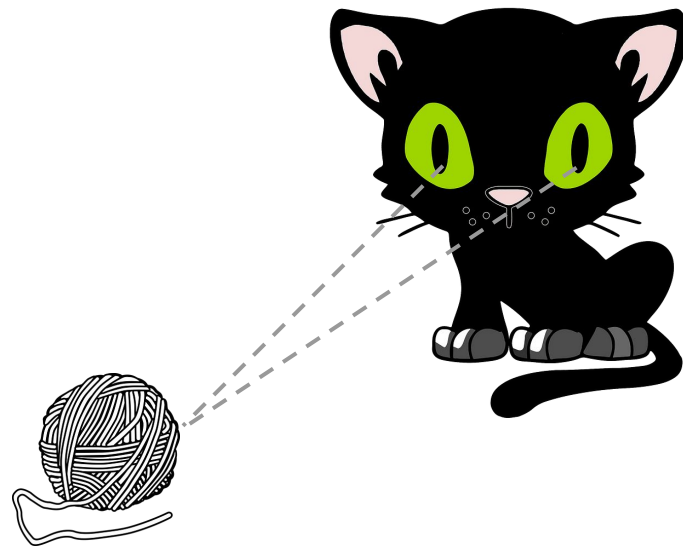


Computer Vision

Class 07



Class 07

01

Stereo Vision

02

Epipolar Geometry

03

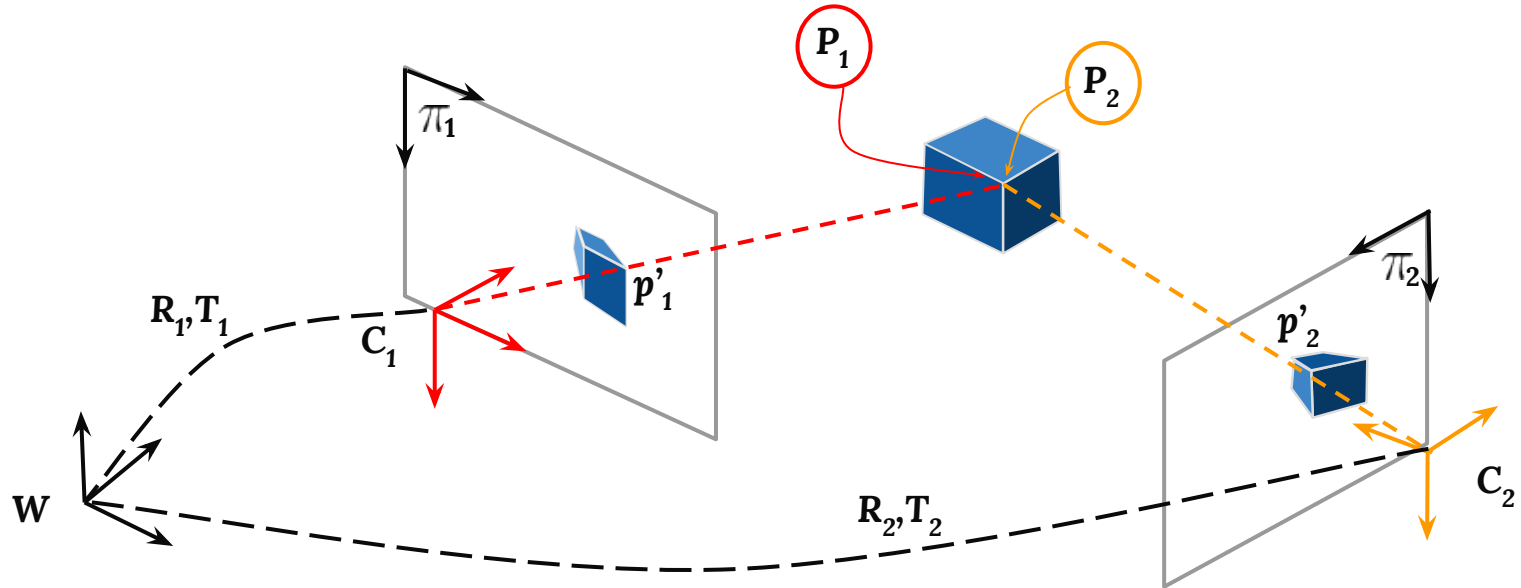
Triangulation

01

Stereo Vision

Stereo Vision

- Two images
- Matching of points between images
- Distance between two corresponding points in the left and right image: disparity
- Allow recovering 3D information



Stereo Vision

- Given two images, **recover the tridimensional structure of the scene and the relative pose of the cameras.**
- The vision system can be:
 - Calibrated (intrinsic and extrinsic parameters are known)
 - Partially calibrated (just intrinsic parameters are known)
 - Not calibrated (all parameters are unknown)

In this course, we will work with calibrated systems.

Pinhole Camera Model

Remember

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Parameters}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Canonical Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix}}_{\text{Extrinsic Parameters}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{3D point}}$$

Diagram illustrating the Pinhole Camera Model equation, showing the relationship between the image point, intrinsic parameters, canonical projection matrix, extrinsic parameters, and the 3D point.

$$\lambda p' = K \Pi_0 g(R, T) P_w \longrightarrow \boxed{\lambda p' = K[R, T]P_w}$$

Stereo System

Initially...

Let's work with the points in the image plane before converting to pixels.

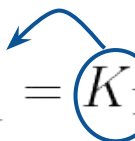
$$\lambda_1 p'_1 = K_1[R_1, T_1] P_w$$

$$\lambda_2 p'_2 = K_2[R_2, T_2] P_w$$

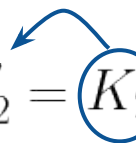
Stereo System

Initially...

Let's work with the points in the image plane before converting to pixels.

$$\lambda_1 p'_1 = \textcircled{K_1} [R_1, T_1] P_w$$


$$\lambda_1 K_1^{-1} p'_1 = [R_1, T_1] P_w$$

$$\lambda_2 p'_2 = \textcircled{K_2} [R_2, T_2] P_w$$


$$\lambda_2 K_2^{-1} p'_2 = [R_2, T_2] P_w$$

Stereo System

Initially...

Let's work with the points in the image plane before converting to pixels.

$$\lambda_1 p'_1 = K_1 [R_1, T_1] P_w$$

$$\lambda_1 \textcircled{K_1^{-1} p'_1} = [R_1, T_1] P_w$$

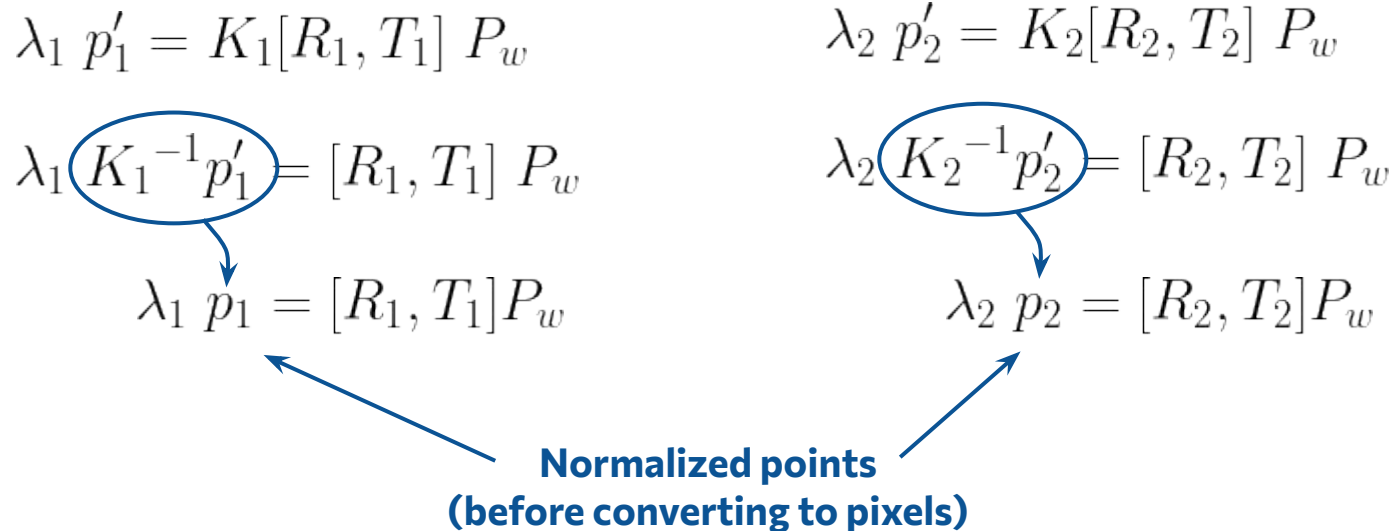
$$\lambda_1 p_1 = [R_1, T_1] P_w$$

$$\lambda_2 p'_2 = K_2 [R_2, T_2] P_w$$

$$\lambda_2 \textcircled{K_2^{-1} p'_2} = [R_2, T_2] P_w$$

$$\lambda_2 p_2 = [R_2, T_2] P_w$$

Normalized points
(before converting to pixels)



