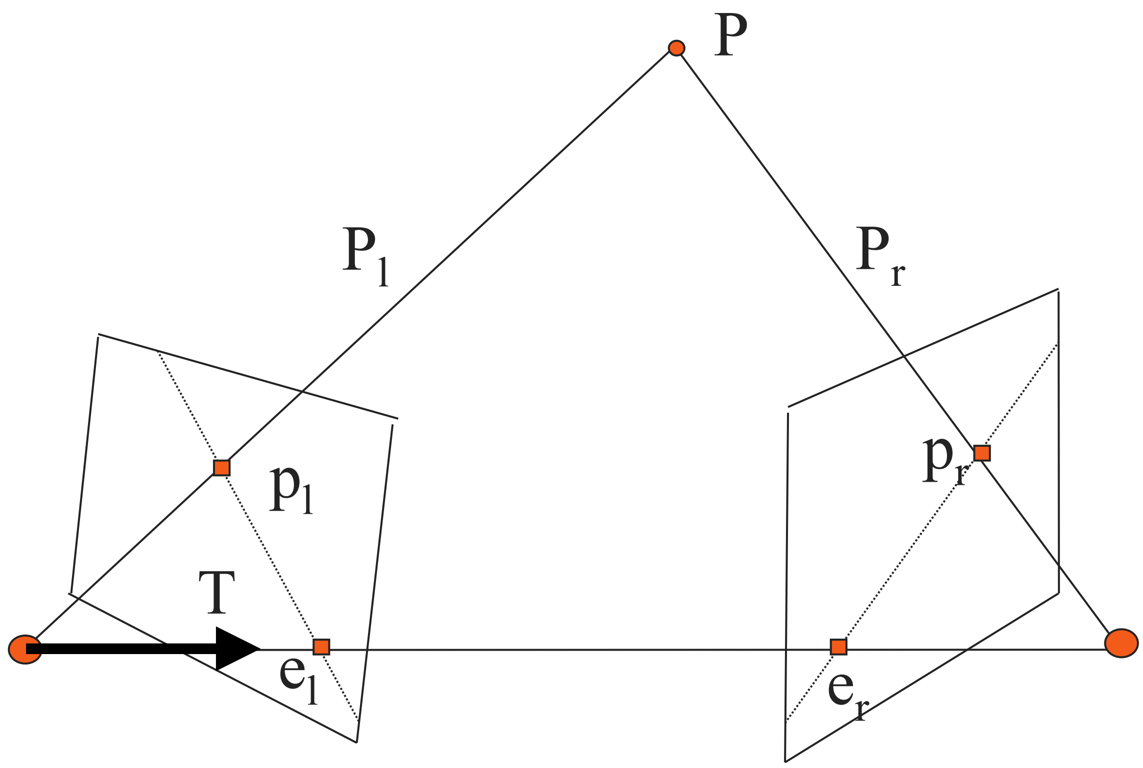
MEAM 620 - Sensing

Spring 2020

C.J. Taylor

2 View Geometry

Epipolar Geometry



$$
\begin{aligned}
p_l &\propto A_l P_l \\
p_r &\propto A_r P_r \\
P_l &= R P_r + T
\end{aligned}
$$

- Note that the vectors P_l, RP_r and T are *coplanar* hence:

$$P_l^T (T \times (RP_r)) = 0 \;\Rightarrow\; P_l^T (\hat{T} R) P_r = 0$$

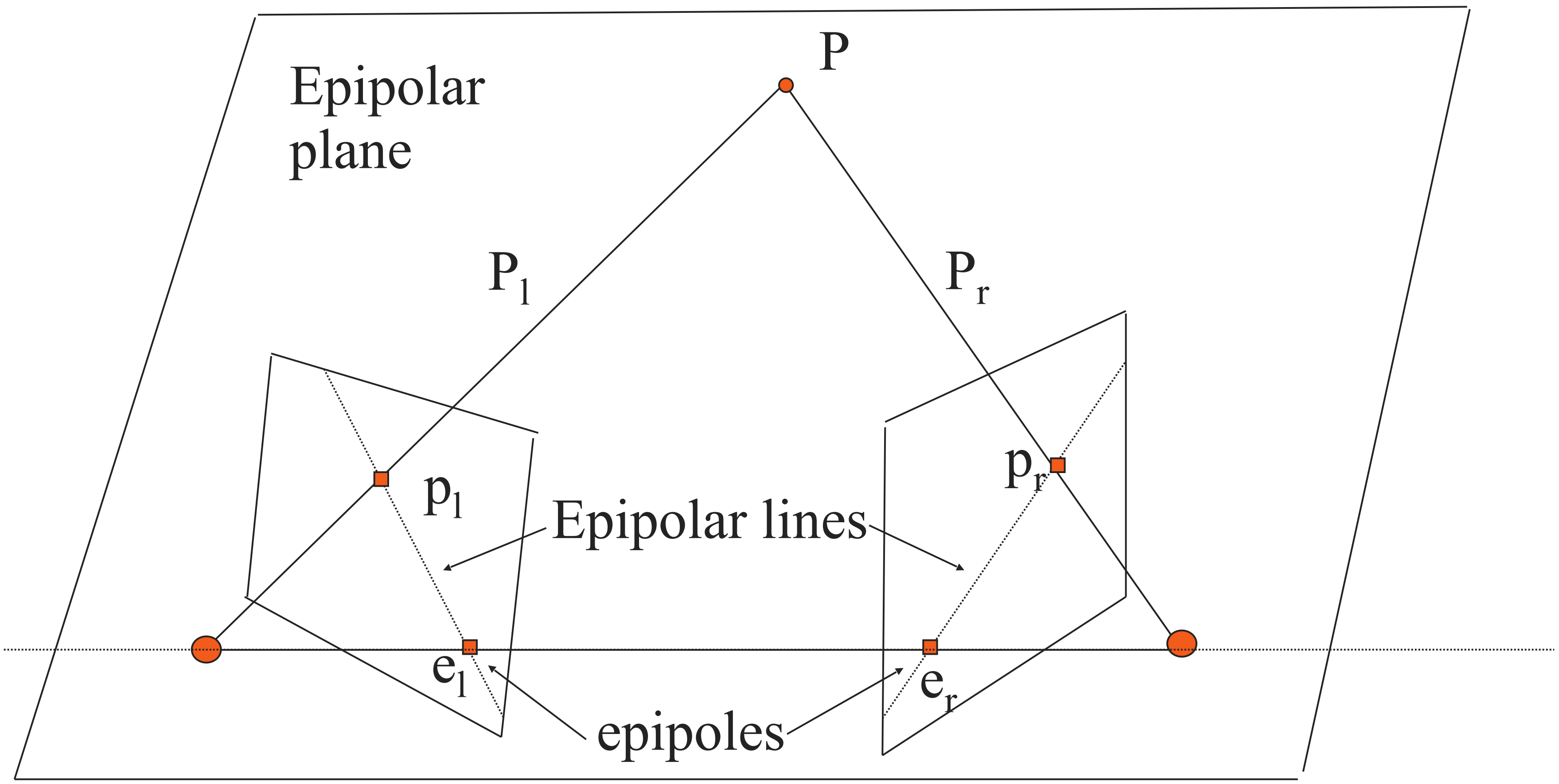
Essential and Fundamental Matrices

- The matrix $E = (\hat{T}R)$, $E \in \mathbb{R}^{3\times3}$ is referred to as the *Essential Matrix*
- The equation $P_l^T E P_r = 0$ is referred to as the epipolar constraint.
- Since $p_l \propto A_l P_l$ and $p_r \propto A_r P_r$ we could also write:

$$(A_l^{-1} p_l)^T E (A_r^{-1} p_r) = 0 \;\Rightarrow\; p_l^T (A_l^{-T} E A_r^{-1}) p_r = 0$$

- The matrix $F = A_l^{-T} E A_r^{-1}$, $F \in \mathbb{R}^{3\times3}$ is referred to as the *Fundamental Matrix*
- The expression $p_l^T F p_r$ is another statement of the epipolar constraint in terms of image coordinates, p_l and p_r.

