured points define the geometric error. How can we do the same for refining F?]

- To set up a geometric cost function for refining F we proceed as follows: Given a canonical configuration (P, P') for the two cameras, we use the current estimate for F to "triangulate" each pair $(\vec{x}_i, \vec{x}_i')$ of corresponding points into a world point $\vec{X}_i$. Subsequently, we use the camera matrices P and P' to project $\vec{X}_i$ back into the two images. Let these projections be denoted $(\hat{\vec{x}}_i, \hat{\vec{x}}')$. We now define the geometric error as $\boxed{d^2_{geom} = \sum_i \left(\|\vec{x}_i - \hat{\vec{x}}_i\|^2 + \|\vec{x}_i' - \hat{\vec{x}}_i'\|^2 \right)}$. As you will see later, both $\hat{\vec{x}}$ and $\hat{\vec{x}}'$ are functions of the elements of F. This error can be cast in the form $\|\vec{X} - \vec{f}(\vec{p})\|^2$ that is needed by nonlinear least squares.

- Before we can translate the above description into an algorithm for geometric cost minimization, we still have a couple of issues to resolve:
  1. There is no unique canonical configuration (P, P') for the two cameras. As mentioned in Lecture 23, canonical configuration means $P = [I|\vec{0}]$ and $P' = [SF|\vec{e}']$ for any 3x3 skew-symmetric S that results in a rank 3 matrix for P'. By convention, we choose $S = [\vec{e}']_\times$. That gives us following specific canonical configuration: $\boxed{P = [I|\vec{0}], \; P' = [[\vec{e}']_\times F|\vec{e}']}$. Recall that $\vec{e}'$ is the left null vector of F: $\vec{e}'^T F = \vec{0}^T$
  2. Given a pair of corresponding points $(\vec{x}, \vec{x}')$, how does one "triangulate" and obtain the world point X_i? We will address this issue next.

Computing the Scene Structure from a Set of Correspondences for a Given Camera Pair (P, P')

- Although we arrived at this topic in the context of setting up a geometric-cost minimization algorithm for refining F, the formulas presented in this section also apply to the computation of the 3D structure of the scene after you have refined F, carried out image rectification, and obtained a large set of final correspondences. I'll therefore present this section in a generic manner — so that it can be used for both geometric-cost minimization and final 3D structure computation.

- Given a correspondence $(\vec{x}, \vec{x}')$ and a pair of cameras (P, P'), what is is the best value for the world point $\vec{X}$ so that $\vec{x} = P\vec{X}$ and $\vec{x}' = P'\vec{X}$? Since $\vec{x}, \vec{x}', P$, and P' are all known, our goal is to express $\vec{x} = P\vec{X}$ and $\vec{x}' = P'\vec{X}$ in the form of a homogeneous system of equations $\boxed{A\vec{X} = \vec{0}}$.

- In order to obtain the homogeneous form $A\vec{X} = \vec{0}$, let's first first on $\vec{x} = P\vec{X}$. Writing $\vec{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, we have $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \times P\vec{X} = \vec{0}$. If we write P as $P = \begin{bmatrix} \vec{P}^{1T} \\ \vec{P}^{2T} \\ \vec{P}^{3T} \end{bmatrix}$, we have $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \times \begin{pmatrix} \vec{P}^{1T}\vec{X} \\ \vec{P}^{2T}\vec{X} \\ \vec{P}^{3T}\vec{X} \end{pmatrix} = \vec{0} \Rightarrow$

$$\begin{aligned} y(\vec{P}^{3T}\vec{X}) - (\vec{P}^{2T}\vec{X}) &= 0 \\ -x(\vec{P}^{3T}\vec{X}) + (\vec{P}^{1T}\vec{X}) &= 0 \\ x(\vec{P}^{2T}\vec{X}) - y(\vec{P}^{1T}\vec{X}) &= 0 \end{aligned}$$

The three equations we get are not linearly independent. In fact, x times the first eqn when added to y times the second eqn gives us the third equation. Therefore, $\vec{x} = P\vec{X}$ gives us 2 eqns:

$$\begin{aligned} y(\vec{P}^{3T}\vec{X}) - \vec{P}^{2T}\vec{X} &= 0 \\ x(\vec{P}^{3T}\vec{X}) - \vec{P}^{1T}\vec{X} &= 0 \end{aligned}$$

- Exactly the same kind of reasoning when applied to $\vec{x}' = P'\vec{X}$ leads to the following two equations:

$$y'(\vec{P}^{3T}\vec{X}) - \vec{P}^{2T}\vec{X} = 0$$
$$x'(\vec{P}^{3T}\vec{X}) - \vec{P}^{1T}\vec{X} = 0$$