Stereo System

Also, let's represent the 3D point directly in each camera frame.

$$\lambda_1 p'_1 = K_1[R_1, T_1] P_w$$

$$\lambda_2 p'_2 = K_2[R_2, T_2] P_w$$

$$\lambda_1 K_1^{-1} p'_1 = [R_1, T_1] P_w$$

$$\lambda_2 K_2^{-1} p'_2 = [R_2, T_2] P_w$$

$$\lambda_1 p_1 = [R_1, T_1] P_w$$

$$\lambda_2 p_2 = [R_2, T_2] P_w$$

**3D point represented in
camera 1 and camera 2
reference frames**

$$\lambda_1 p_1 = P_1$$

$$\lambda_2 p_2 = P_2$$

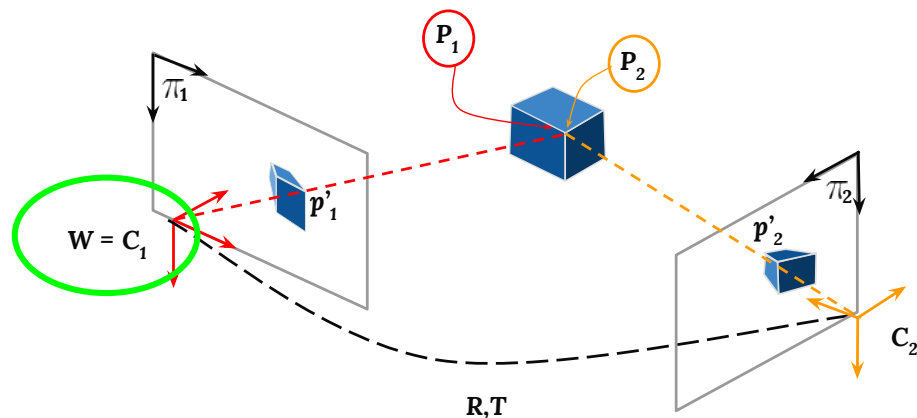
Stereo System

Now, let's consider the Camera 1 frame as the World frame.
Thus:

$$R_1 = I \quad \text{and} \quad T_1 = [0, 0, 0]^T$$

$$R_2 = R \quad \text{and} \quad T_2 = T$$

$$\lambda_1 p_1 = P_1 \quad \lambda_2 p_2 = P_2$$



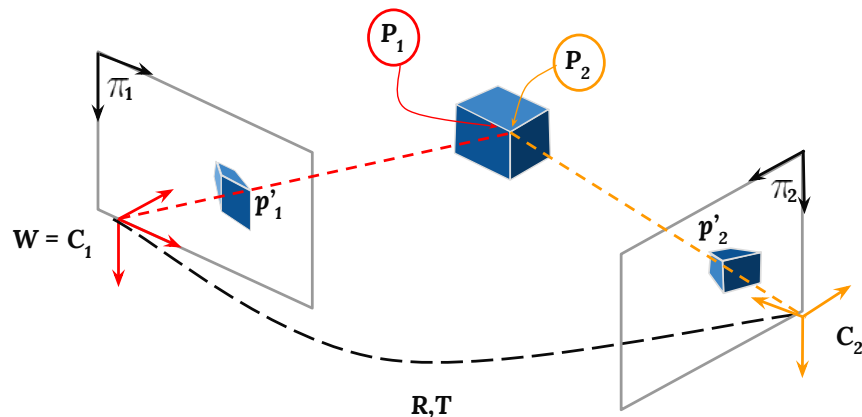
Stereo System

Now, let's consider the Camera 1 frame as the World frame.
Thus:

$$R_1 = I \quad \text{and} \quad T_1 = [0, 0, 0]^T$$

$$R_2 = R \quad \text{and} \quad T_2 = T$$

$$\lambda_1 p_1 = P_1 \quad \lambda_2 p_2 = P_2$$



And write down the relationship
between the 3D point coordinates in
Camera 1 and Camera 2 reference
frames.

$$\longrightarrow P_2 = RP_1 + T$$

Stereo System

Now, let's consider the Camera 1 frame as the World frame.
Thus:

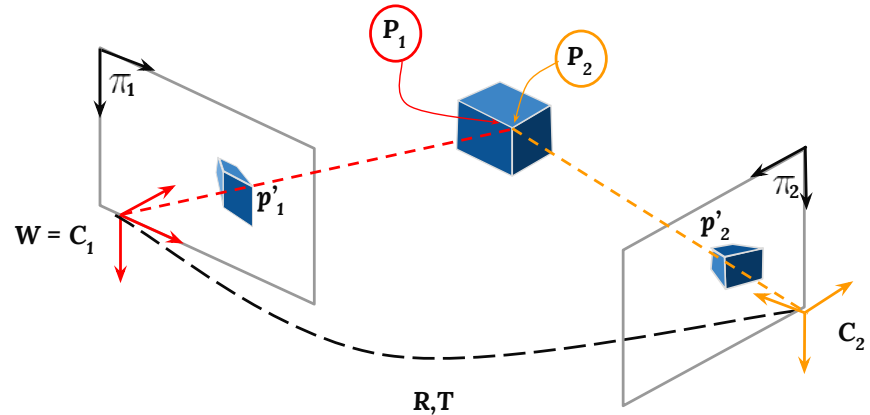
$$R_1 = I \quad \text{and} \quad T_1 = [0, 0, 0]^T$$

$$R_2 = R \quad \text{and} \quad T_2 = T$$

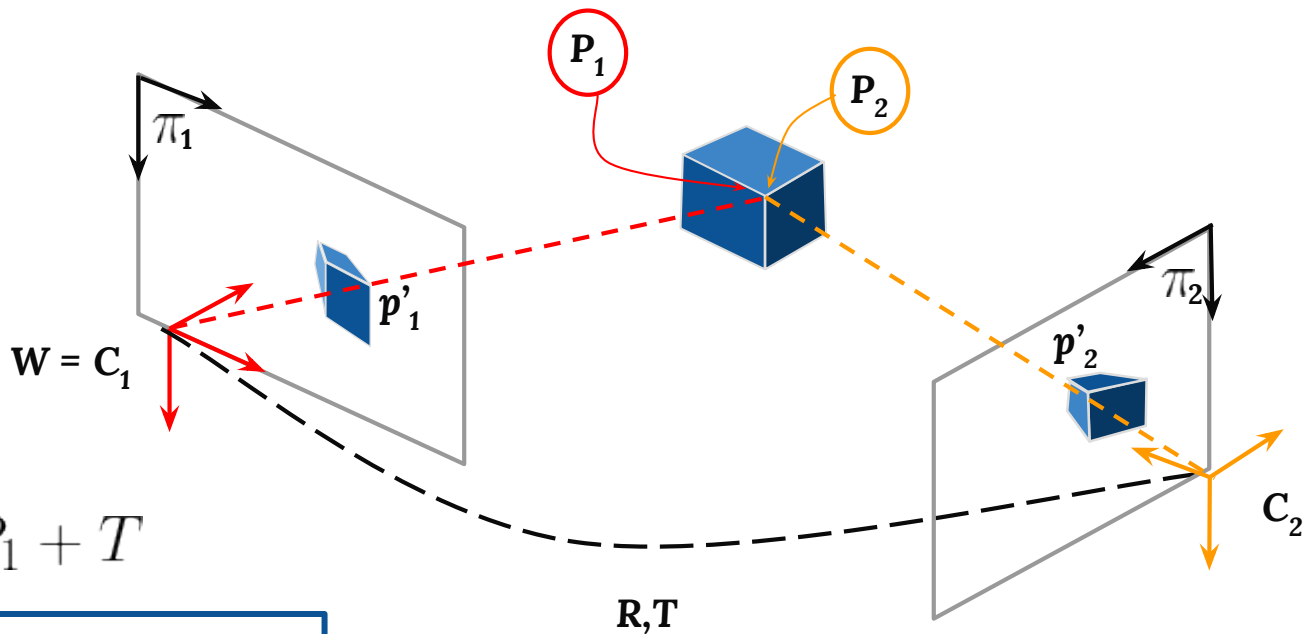
$$\lambda_1 p_1 = P_1 \quad \lambda_2 p_2 = P_2$$