Epipolar Geometry

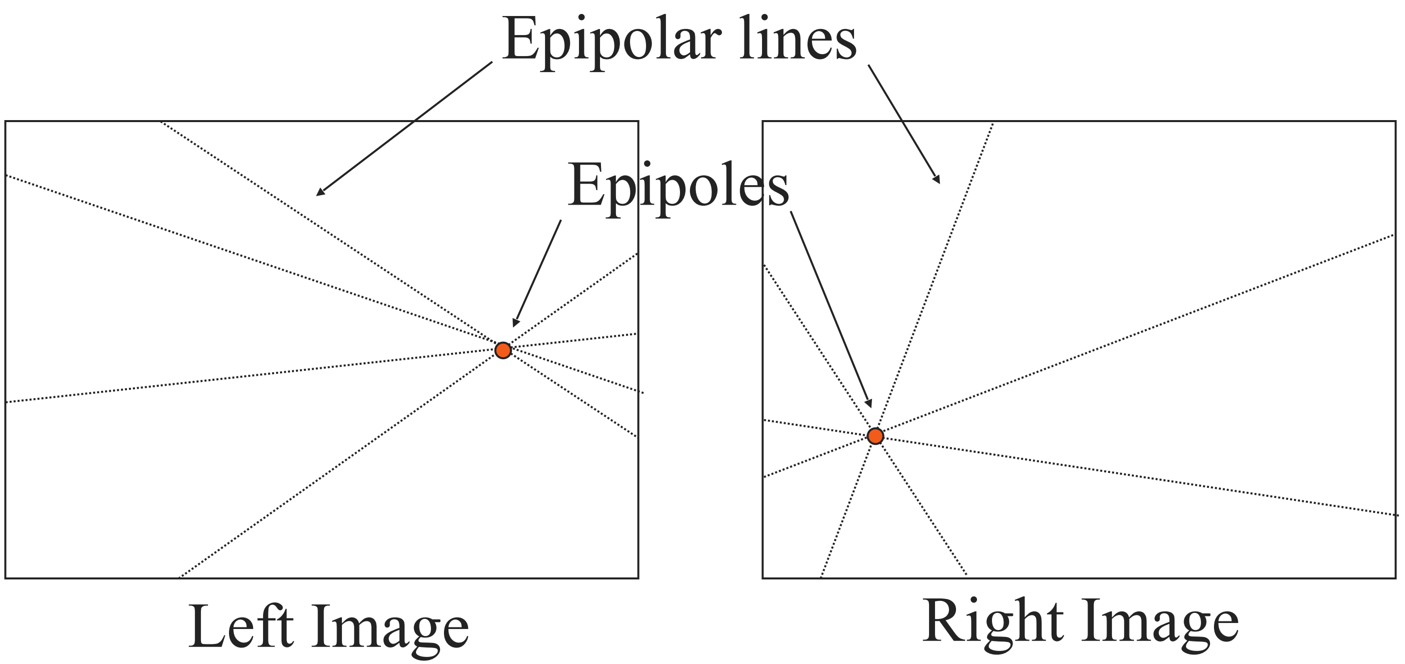


Epipolar Geometry (Perspective Projection)

- Consider the set of planes that pass through the centers of projection of both cameras. These planes are referred to as *epipolar planes*
- In each image, the points on a given epipolar plane project onto an *epipolar line*.

Epipolar Geometry continued

- The *epipole* in each image corresponds to the projection of the center of projection of the other camera.
- All of the epipolar lines in each image pass through the epipole in that image



Consequences

- The epipolar constraint
  - For every point observed in the left image we know that its correspondent must lie along the corresponding epipolar line in the right image
  - For every epipolar line in the left image there is a corresponding epipolar line in the right image
- This observation can substantially simplify the search for correspondences

Recovering F from point correspondences

- Given 8 or more correspondences we can recover the fundamental matrix, F, from the resulting set of homogenous linear equations via SVD.

$$p_l^T F p_r = 0$$

$$\begin{pmatrix} p_l^1 & p_l^2 & p_l^3 \end{pmatrix} \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} p_r^1 \\ p_r^2 \\ p_r^3 \end{pmatrix} = 0$$

$$\sum_{i=1}^{3} \sum_{j=1}^{3} p_l^i p_r^j F_{ij} = 0$$

Recovering E from F

- Given the Fundamental matrix F and the intrinsic matrices, A_l and A_r we can easily recover the Essential matrix, up to an unknown scale factor.

$$F \propto A_l^{-T} E A_r^{-1} \;\Rightarrow\; E \propto A_l^T F A_r$$

Solving Homogenous Least Squares Problem

- Accumulating all of the homogenous linear equations into a tall, skinny linear system we end up with the following optimization problem: $\min_{\mathbf{x}} \|M\mathbf{x}\|^{\mathbf{2}}$. Where $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ is a vector of our unknowns (in this case the elements of the matrix F) and $M \in \mathbb{R}^{m \times n}$ is a matrix that collects the homogenous constraints.

- The solution to $\min_{\mathbf{x}} \|M\mathbf{x}\|^{\mathbf{2}}$ is the eigenvector of the positive semi-definite matrix $M^T M$ with the smallest eigenvalue.

- This problem is often solved by computing the singular value decomposition, svd, of the matrix M since this is typically a more numerically stable approach.

Singular Value Decomposition (SVD)

- Remember that every matrix $M \in \mathbb{R}^{m \times n}$ (we assume $m \geq n$ wlog) can be decomposed into the product of 3 matrices $U \in \mathbb{R}^{m \times n}$, $S \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{n \times n}$ as follows

$$M = USV^T$$

  Where $U^T U = V^T V = I_n$ and S is a diagonal matrix where the diagonal entries are all non-negative and sorted in decreasing order of magnitude.

- Given this decomposition it is easy to see that the eigenvalues of $M^T M$ correspond to the squares of the singular values and the eigenvector corresponding to the smallest eigenvalue is the last column of the matrix V.

- The SVD is frequently used for a number of purposes in Computer Vision, Machine Learning and Robotics.

Decomposing the Essential Matrix

- It is possible to decompose the essential matrix, E, to recover the corresponding rotation matrix R and the translation vector T up to an unknown scale factor.

- To see this recall that $E = \hat{T}R$ where $\hat{T} \in \mathbb{R}^{3\times 3}$ is a skew symmetric matrix.

- T can be rewritten as $T = \lambda\Theta e_z$ for some $\Theta \in SO(3)$. So $\hat{T}$ can be rewritten as $\hat{T} = \lambda\Theta\hat{e}_z\Theta^T$

- Note that

$$\hat{e}_z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = diag(1,1,0)W, \quad W = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad W \in SO(3)$$

- Putting it all together we get $E \propto (\Theta)\, diag(1,1,0)\, (W\Theta^T R)$. From which we conclude that E should have two equal singular values and one singular value of 0.

Decomposing the Essential Matrix

- Putting it all together we get $E \propto (\Theta)\, diag(1,1,0)\, (W\Theta^T R)$. From which we conclude that E should have two equal singular values and one singular value of 0.

- Associating the terms above with the singular value decomposition of E, namely $E = USV^T$ we see that $U \propto \Theta$ and $V^T \propto W\Theta^T R$.

- When we consider that $E \propto \hat{T}R$ we see that $T \propto u_3$ where u_3 denotes the last column of U.

- To summarize, given an essential matrix and its singular value decomposition $E = USV^T$ there are 4 possible solutions for R and T as follows:

$$(UWV^T, u_3),\ (UWV^T, -u_3),\ (UW^TV^T, u_3),\ (UW^TV^T, -u_3)$$

  Note that we have assumed that we have scaled things so that U and V are both in $SO(3)$. That is both have determinant 1.

- To deduce which one is the true solution we should plug them back in and determine which one results in the reconstructed points being in front of both cameras. This is referred to as a cheirality constraint.

Summary

- Given a single camera (monocular system) observing a set of stationary points as it moves through the scene from one location to another, you can recover the fundamental matrix, F, from the 8 point algorithm (assuming you have at least 8 such correspondences).
- If you know the matrix of intrinsic parameters you can recover the essential matrix E from the fundamental matrix F.
- If you know the Essential matrix you can decompose it analytically to recover the motion of the camera between the two frames, R and T up to an unknown scale factor.