- Stacking the two equations at the bottom of the previous page with the two equations shown above, we end up with the homogeneous system of equations $A\vec{X} = 0$ where the 4x4 A is:

$$A = \begin{bmatrix} x\vec{P}^{3T} - \vec{P}^{1T} \\ y\vec{P}^{3T} - \vec{P}^{2T} \\ x'\vec{P}'^{3T} - \vec{P}'^{1T} \\ y'\vec{P}'^{3T} - \vec{P}'^{2T} \end{bmatrix} \quad \begin{matrix} \uparrow \\ 4 \\ \downarrow \end{matrix} \qquad \leftarrow 4 \rightarrow$$

- As we have done before, we can use the linear least-squares algorithm to solve for $\vec{X}$. Our goal is to minimize $\|A\vec{X}\|$ subject to the constraint $\|\vec{X}\| = 1$. The solution is given by the smallest eigenvector of A^TA. Note that since $\vec{X}$ is a homogeneous 4-vector for a point in 3D space, the constraint $\|\vec{X}\| = 1$ is a matter of no consequence.

- What I have described so far in this section is all you need if you want to calculate the points $\vec{X}$ for plugging into the geometric cost minimization framework for the refinement of the fundamental matrix as described in the previous section.

- However, if you are engaged in scene structure computations after you have found the final set of correspondences, you need to bear in mind the following issues:
  - The scene structure you calculate will be Euclidean (meaning an exact replica of the original) only when the cameras are calibrated.
  - However, there is a lot of research interest in constructing the scene structure with uncalibrated cameras. In the algorithms that have been developed, while the camera can figure out on its own how to compute its K matrix, you rely on the constraints you can derive from multiple images of the same scene to compute a structure that bears affine (or, even similarity) resemblance to the original scene.
  - In general, if you have only two views and you are using uncalibrated cameras, the best you can do is to construct a projective resemblance to the original scene. What that means is that your reconstruction will be related to the original scene by a 4x4 homography H. I'll dwell on this in greater detail in the rest of this section.

- To understand the claim made above, let's see what happens to an actual camera pair (P, P') when the world 3D is subject to a projective transformation with a 4x4 homography. Before this transformation, we have $\vec{x} = P\vec{X}$, $\vec{x}' = P'\vec{X}$. Now let's transform the world frame by $\vec{X}_{new} = H\vec{X}$ where H is any 4x4 homography. So we have $\vec{X} = H^{-1}\vec{X}_{new}$. That leads to $\vec{x} = (PH^{-1})\vec{X}_{new}$ and $\vec{x}' = (P'H^{-1})\vec{X}_{new}$.

Therefore, a projective distortion of the 3D space is tantamount to the two cameras becoming modified projectively by right-multiplication with H^{-1}!

- You saw on pages 23-5 and 23-6 (Lecture 23) that a fundamental matrix F is invariant to the right-multiplication of the camera matrices P and P' by a 4x4 projective homography. [We can also show that for every camera P there exists a 4x4 projective homography so that $P \cdot H = [I|\vec{0}]$.]

- Putting two and two together, we conclude that when you reconstruct a scene with the two cameras in a canonical configuration, your reconstruction will be related to the true scene through a projective transformation. What that means is that if the scene contains an object like the one at (A), its reconstruction with two uncalibrated cameras will look like what is shown at (B). [When we say 'uncalibrated cameras', we really mean that. We don't find even the intrinsic parameters for the cameras. We do a good job of estimating the fundamental matrix directly from the images. We then construct a canonical configuration for the cameras with F and its left null vector. Finally, we compute the scene structure with the cameras in this configuration.]

(A)

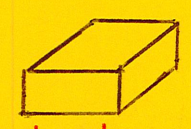
original object

(B)

Reconstruction with two uncalibrated cameras

(C)

Affine restoration of the reconstruction



(D)

Similarity restoration of the reconstruction

- We are then faced with the question of whether it is possible to remove the distortion in the reconstructed structure to any extent at all. The answer depends on how rich the scene is and whether or not we can estimate from the reconstructed structure the image of $\vec{\pi}_\infty$ and the image of Ω_∞.