And write down the relationship
between the 3D point coordinates in
Camera 1 and Camera 2 reference
frames.

$$\begin{aligned} &\longrightarrow P_2 = R P_1 + T \\ &\lambda_2 p_2 = \lambda_1 R p_1 + T \end{aligned}$$



Stereo System



$$P_2 = RP_1 + T$$

$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$

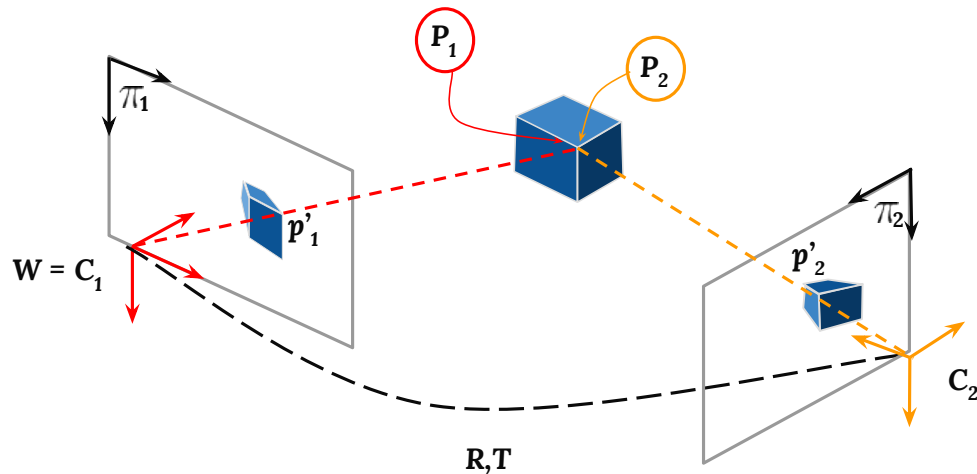
02

Epipolar Geometry

Epipolar Geometry

From the relation between the projections on the two images...

$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$



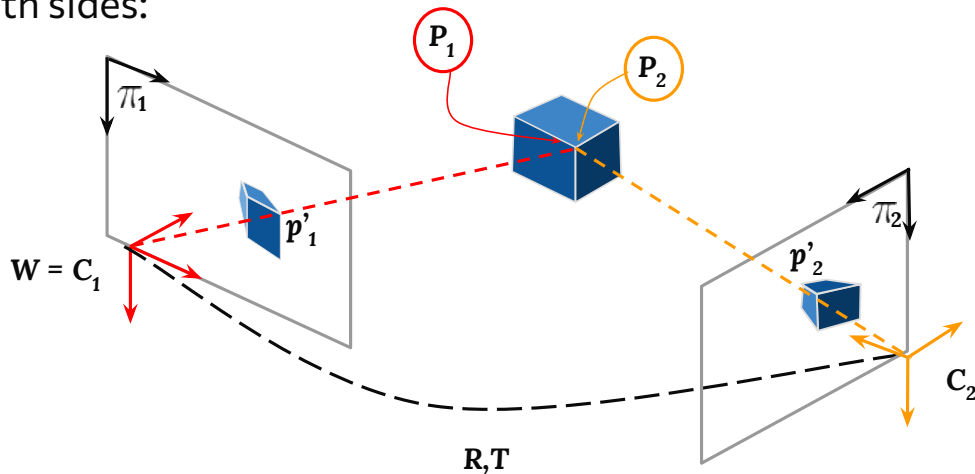
Epipolar Geometry

From the relation between the projections on the two images...

$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$

Performing the cross product with T on both sides:

$$\lambda_2 \hat{T} p_2 = \lambda_1 \hat{T} R p_1 + \hat{T} T$$



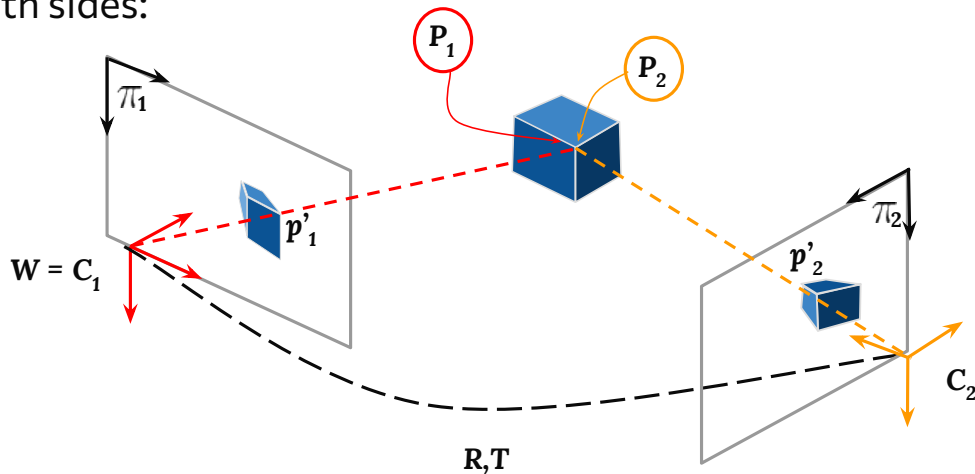
Epipolar Geometry

From the relation between the projections on the two images...

$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$

Performing the cross product with T on both sides:

$$\lambda_2 \hat{T} p_2 = \lambda_1 \hat{T} R p_1 + \cancel{\hat{T} T} \mathbf{0}$$



Epipolar Geometry

Now, from the relation between the points of the two images...

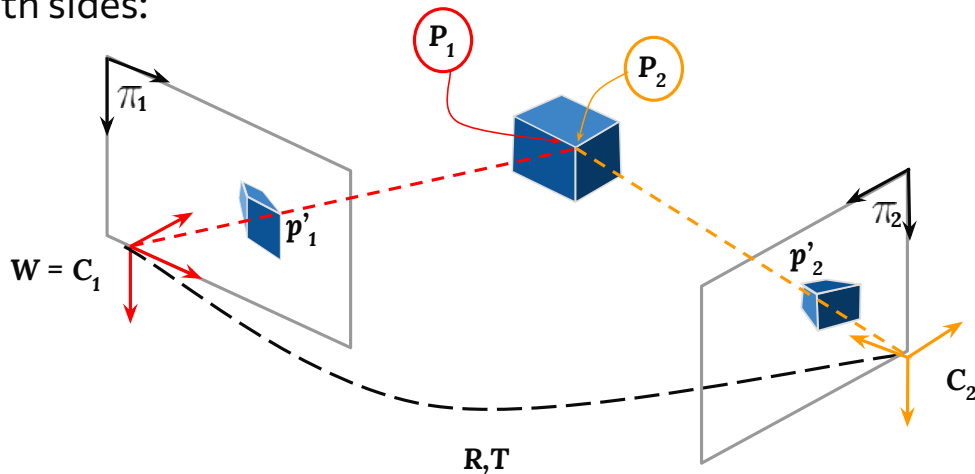
$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$

Performing the cross product with T on both sides:

$$\lambda_2 \hat{T} p_2 = \lambda_1 \hat{T} R p_1 + \cancel{\hat{T} T} \mathbf{0}$$

Performing the inner product with p_2 :

$$\lambda_2 p_2^T \hat{T} p_2 = \lambda_1 p_2^T \hat{T} R p_1$$



Epipolar Geometry

Now, from the relation between the points of the two images...