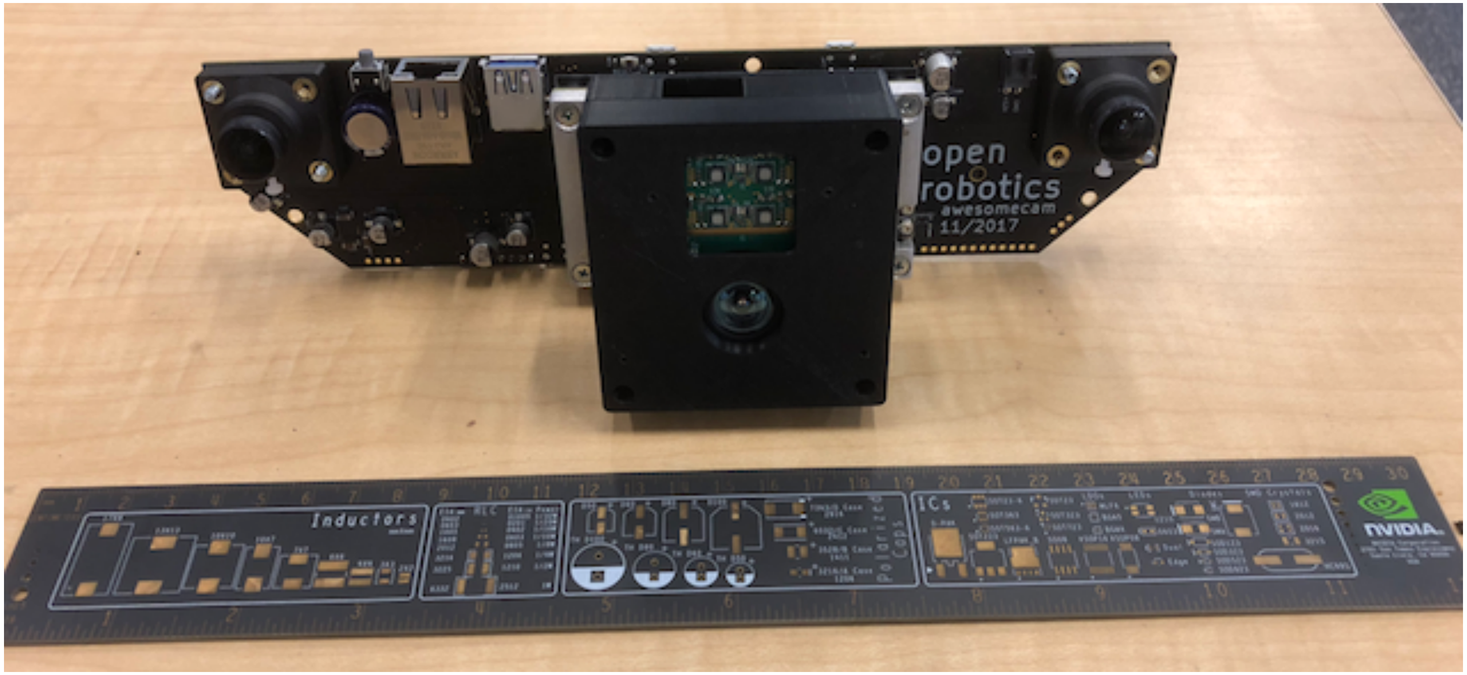
Recovering Scale

- Note that given only image measurements of a static scene there will always be a fundamental scale ambiguity which cannot be resolved from the image measurements alone.
- To resolve the scale ambiguity you need another source of metric information such as a known length in the scene which you can use to rescale the answer.
- Scale can also be recovered from accelerometer measurements since the accelerometer provides scaled measurements of the second derivative of position. Systems that couple IMU and camera measurements together to estimate pose are referred to as Visual Inertial Odometry (VIO) systems. (Note that the platform needs to undergo significant translational acceleration to produce a non-zero accelerometer reading for scaling purposes)

Stereo Rigs

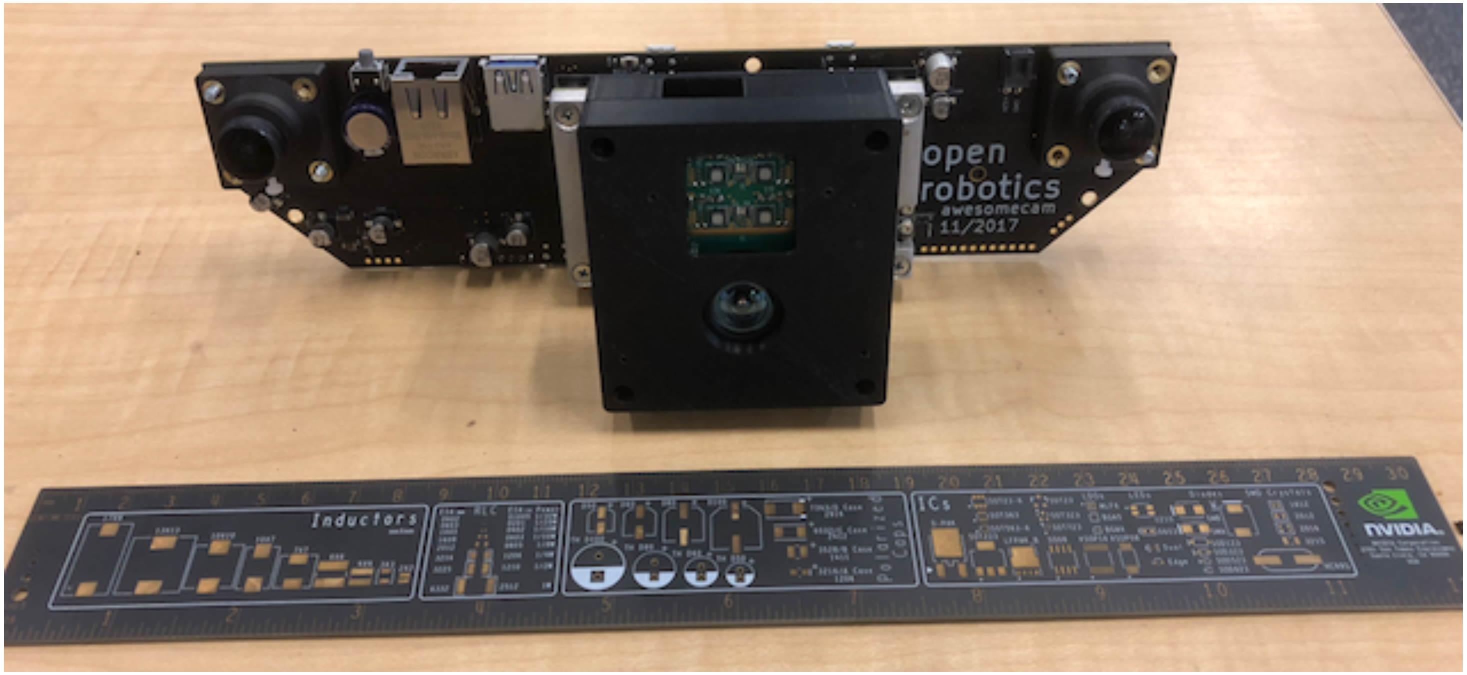
Stereo

- In order to derive 3d measurements of the scene we often construct systems with two cameras separated by a fixed baseline. These systems are referred to as stereo rigs.
- With two cameras we can triangulate features in the scene to determine their depth much as the human visual system does.



Stereo

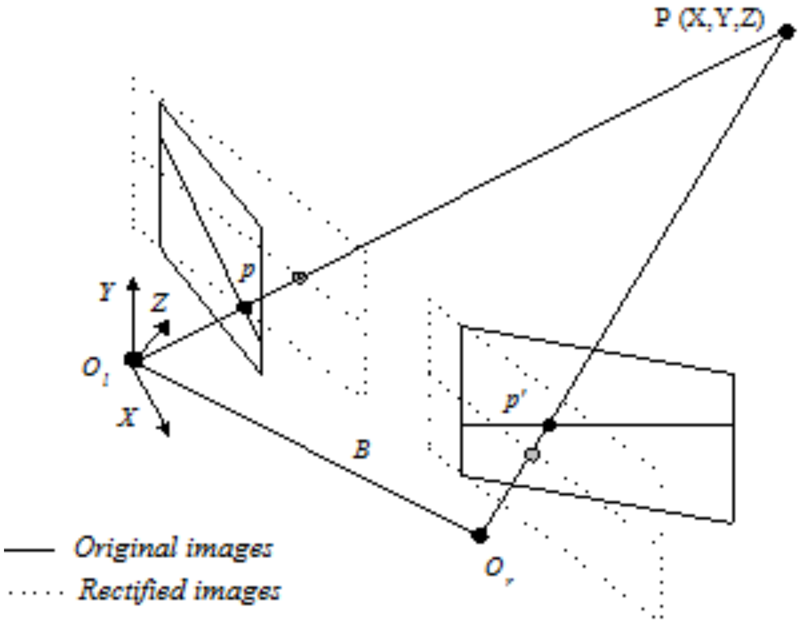
- All of the equations we derived for 2 view systems apply to the stereo configuration shown below.
- We can calibrate the system to recover the intrinsic parameters of both cameras and the relative transformation between them



Stereo Rectification

Stereo Rectification

- We will find it convenient to rectify the two images we get from the stereo rig to produce a simpler geometry.
- The rectified images will appear to come from a pair of cameras with identical intrinsic parameters whose optical axes are parallel and where the stereo baseline is aligned with the x-axis
- In this case corresponding features in the world will image on the same row in both images. This is also referred to as placing the epipoles at infinity.