- As you know from Lecture 7 (page 7-4), a 4x4 homography is affine iff $\vec{\pi}_\infty$ goes into $\vec{\pi}_\infty$. What that implies is that a purely projective distortion maps $\vec{\pi}_\infty$ to a plane $\vec{\pi}$ in 3D space. Assume for a moment that we can estimate such a $\vec{\pi}$ from the scene reconstruction shown at (B) above. Then the homography H that would eliminate the "purely" projective distortion and give you the affine reconstruction shown at (C) is given by

$$H = \begin{bmatrix} I_{3\times 3} & \vec{0} \\ \hline \vec{\pi}^T & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{bmatrix} \quad \text{assuming } \vec{\pi} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix}.$$

That this is the correct 4x4 homography to use follows from the fact (see Lecture 7) that when world 3D is transformed with a 4x4 homography H, a plane $\vec{\pi}$ is transformed as $H^{-T}\vec{\pi}$. You can yourself show that with the H shown we get $\boxed{H^{-T}\vec{\pi} = \vec{\pi}_\infty = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}}$.

- So the main issue with creating an affine restoration of the scene structure is estimating from the projective structure the plane $\vec{\pi}$ to which $\vec{\pi}_\infty$ was transformed. Perhaps the easiest way to construct this plane in a projective reconstruction is to estimate the 3D coordinates of the points of intersections of the reconstructions of three non-coplanar sets of parallel lines in the original scene.

- After we have gotten rid of the purely projective distortion and obtained an affine reconstruction of the scene, the next question is whether we can take the reconstruction one step further — to obtain a similarity reconstruction as shown at (D) on the previous page. The answer is 'yes' provided we can estimate the image of the absolute conic Ω_∞ in one of the cameras (after the application of the homography that gives us the affine reconstruction.). This is explained in what follows.

- Just like a 4x4 transformation is affine iff $\vec{\pi}_\infty$ remains unchanged, a 4x4 transformation belongs to the similarity group iff Ω_∞ remains unchanged. The proof in both directions is pretty simple. First let's prove that if Ω_∞ remains unchanged then a 4x4 homography H is of similarity type: If Ω_∞ remains fixed under H, then so does $\vec{\pi}_\infty$ since Ω_∞ "resides" on $\vec{\pi}_\infty$. This means H must be at least affine: $H = \begin{bmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix}$. We know that when 3D space transforms as H, a quadric Q transforms as $H^{-T} Q H^{-1}$. Although we are accustomed to thinking of Ω_∞ as a conic on $\vec{\pi}_\infty$, you can also think of it as a quadric consisting of points whose first three homogeneous coordinates obey $(x_1\ x_2\ x_3) I \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0$ and whose last coordinate obeys $x_4 = 0$. What that means is that the matrix Q in $H^{-T} Q H^{-1}$ will act as I_{3x3} with respect to the first three homogeneous coordinates. If you plug the affine form of H in this, you'll see that $H^{-T} Q H^{-1}$ reduces to $A^{-T} I A^{-1}$ with respect to the first three homogeneous coordinates. Since the transformation $H^{-T} Q H^{-1}$ must again yield the ~~ab~~ absolute conic, so it must be the case that $A^{-T} I A^{-1} = cI$ for some scalar multiplier c. That is the same as saying $AA^T = cI$ which is the condition for H to be a similarity transform. (The proof in the opposite direction is even simpler.)

- So in order to restore the scene structure to the similarity level, our goal should be to find the image of Ω_∞ in one of the cameras and figure out the transformation that takes this image back to $K^{-T} K^{-1}$ (which is the image of the Absolute Conic in the standard Euclidean coordinate frame).

- As it turns out it's best to find the Ω_∞ restoring transformation after you have restored the scene structure to the affine level — meaning after you have ensured that the image of $\vec{\pi}_\infty$ is back where it belongs.

- I'll now state without proof the following procedure for restoring the scene structure to similarity level: Choose one of the two cameras in which you want to restore the image of Ω_∞ to $K^{-T} K^{-1}$. If P was the projection matrix for this camera before the affine restoration, after that restoration this camera matrix would become $P H_a^{-1}$ where H_a is the homography used for restoring the scene structure to the affine level. Let's express this projection matrix as $[M | \vec{m}]$. One can show that the image of the absolute conic in this affine-distorted camera is ~~(M I M)~~ $(M^T K^{-T} K^{-1} M)^{-1}$. This image of

of Ω_∞ can be restored to its standard form $K^{-T}K^{-1}$ by the following homography: $H = \begin{bmatrix} A^{-1} & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix}$ where the 3x3 A is given by the Cholesky decomposition $\longrightarrow$ $AA^T = (M^T K^{-T} K^{-1} M)^{-1}$. What this implies is the similarity reconstruction of the scene structure cannot be carried out from just two views without access to the intrinsic parameters of the camera as represented by the matrix K.