$$\lambda_2 p_2 = \lambda_1 R p_1 + T$$

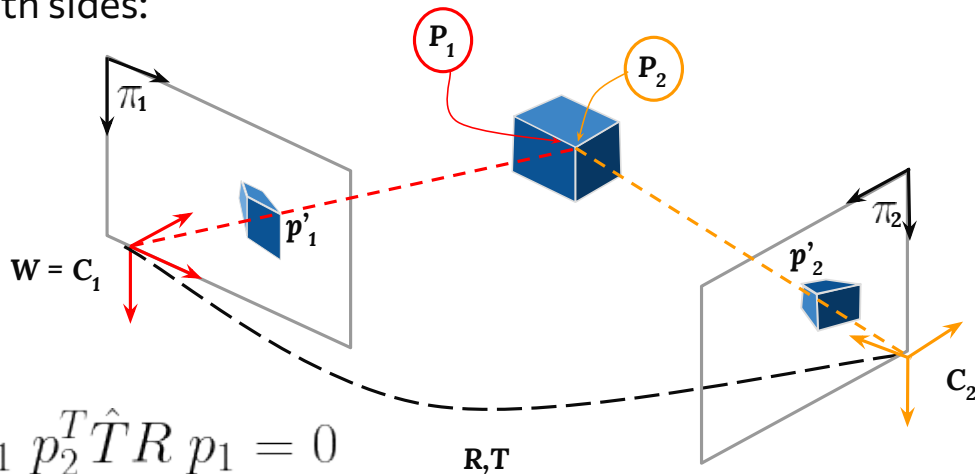
Performing the cross product with T on both sides:

$$\lambda_2 \hat{T} p_2 = \lambda_1 \hat{T} R p_1 + \cancel{\hat{T} T}^0$$

Performing the inner product with p_2 :

$$\cancel{\lambda_2 p_2^T \hat{T} p_2}^0 = \lambda_1 p_2^T \hat{T} R p_1$$

$$\lambda_1 p_2^T \hat{T} R p_1 = 0$$



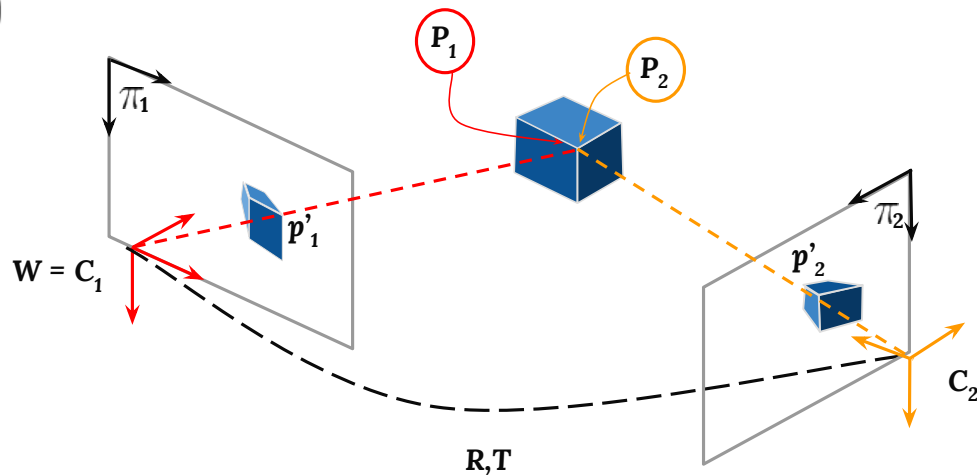
Epipolar Geometry

We can disconsider the scale factor, since the equation equals zero, and we end up with:

$$\lambda_1 p_2^T \hat{T} R p_1 = 0$$

↙

$$p_2^T \hat{T} R p_1 = 0$$



Epipolar Geometry

We can disconsider the scale factor, since the equation equals zero, and we end up with:

$$\lambda_1 p_2^T \hat{T} R p_1 = 0$$

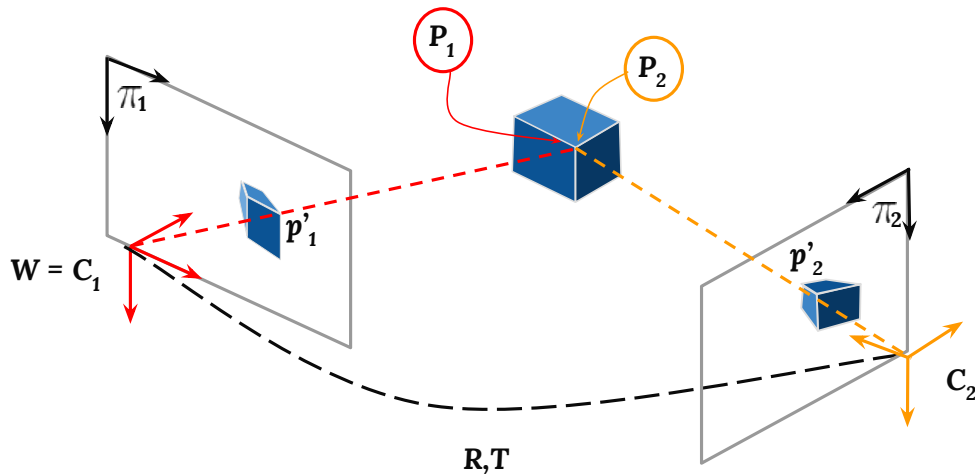
$$p_2^T \hat{T} R p_1 = 0$$

Essential Matrix

$$E = \hat{T} R$$

Epipolar Constraint

$$p_2^T E p_1 = 0$$

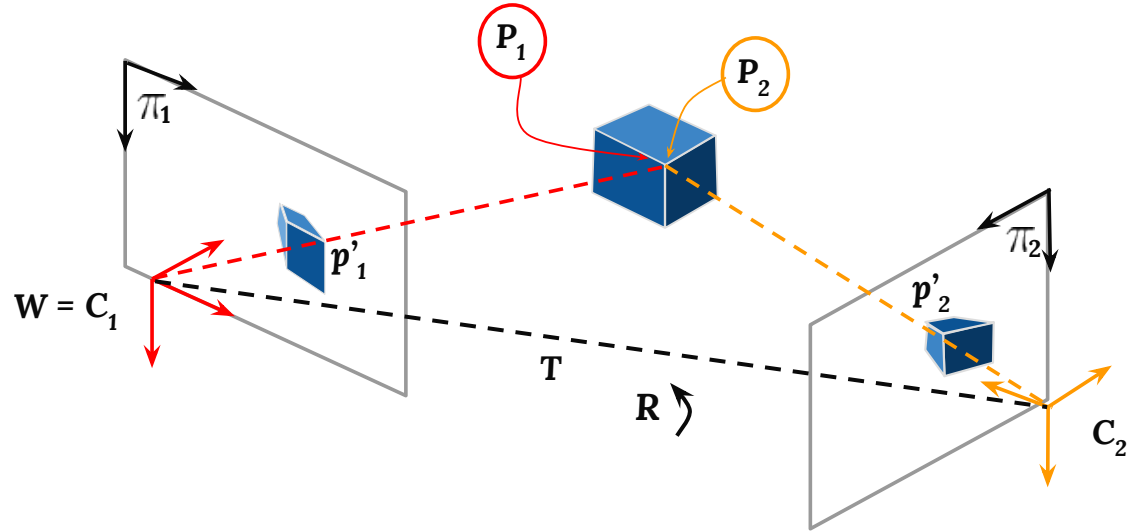


Epipolar Constraint

Let's understand what is epipolar geometry and what is the epipolar constraint.