Stereo Rectification

- Stereo rectification is simply a transformation of the images that simplifies subsequent subsequent stereo tasks like matching and triangulation.

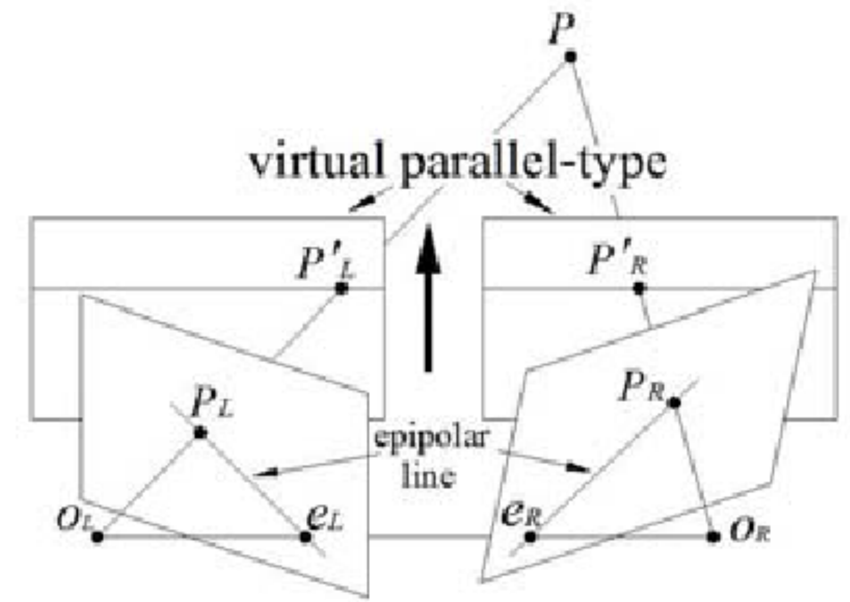
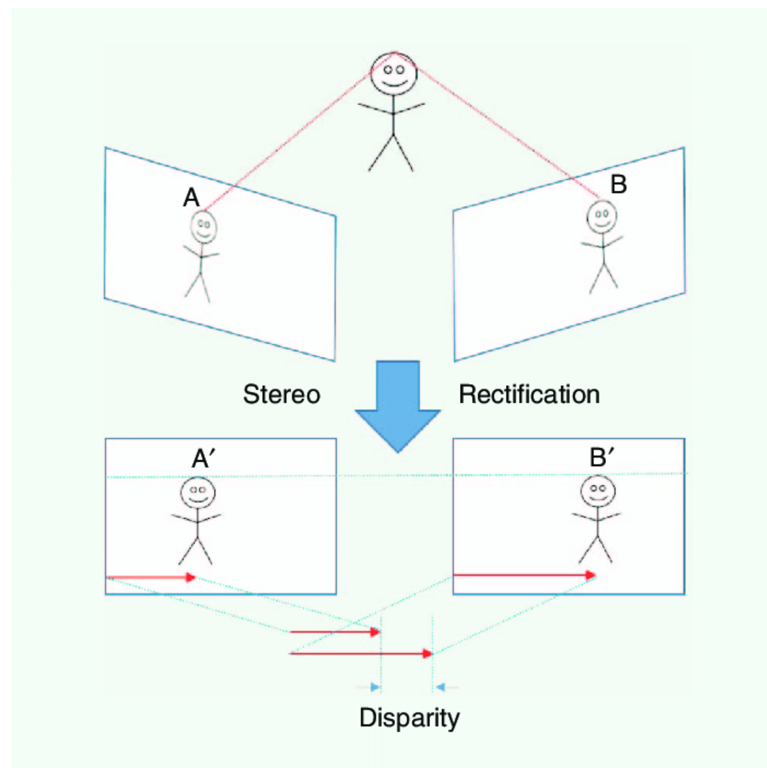


Figure 3. The schematic of stereo rectification

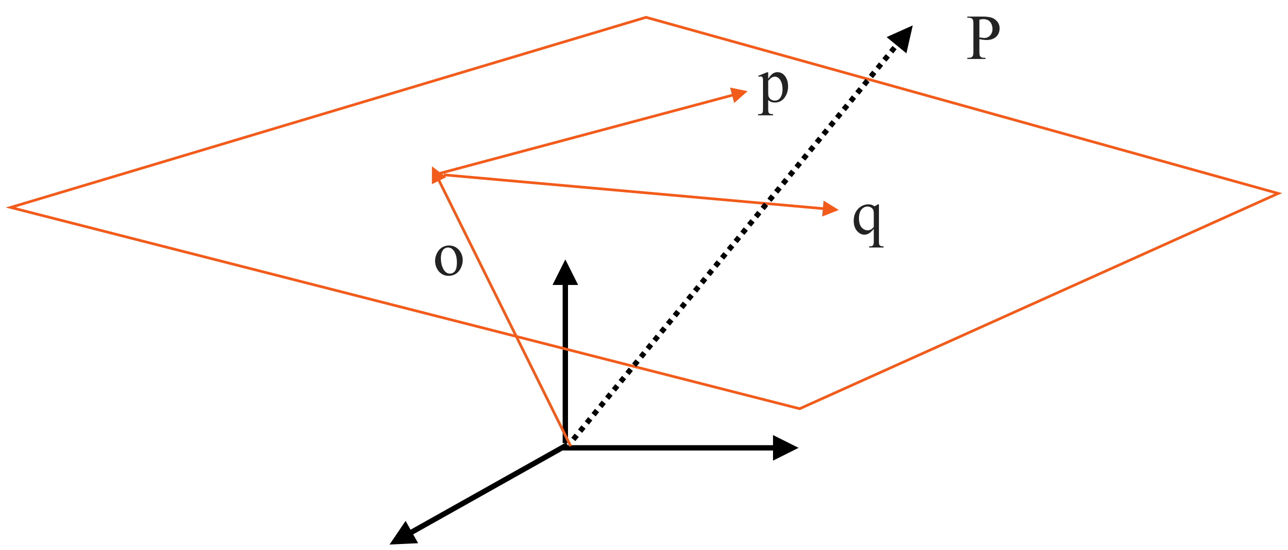


Homographies

- Consider the figure shown below which involves a coordinate frame and a plane defined by an origin point $\mathbf{o} \in \mathbb{R}^{\mathbf{3}}$ and two axis vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{\mathbf{3}}$. Note that every ray that intersects this plane can be associated with a ray through the origin as follows.

$$P = u\mathbf{p} + v\mathbf{q} + \mathbf{o} = \begin{pmatrix} \mathbf{p} & \mathbf{q} & \mathbf{o} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} \mathbf{p} & \mathbf{q} & \mathbf{o} \end{pmatrix}$$

- The matrix $H \in \mathbb{R}^{3\times3}$ is invertible as long as the plane is not degenerate and does not pass through the origin. It is referred to as a homography.