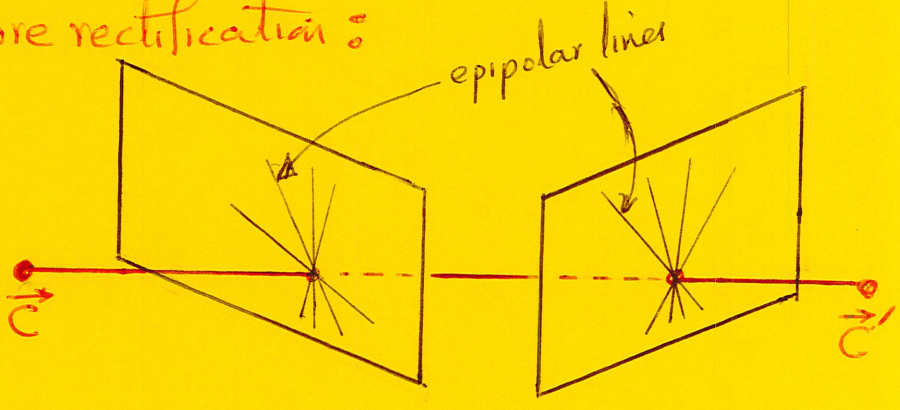
Image Rectification

- Before you can compute your projectively distorted scene structure (with the cameras in a canonical configuration), you must use the refined estimate for the fundamental matrix F for determining your final list of image pixel correspondences $(\vec{x}_i, \vec{x}_i')$ for $i = 1, 2, 3, \ldots, N$ for ~~some~~ hopefully a large value for N.

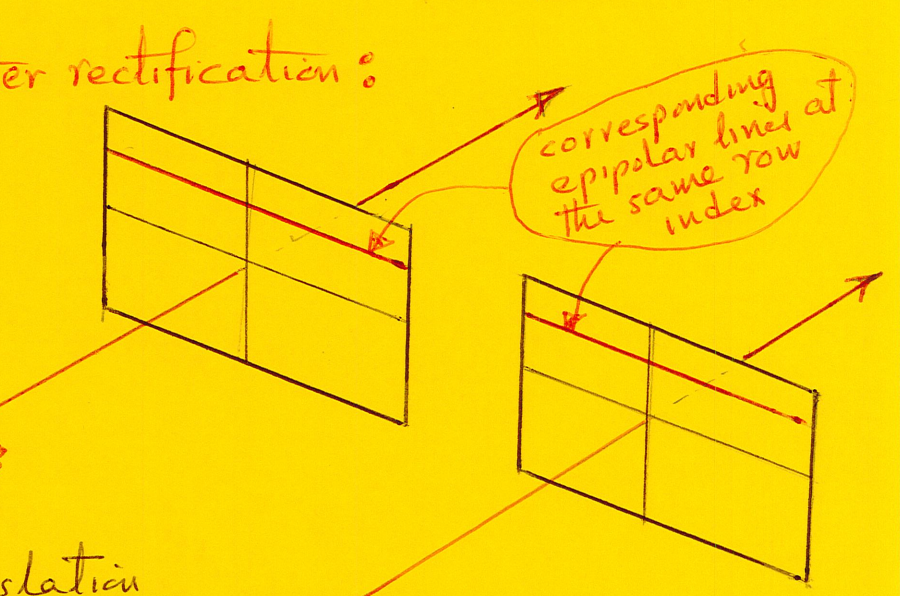
- The following question is very relevant to finding a large number of correspondences: Is it possible to apply 3x3 homographies to the two images so that the epipolar lines will be parallel to the rows and will match up between the two images? If we refer to these homographies by H and H′ —assuming they exist— applying them to the two images and generating a new pair of stereo images with matched rows for the epipolar lines is called image rectification.

- As illustrated by the lower figure on the right, we want the epipolar lines to become parallel to the image rows. This means we want to move the epipoles out to infinity. Assuming the image x-axis is along the rows, we want $\vec{e} = \vec{e}' = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$.

Before rectification:



After rectification:



- Pretend for a moment that the epipole in the right image is at $(f, 0, 1)^T$. It is easy to show that the homography $G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$ will take that epipole to $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$.

f is just some number. It has nothing to do with the focal length

- The homography H′ we need for the right image would then be $H' = GRT$ where R and T are the rotation and the translation needed to move the epipole from wherever it is to $\begin{pmatrix} f \\ 0 \\ 1 \end{pmatrix}$.

from the right null vector of F

- Finding the homography H for the left camera is more challenging because we not only want to send its epipole to $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, we also want the epipolar lines in the left camera to match up with those in the right camera. See Hartley and Zisserman how to best find H for a given H′. Note that H and H′ are just for solving the corresp. problem. For scene reconstruction, you'll continue to use the original images