$$p_2^T E p_1 = 0$$

$$p_2^T \hat{T} R p_1 = 0$$



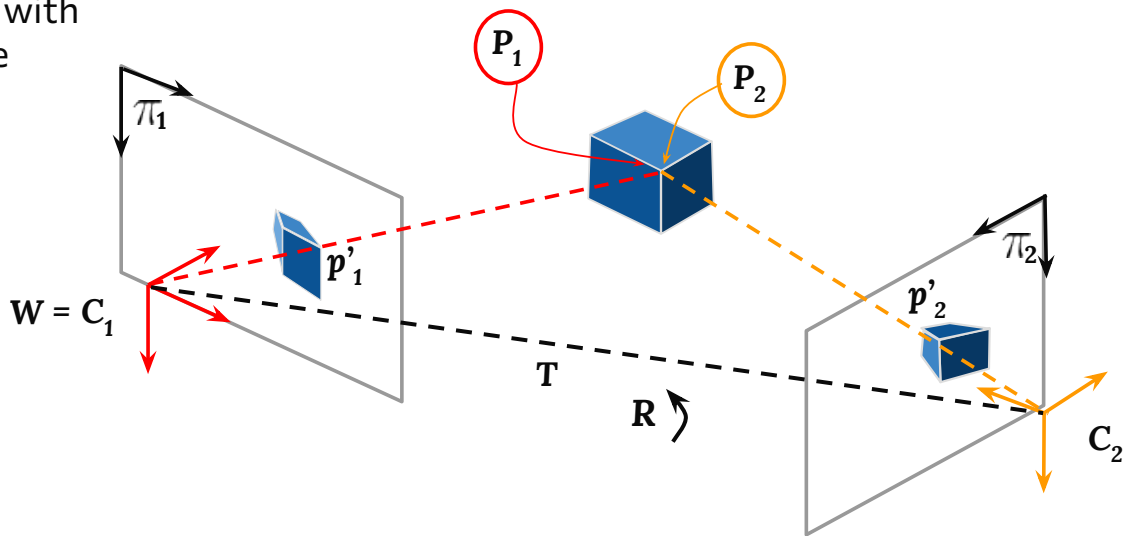
Epipolar Constraint

Let's understand what is epipolar geometry and what is the epipolar constraint.

$$p_2^T E p_1 = 0$$

Representation of p_1 with
the orientation of the
reference frame C_2

$$p_2^T \hat{T} R p_1 = 0$$



Epipolar Constraint

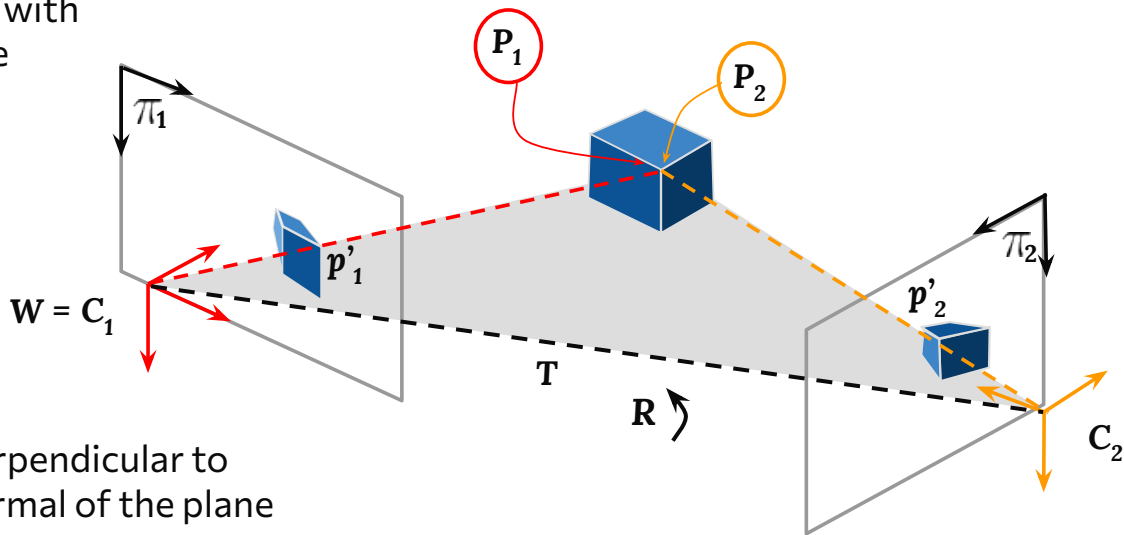
Let's understand what is epipolar geometry and what is the epipolar constraint.

$$p_2^T E p_1 = 0$$

Representation of p_1 with the orientation of the reference frame C_2

$$p_2^T \hat{T} R p_1 = 0$$

Vector perpendicular to $R p_1 \rightarrow$ normal of the plane



Epipolar Constraint

Let's understand what is epipolar geometry and what is the epipolar constraint.

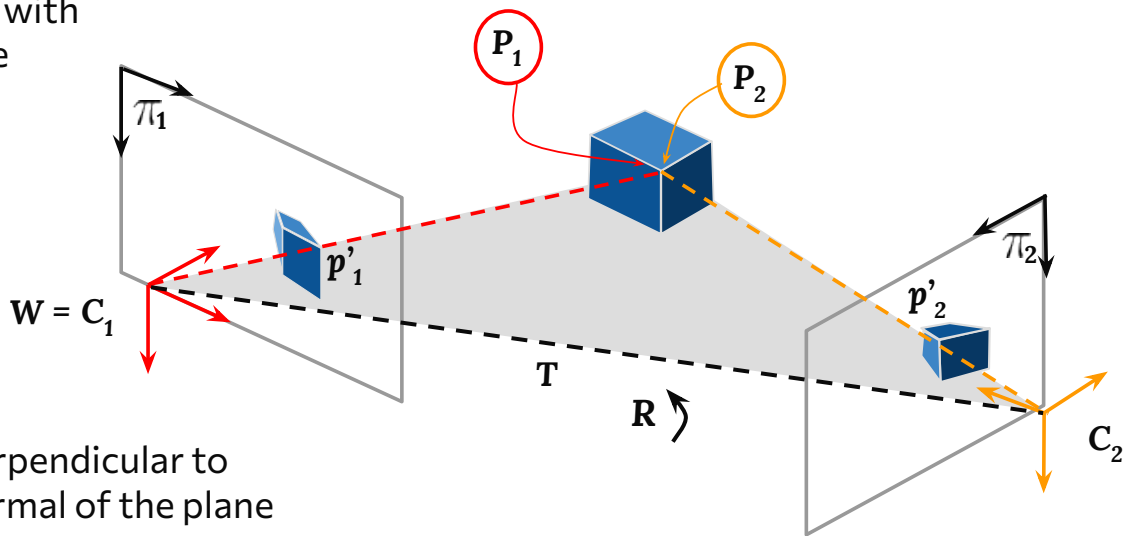
$$p_2^T E p_1 = 0$$

Representation of p_1 with the orientation of the reference frame C_2

$$p_2^T (\hat{T} R p_1) = 0$$

Since p_2 is on the plane, the inner product is zero

Vector perpendicular to $R p_1 \rightarrow$ normal of the plane



Epipolar Constraint

Thus:

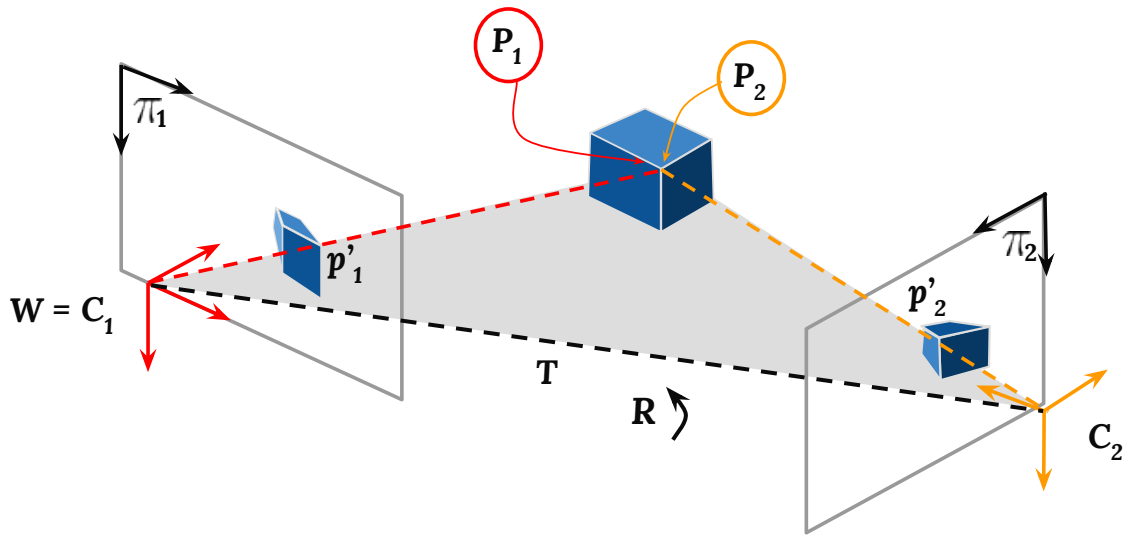
- The 3D point
- Its projections on the image plane
- The centers of projection of each camera reference frame

They all lie on the same plane.