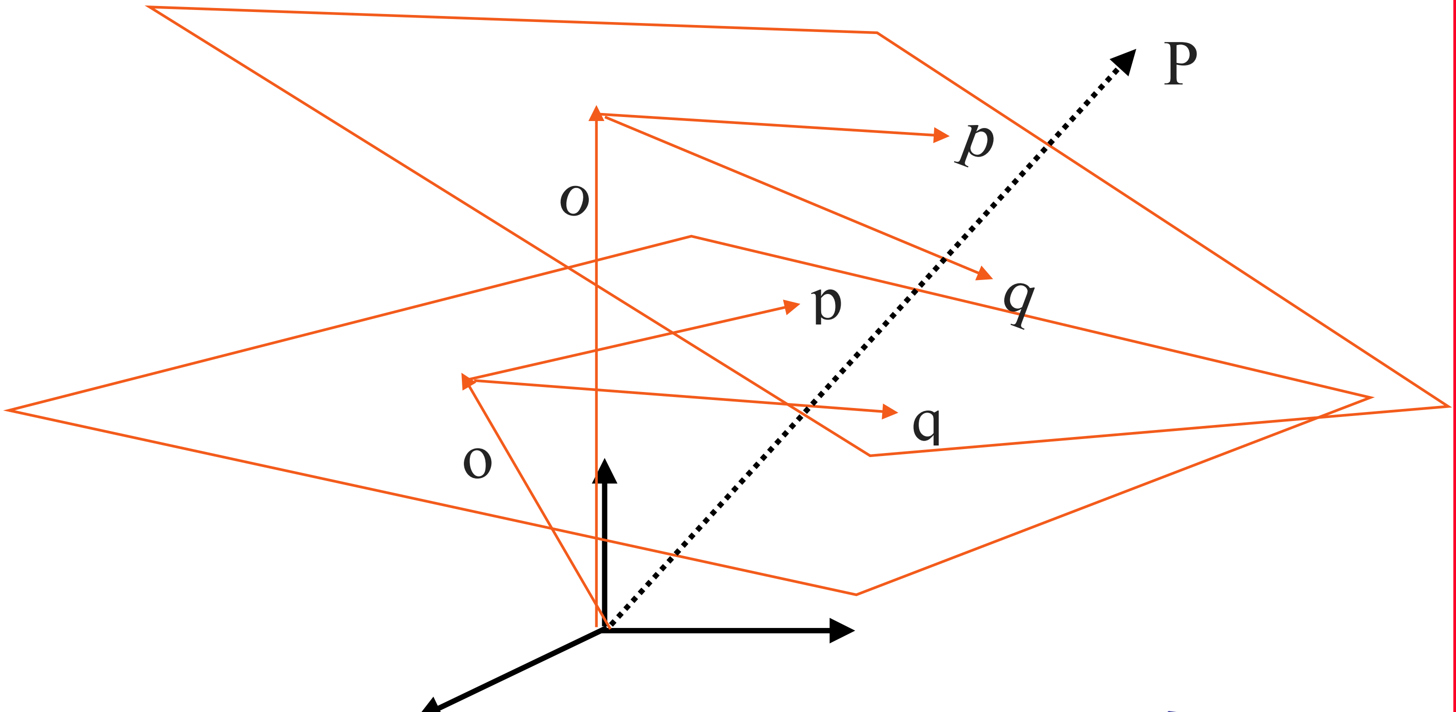


Rectifying Homographies

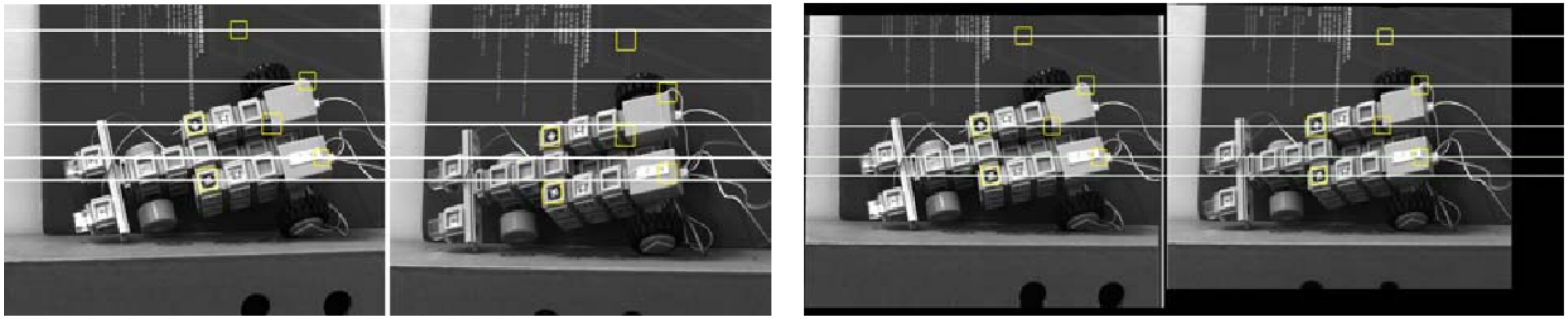
- Consider 2 planes defined by two homographies:

$$P \propto H_1 \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \propto H_2 \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \propto H_1^{-1} H_2 \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$$

- So we can transform point coordinates between the 2 image planes via a 3×3 matrix $H_{12} \propto H_1^{-1} H_2$



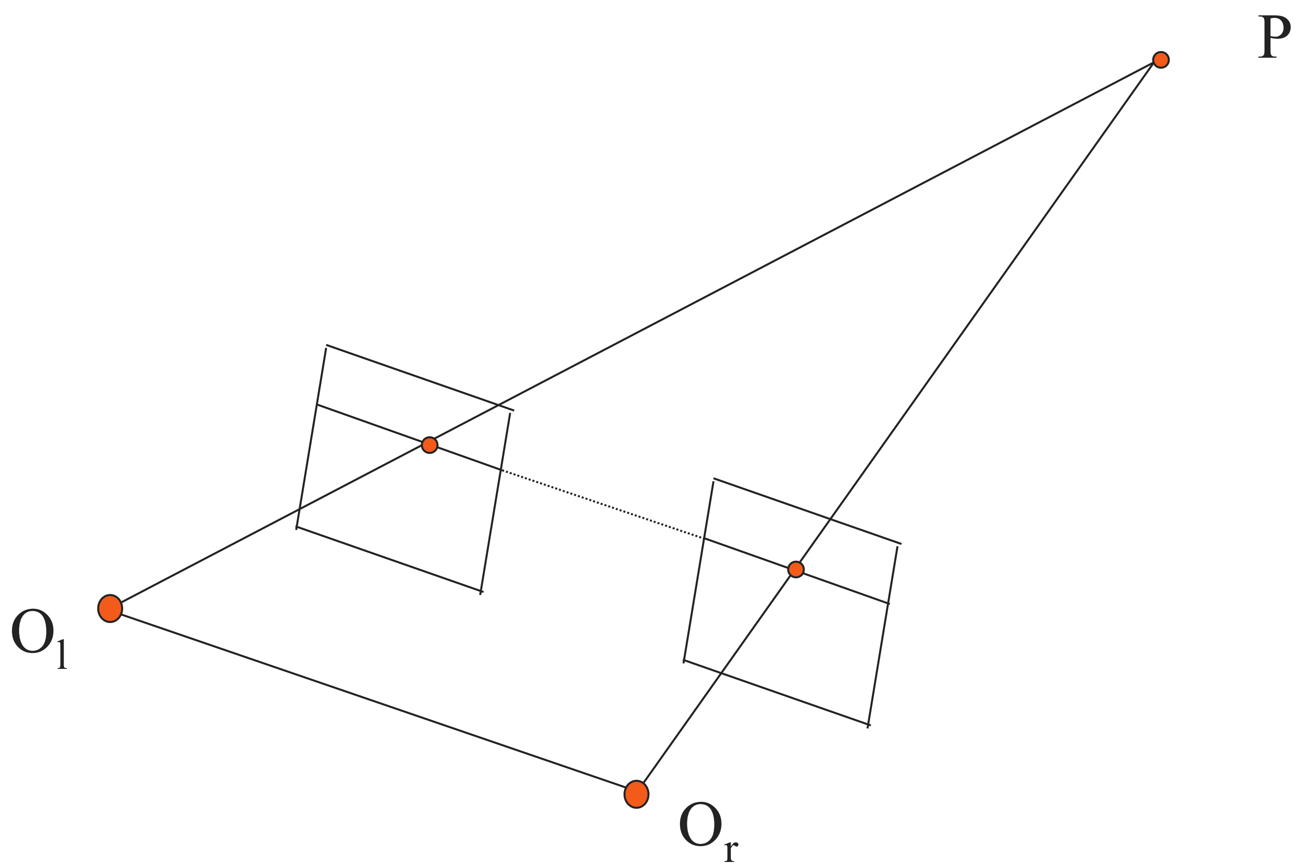
Rectifying Homographies



a) Before rectification

b) After rectification

Special Case

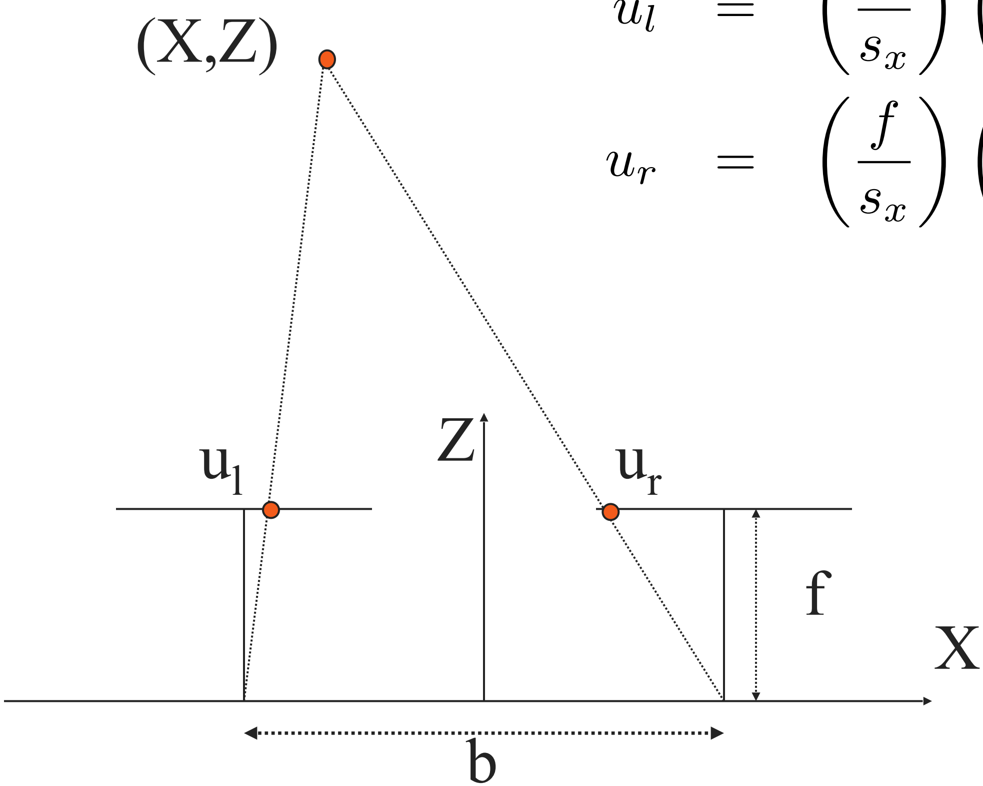


Special Case

- In the special case of the stereo setup shown above where the image planes are aligned with each other, the epipolar lines correspond to rows in the image
- That is the epipoles in both images are at infinity along the x axis.

Simplest Stereo System

- Consider two 1D cameras with identical intrinsic parameters



$$
\begin{aligned}
u_l &= \left(\frac{f}{s_x}\right)\left(\frac{X}{Z}\right) + c_x \\
u_r &= \left(\frac{f}{s_x}\right)\left(\frac{X-b}{Z}\right) + c_x
\end{aligned}
$$

Recovering Depth

- Given a correspondence between two locations in the two images the depth of a feature can be computed from the *disparity* between the two image locations
- Note the inverse relationship between disparity and depth which has important practical consequences

$$
\begin{aligned}
d &= (u_l - u_r) = \left(\frac{f}{s_x}\right)\left(\frac{b}{Z}\right) \\
Z &= \left(\frac{f}{s_x}\right)\left(\frac{b}{d}\right)
\end{aligned}
$$

Recovering Depth

- The inverse relationship between disparity and depth means that points that are very distant appear to have approximately zero disparity
- Another important consequence is that the error in the reconstructed depth increases as you go further away since a small difference in disparity translates to a large difference in depth as disparity approaches 0

$$
\begin{aligned}
d &= (u_l - u_r) = \left(\frac{f}{s_x}\right)\left(\frac{b}{Z}\right) \\
Z &= \left(\frac{f}{s_x}\right)\left(\frac{b}{d}\right)
\end{aligned}
$$

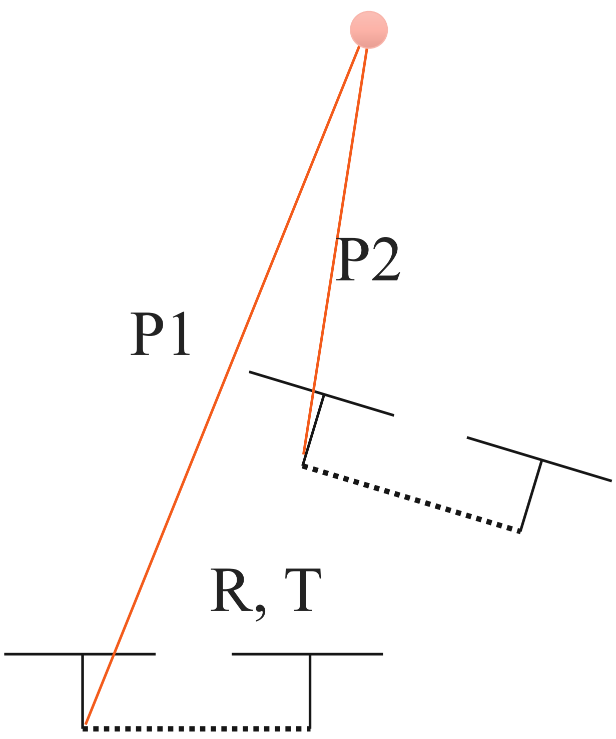
Stereo Odometry

Issues with recovering pose from E

- As we saw it is possible to recover the motion of a camera up to a scale factor from a series of feature correspondences
- There are two issues with this approach
  - You cannot recover scale from image measurements only
  - If the translation is 0 the essential matrix is degenerate and you cannot recover anything.

Stereo odometry

- An alternate approach is to equip the flying platform with a stereo rig with a known baseline then you can track stereo correspondences between frames and use these correspondences to recover the motion of the system



$$P_1 = \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}, \; P_2 = \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix}$$

$$P_1 = RP_2 + T$$

$$R \in SO(3), T \in \mathbb{R}^3$$

Stereo measurements

- To perform stereo odometry it will be convenient to normalize our measurements for image coordinates and disparity as follows where, f , cx, and cy are intrinsic parameters and b is the length of the stereo baseline

$$
\begin{aligned}
u &= f\left(\frac{X}{Z}\right) + c_x \\
v &= f\left(\frac{Y}{Z}\right) + c_y \\
d &= f\left(\frac{b}{Z}\right)
\end{aligned}
\qquad\qquad
\begin{aligned}
u' &= \left(\frac{X}{Z}\right) = (u - c_x)/f \\
v' &= \left(\frac{Y}{Z}\right) = (v - c_y)/f \\
d' &= \left(\frac{1}{Z}\right) = d/(fb)
\end{aligned}
$$

Stereo odometry

$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = R \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} + T$$

$$\left(\frac{Z_1}{Z_2}\right) \begin{pmatrix} X_1/Z_1 \\ Y_1/Z_1 \\ 1 \end{pmatrix} = R \begin{pmatrix} X_2/Z_2 \\ Y_2/Z_2 \\ 1 \end{pmatrix} + (1/Z_2)T$$

$$\left(\frac{Z_1}{Z_2}\right) \begin{pmatrix} u'_1 \\ v'_1 \\ 1 \end{pmatrix} = R \begin{pmatrix} u'_2 \\ v'_2 \\ 1 \end{pmatrix} + d'_2 T$$

Stereo odometry

$$\left(\frac{Z_1}{Z_2}\right)\begin{pmatrix} u_1' \\ v_1' \\ 1 \end{pmatrix} = R\begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T$$

$$\Rightarrow \quad \left(\frac{Z_1}{Z_2}\right) = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}\left[R\begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T\right]$$

Stereo odometry

- We can compute the discrepancy vector, delta, which should be zero when we have the correct R and T

$$\delta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \left[\left[R \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right] - \left(\frac{Z_1}{Z_2} \right) \begin{pmatrix} u_1' \\ v_1' \\ 1 \end{pmatrix} \right]$$

$$\delta = \begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} \left[R \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right]$$