The Epipolar Plane

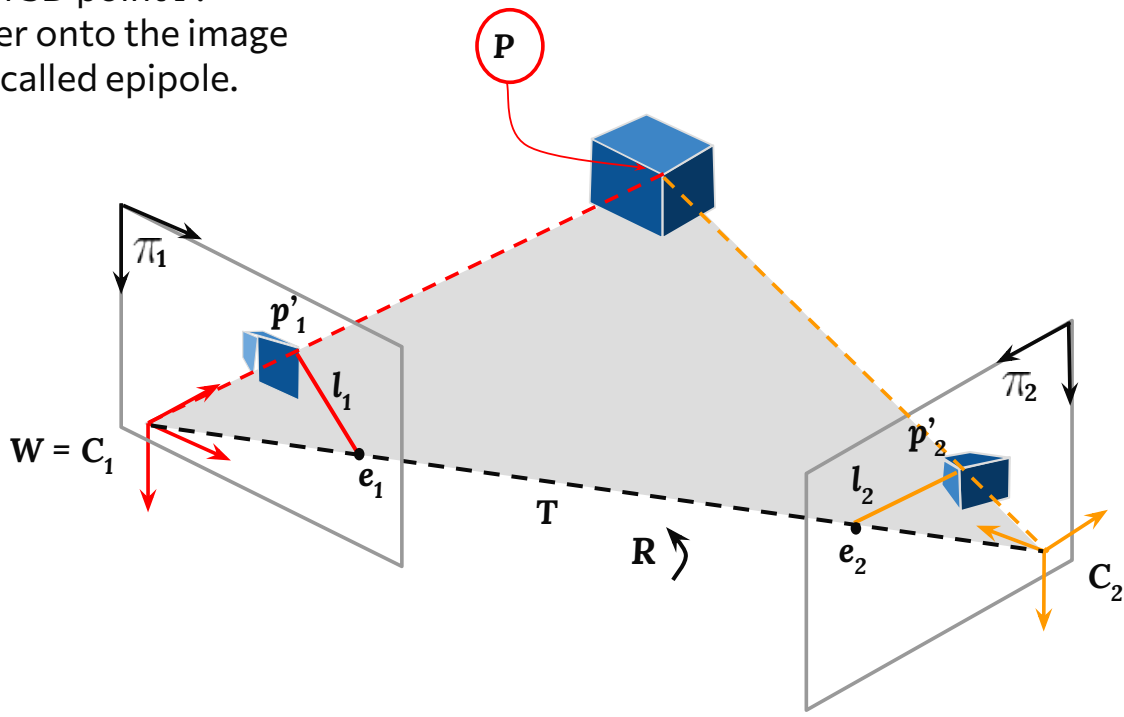
$$p_2^T E p_1 = 0$$

$$p_2^T \hat{T} R p_1 = 0$$

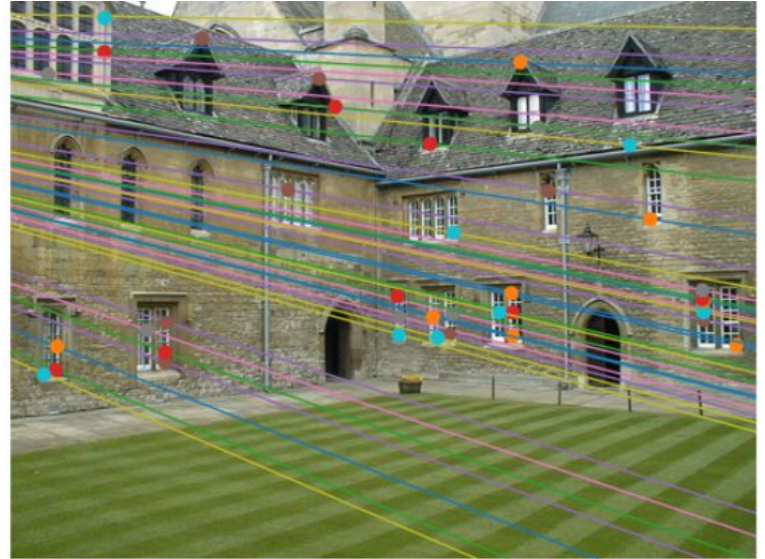
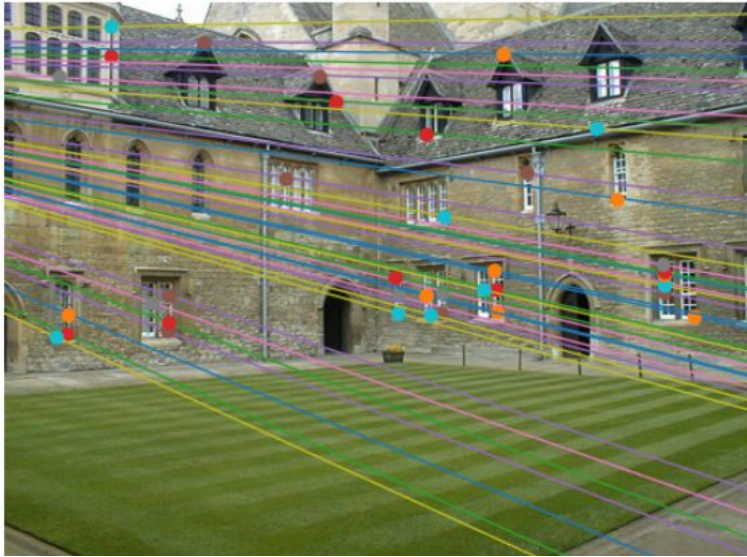


Epipolar Geometry

- The plane (C_1, C_2, P) is called Epipolar Plane associated with the point P .
- There is one epipolar plane for each 3D point P .
- The projection of one camera center onto the image plane of the other camera frame is called epipole.
- The intersection of the epipolar plane of P with one image plane is a line, which is called epipolar line.
- There is an epipolar line on each image, l_1 and l_2 .
- The correspondent points p_1 and p_2 lie on the epipolar lines.