Stereo odometry

- If we believe that the rotation is relatively small we can approximate R as follows
- This can be viewed as a linear constraint in the unknowns, w and T

$$R \approx (I + \hat{\omega}) R_0$$

$$\delta = \begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} \left[(I + \hat{\omega}) R_0 \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right] \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Stereo odometry

- This can be viewed as a linear constraint in the unknowns, w and T

$$\begin{pmatrix} 1 & 0 & -u'_1 \\ 0 & 1 & -v'_1 \end{pmatrix} [-\hat{y}\omega + d'_2 T] \approx - \begin{pmatrix} 1 & 0 & -u'_1 \\ 0 & 1 & -v'_1 \end{pmatrix} y$$

$$y = R_0 \begin{pmatrix} u'_2 \\ v'_2 \\ 1 \end{pmatrix}$$

$$[-\hat{y}\omega + d'_2 T] = \left[\begin{pmatrix} 0 & y_3 & -y_2 \\ -y_3 & 0 & y_1 \\ y_2 & -y_1 & 0 \end{pmatrix} \omega + \begin{pmatrix} d'_2 & 0 & 0 \\ 0 & d'_2 & 0 \\ 0 & 0 & d'_2 \end{pmatrix} T \right]$$

Stereo odometry

- Putting it all together we get:

$$\begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} [-\hat{y}\omega + d_2' T] \approx b$$

$$\begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} \begin{pmatrix} 0 & y_3 & -y_2 & d_2' & 0 & 0 \\ -y_3 & 0 & y_1 & 0 & d_2' & 0 \\ y_2 & -y_1 & 0 & 0 & 0 & d_2' \end{pmatrix} \begin{pmatrix} \omega \\ T \end{pmatrix} \approx b$$

$$y = R_0 \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix}, \quad b = -\begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} y$$

Stereo odometry

- All of the stereo measurement equations can be accumulated into a linear system of the form $Ax = b$ where $x = (w, T)$ is a vector composed of the six unknowns
- Since each stereo correspondences yields 2 independent linear constraints we can solve this given at least 3 stereo correspondences.
- Typically we have many more than 3 correspondences which leads to an over constrained linear system and we obtain the least squares solution which optimally combines all of the constraints.
- Note that this solution will work even when the translation is zero.

Detecting and Matching Features

Convolution

- Definition:

$$I_A(i,j) = I * A = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h,k) I(i-h, j-k)$$

Some facts about Convolution

- A linear operation
  - X*(aY + bZ) = a(X*Y) + b(X*Z)
- An associative operation
  - X*(Y*Z) = (X*Y)*Z
- Convolution in the spatial or time domain corresponds to multiplication in the frequency domain.

Common Kernels

- Two convolution kernels that are commonly used for noise reduction are
  - The mean kernel
  - The Gaussian Kernel

Computation

- Convolution is a fairly expensive operation requiring a fairly large number of computations on typical images.
- Many computer architectures provide specialized instructions for these kinds of operations egs MMX

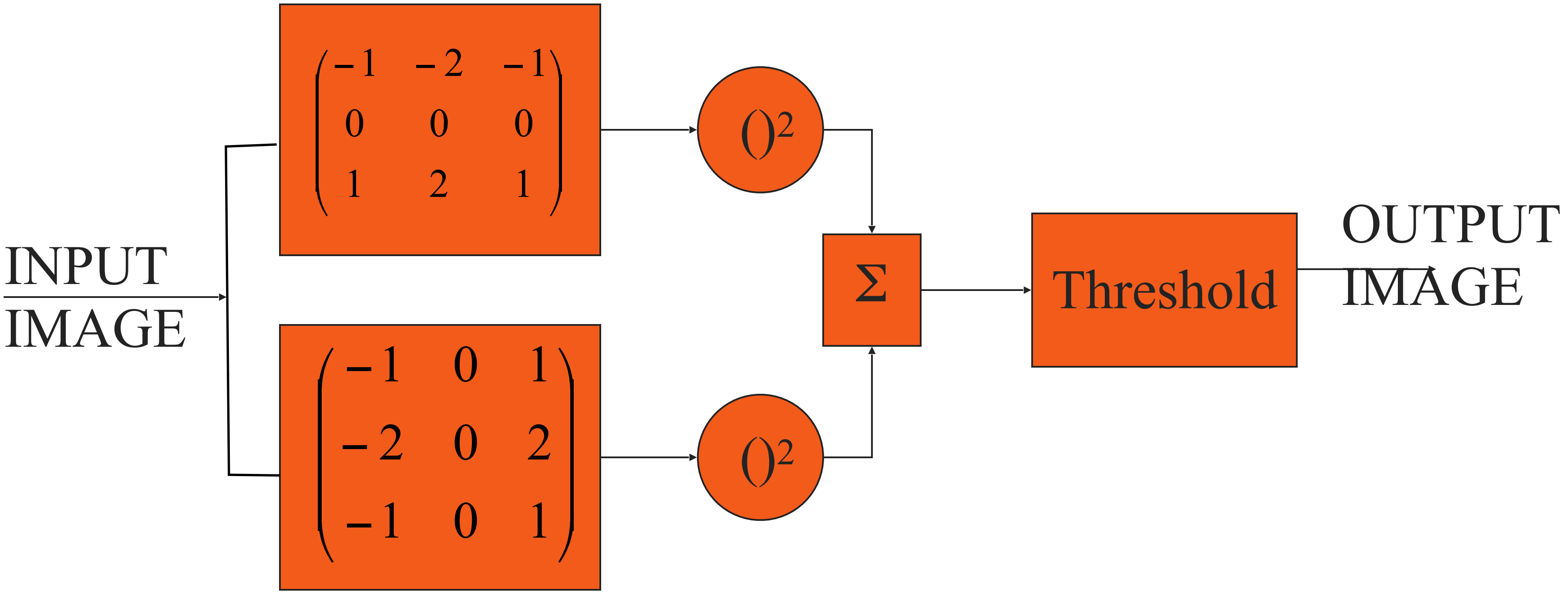
Edge Detection

- The goal in edge detection is to locate places in the image where there is a significant intensity change
- The thinking is that these edges are particularly salient locations in the image

Edge Detection Schemes

- The basic approach to edge detection is to compute “spatial derivatives” of the intensity image
- The act of taking spatial derivatives is usually approximated by convolution

Sobel



INPUT IMAGE
$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$
$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$
$()^2$
$()^2$
Σ
Threshold
OUTPUT IMAGE

Detecting Corners

- We are often interested in detecting point features in an image
- These features are usually defined as regions in the image where there is significant edge strength in two or more directions

Detecting corners

- If Ex and Ey denote the gradients of the intensity image, E(x,y), in the x and y directions then the behavior of the gradients in a region around a point can be obtained by considering the following matrix
- Note that the summation is performed over a window encircling the feature point

$$C = \sum \begin{pmatrix} E_x \\ E_y \end{pmatrix} \begin{pmatrix} E_x & E_y \end{pmatrix} = \sum \begin{pmatrix} E_x^{\,2} & E_x E_y \\ E_x E_y & E_y^{\,2} \end{pmatrix}$$

Examining the matrix

- One way to decide on the presence of a corner is to look at the eigenvalues of the 2 by 2 matrix C.
  - If the area is a region of constant intensity we would expect both eigenvalues to be small
  - If it contains a edge we expect one large eigenvalue and one small one
  - If it contains edges at two or more orientations we expect 2 large eigenvalues
  - One approach is to check whether the minimum eigenvalue is greater than a threshold, this is known as the Shi - Tomasi criterion

Finding corners

- One approach to finding corners is to find locations where the **smaller** eigenvalue is greater than some threshold
- We could also imagine considering the **ratio** of the two eigenvalues

Issues

- Localization - it can be difficult to precisely localize the corner in the intensity image
- Modeling - It can be helpful to have a model of the corners you are trying to find in order to detect and localize them more systematically

Basic Template Matching

- The most straightforward approach to finding a particular patter in the image is to look at every patch of pixels and to directly compare the intensity values in that patch to the intensity values in the search pattern.
- Consider two vectors of intensity values, x and y, One way to compare them is by simply computing the Sum of Squared Differences (SSD)

$$SSD(x,y) = \sum_i (x_i - y_i)^2$$

Feature Matching

- For every corner that we detect in the image we can compute a descriptor which is a vector that characterizes the region around the corner
- To match two features we compare their descriptors and the smaller the discrepancy the better the match
- Common descriptors include, SIFT, SURF, and BRIEF
- Modern feature detectors are much more sophisticated than simple template matching schemes. They can successfully match image patches that have been translated, rotated and moderately rescaled.

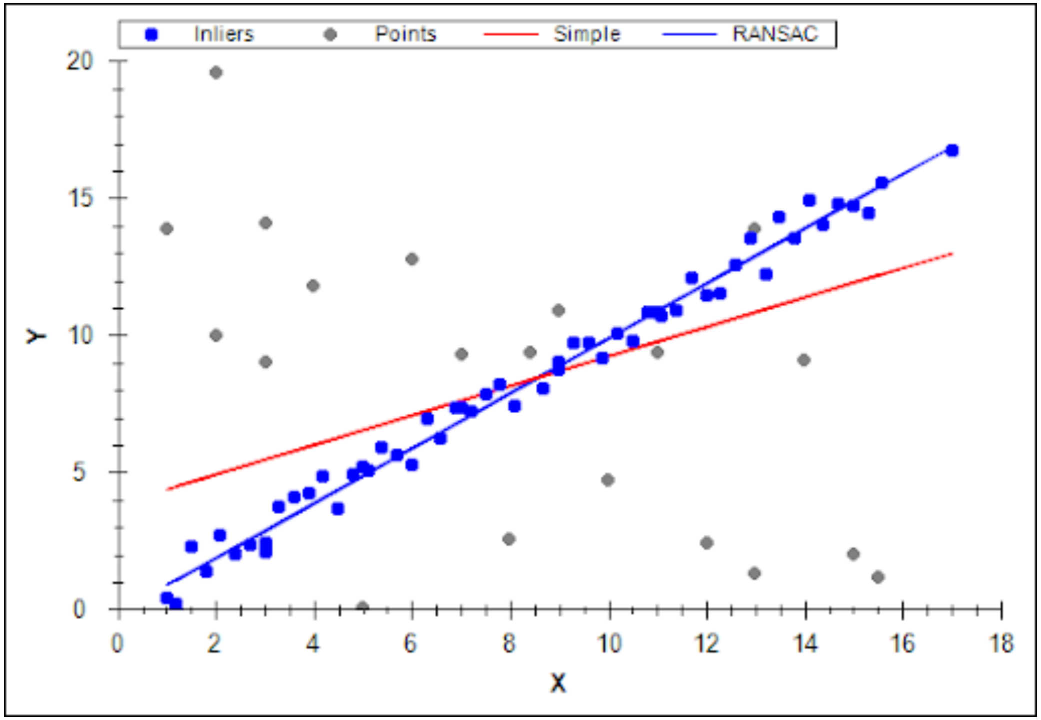
Matching Features Robustly - RANSAC

Robust Feature Matching

- So far we have assumed that all of the feature correspondences that we have been provided with are correct. That is they correctly match one point in one image to its correct match in the other.
- In practice the matching process is good but not perfect and mistakes can be made.
- Even a single incorrect measurement can cause problems for subsequent computational stages that typically use these correspondences in a least squares fitting procedure to recover the essential matrix, E, or the Rotation and translation

Example fitting a line

- The problem can be illustrated with the example shown below of fitting a line to some data points.
- When outliers are present the simple least squares solution can be very badly off



$$y_i \approx m x_i + c$$

$$\begin{pmatrix} x_i & 1 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} \approx y_i$$

Robust Feature Matching

- Our fitting problem can be formulated as a linear least squares problem where the individual constraints are accumulated into a single overconstrained system, Ax = b, and then solved using least squares

$$\begin{pmatrix} x_i & 1 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} \approx y_i$$

$$A \begin{pmatrix} m \\ c \end{pmatrix} \approx b, \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \; m \geq n$$

Standard Least squares

- The standard least squares formulation leads to the simple and elegant pseudo inverse solution. The problem is that this solution is very sensitive to outliers.

$$\min_x \|Ax - b\| \Rightarrow x = (A^T A)^{-1}(A^T b)$$

RANSAC

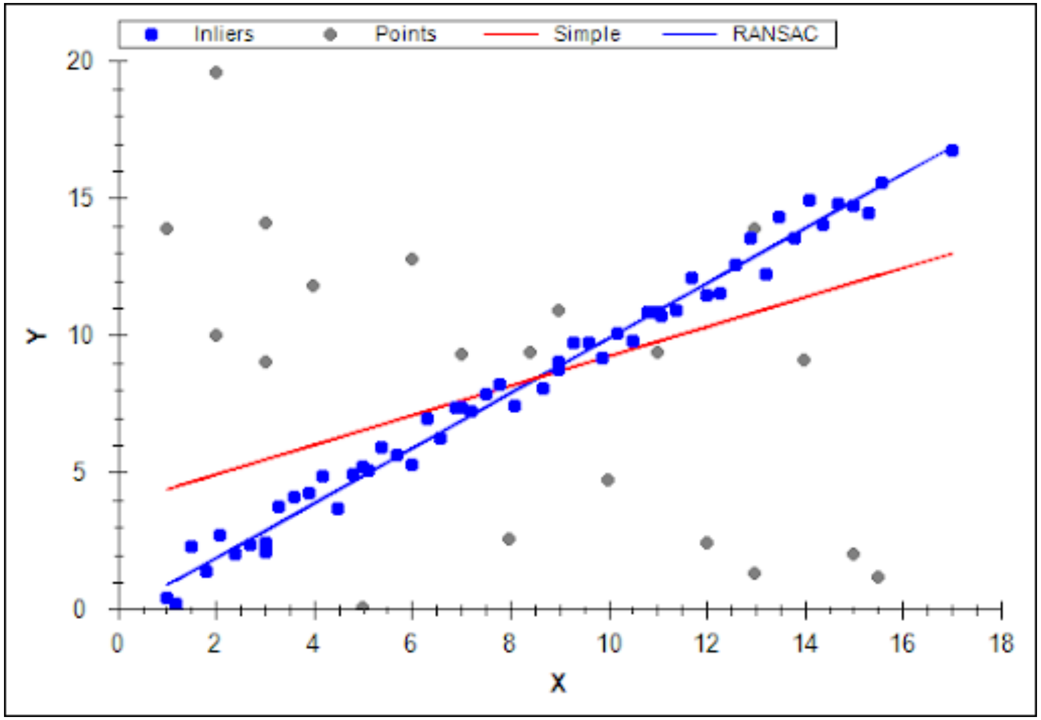
- One way to deal with outliers is by employing the Random Sample Consensus algorithm aka RANSAC.
- In the context of a linear system this involves selecting at random n rows from our linear system $Ax=b$. These n linear equations are just enough to solve for x exactly. We then see which other rows in the system roughly agree with the proposed solution.
- We retain the solution with the largest number of inliers.

RANSAC

- Given an overconstrained linear system $Ax \approx b$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $m > n$.
- Repeat k times
  - Select n equations from $Ax = b$ at random without replacement
  - Solve for x from this selection
  - Determine the inliers to this proposed solution by finding all of the rows for which $|A_i x - b_i| < \tau$ where τ is an appropriately chosen threshold
  - Retain the solution with the largest number of inliers and the corresponding inlier set
- Calculate the least squares fit to the largest inlier set discovered

Example fitting a line

- The problem can be illustrated with the example shown below of fitting a line to some data points.
- When outliers are present the simple least squares solution can be very badly off



$$y_i \approx m x_i + c$$

$$\begin{pmatrix} x_i & 1 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} \approx y_i$$

Stereo odometry

- We can compute the discrepancy vector, delta, which should be zero when we have the correct R and T
- Given a proposed solution we can determine which correspondences are inliers to that solution by considering the norm of delta

$$\delta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \left[\left[R \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right] - \left(\frac{Z_1}{Z_2} \right) \begin{pmatrix} u_1' \\ v_1' \\ 1 \end{pmatrix} \right]$$

$$\delta = \begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} \left[R \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right]$$

$$\delta = \begin{pmatrix} 1 & 0 & -u_1' \\ 0 & 1 & -v_1' \end{pmatrix} \left[(I + \hat{\omega}) R_0 \begin{pmatrix} u_2' \\ v_2' \\ 1 \end{pmatrix} + d_2' T \right]$$