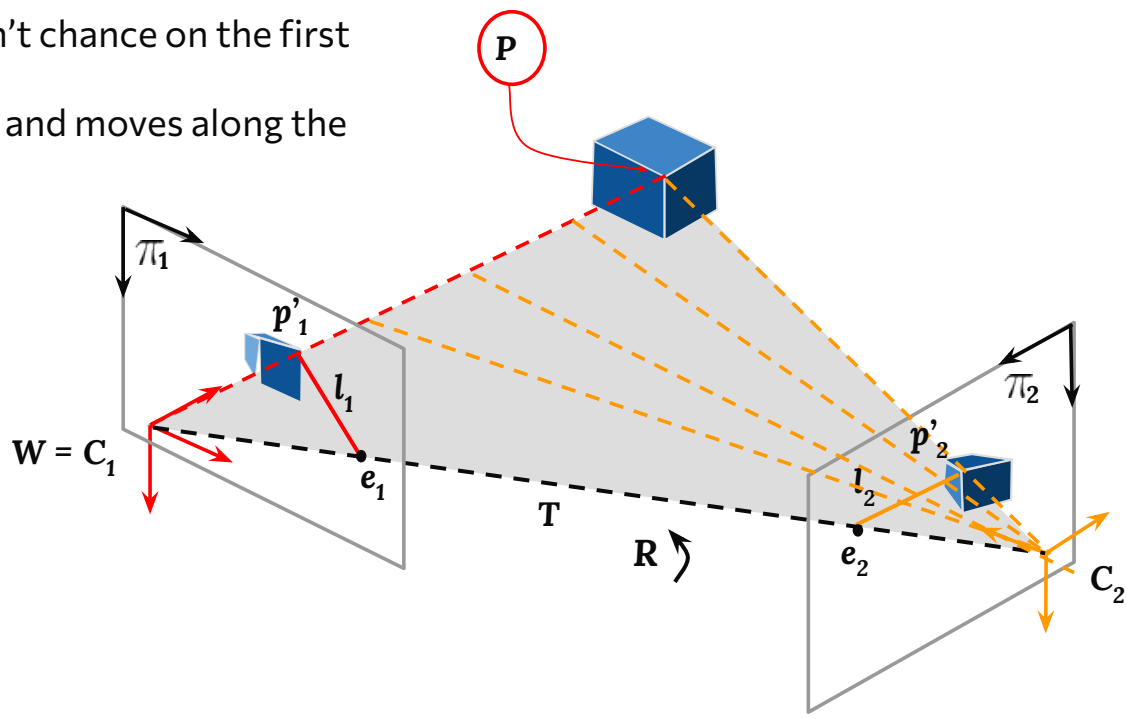


Epipolar Geometry



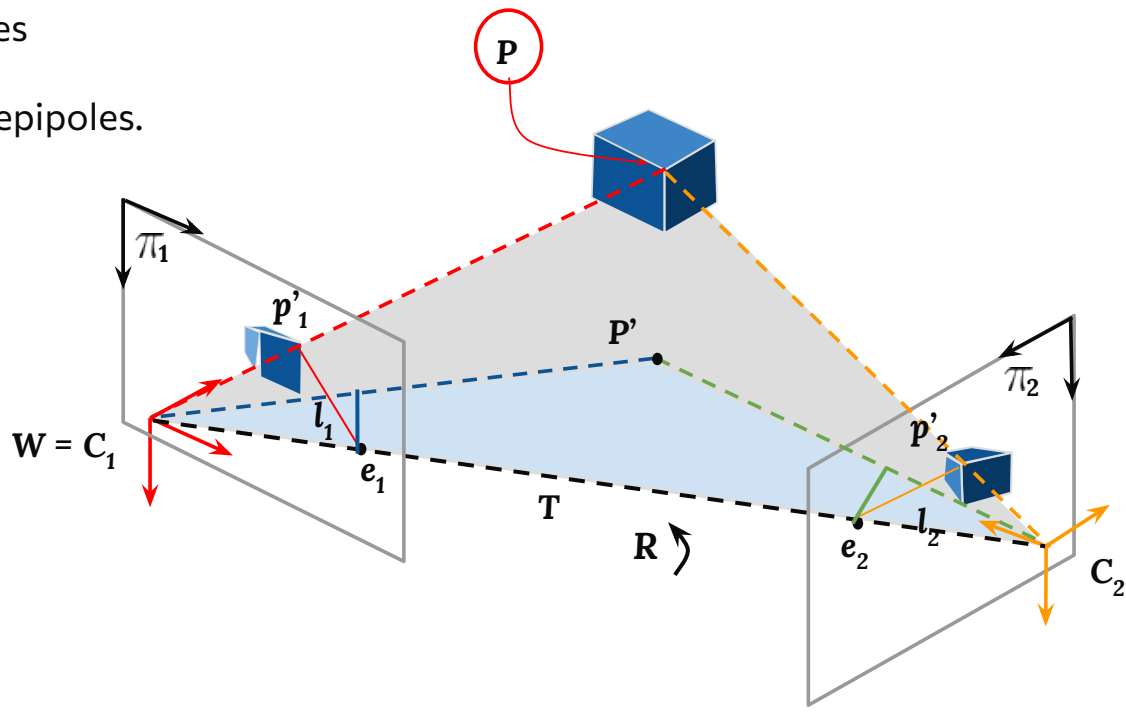
Epipolar Geometry

- If the 3D point moves along the ray (C_1, P) , changing its position:
 - although its projection doesn't change on the first image
 - it changes on the second one and moves along the epipolar line l_2 on that image.



Epipolar Geometry

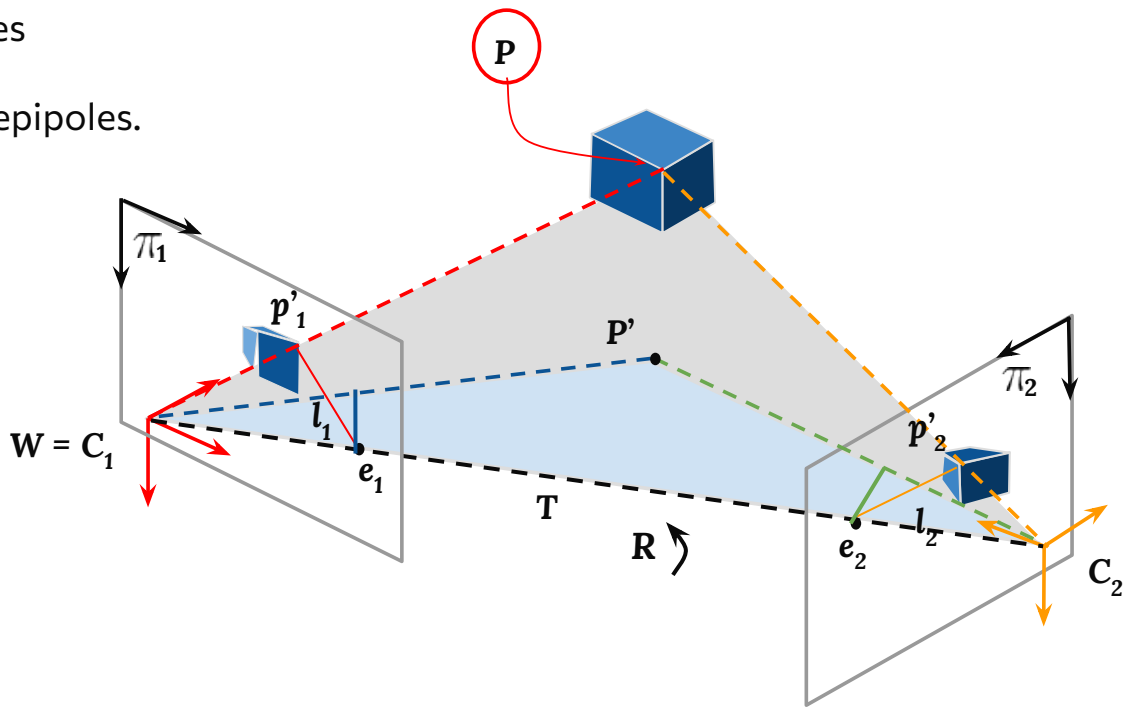
- If the 3D point does not move along the ray (C_1, P):
 - there is another Epipolar Plane
 - that defines new epipolar lines
- All epipolar lines pass through the epipoles.



Epipolar Geometry

- If the 3D point does not move along the ray (C_1, P):
 - there is another Epipolar Plane
 - that defines new epipolar lines
- All epipolar lines pass through the epipoles.

Because the projections change in the images and because there is a disparity between them, we can recover 3D information from a stereo system.



Essential Matrix and Fundamental Matrix

Essential Matrix

$$E = \hat{T} R$$

$$p_2^T \hat{T} R p_1 = 0$$

$$p_2^T E p_1 = 0$$

Relates **normalized points**
from left and right images.

Essential Matrix and Fundamental Matrix

Essential Matrix

$$E = \hat{T}R$$

$$p_2^T \hat{T}R p_1 = 0$$

$$p_2^T E p_1 = 0$$

Relates **normalized points**
from left and right images.

$$p_1 = K_1^{-1} p'_1$$

Normalized

Pixel

$$p_2 = K_2^{-1} p'_2$$

$$p_2^T \hat{T}R p_1 = 0$$


Essential Matrix and Fundamental Matrix

Essential Matrix

$$E = \hat{T} R$$

$$p_2^T \hat{T} R p_1 = 0$$

$$p_2^T E p_1 = 0$$

Relates **normalized points**
from left and right images.

$$\begin{array}{ccc} p_1 = K_1^{-1} p'_1 & & p'_1 \\ \uparrow & & \uparrow \\ \text{Normalized} & & \text{Pixel} \\ \downarrow & & \downarrow \\ p_2 = K_2^{-1} p'_2 & & p'_2 \end{array}$$

$$p_2^T \hat{T} R p_1 = 0$$

$$(K_2^{-1} p'_2)^T \hat{T} R K_1^{-1} p'_1 = 0$$

$$p_2'^T K_2^{-T} \hat{T} R K_1^{-1} p'_1 = 0$$

Essential Matrix and Fundamental Matrix

Essential Matrix

$$E = \hat{T}R$$

$$p_2^T \hat{T}R p_1 = 0$$

$$p_2^T E p_1 = 0$$

Relates **normalized points** from left and right images.

$$\begin{array}{ccc} p_1 = K_1^{-1} p'_1 & & \\ \uparrow \text{Normalized} & & \uparrow \text{Pixel} \\ p_2 = K_2^{-1} p'_2 & & \end{array}$$

$$p_2^T \hat{T}R p_1 = 0$$

$$(K_2^{-1} p'_2)^T \hat{T}R K_1^{-1} p'_1 = 0$$

$$p_2'^T \underbrace{K_2^{-T} \hat{T}R K_1^{-1}}_{\text{Fundamental Matrix}} p'_1 = 0$$

Fundamental Matrix

Relates **points in pixels** from left and right images.

$$F = K_2^{-T} \hat{T}R K_1^{-1}$$

$$p_2'^T F p'_1 = 0$$

Essential Matrix and Fundamental Matrix

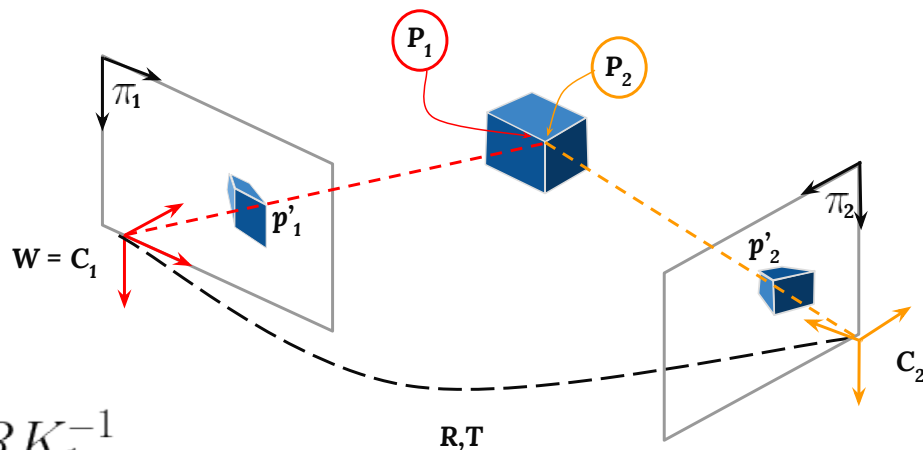
Essential Matrix

Relates **normalized points** from left and right images.

$$\underline{E = \hat{T}R}$$

$$p_2^T \hat{T}R p_1 = 0$$

$$p_2^T E p_1 = 0$$



Fundamental Matrix

Relates **points in pixels** from left and right images.

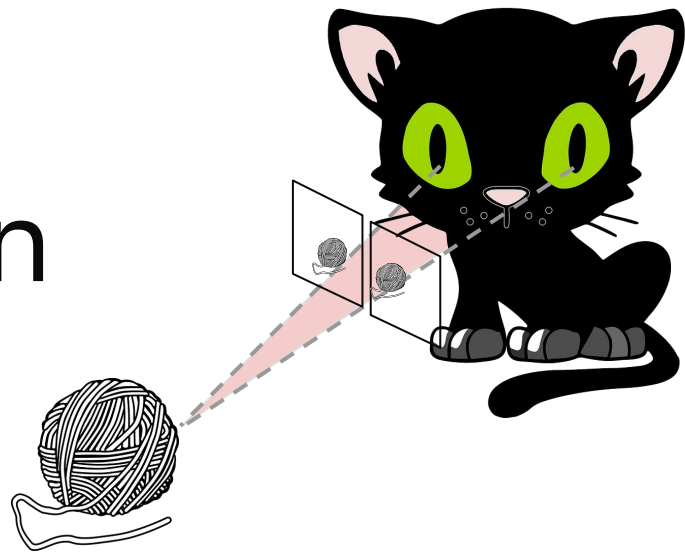
$$\underline{F = K_2^{-T} \hat{T}R K_1^{-1}}$$

$$p_2'^T K_2^{-T} \hat{T}R K_1^{-1} p_1' = 0$$

$$p_2'^T F p_1' = 0$$

03

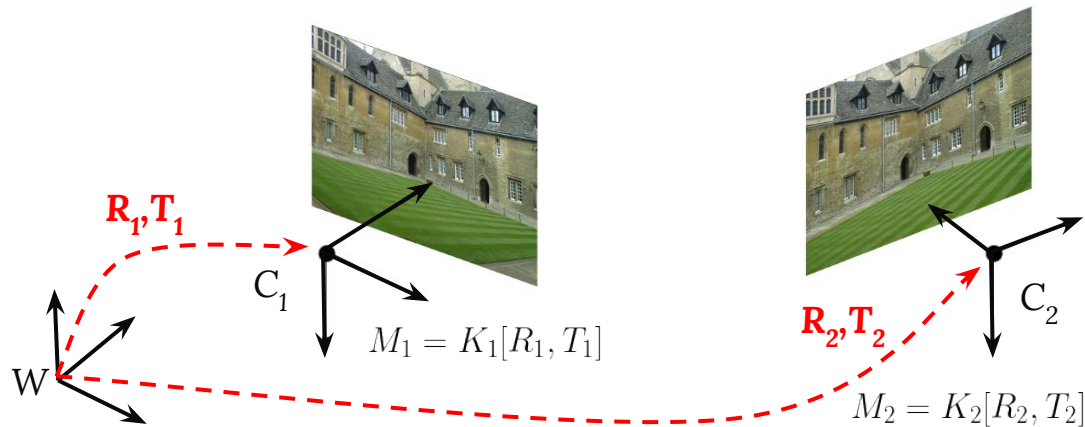
Triangulation



Triangulation

Calibrated Stereo System

Given known camera matrices, a set of point correspondences can be triangulated to recover the 3D positions of these points in the world reference frame.



Triangulation

Calibrated Stereo System

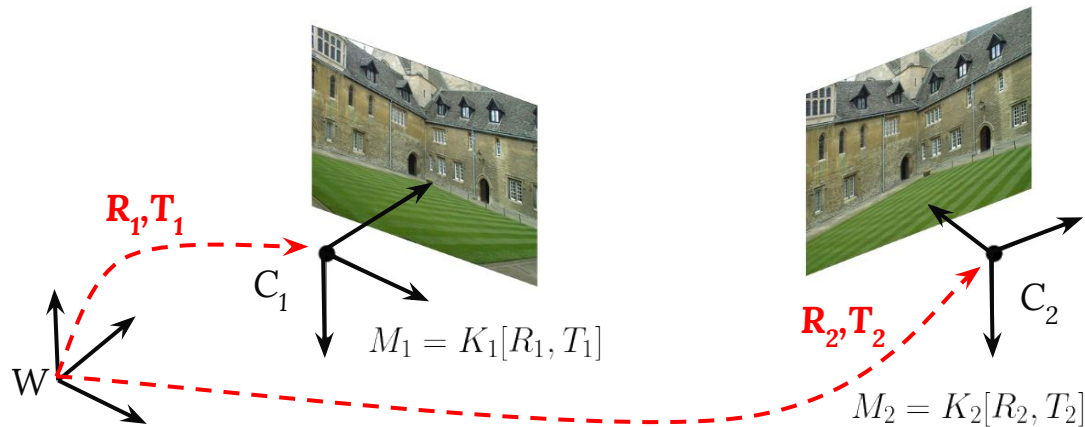
Given known camera matrices, a set of point correspondences can be triangulated to recover the 3D positions of these points in the world reference frame.

$$\lambda_1 p'_1 = K_1[R_1, T_1] P_w$$

$$\lambda_1 p'_1 = M_1 P_w$$

$$\lambda_2 p'_2 = K_2[R_2, T_2] P_w$$

$$\lambda_2 p'_2 = M_2 P_w$$



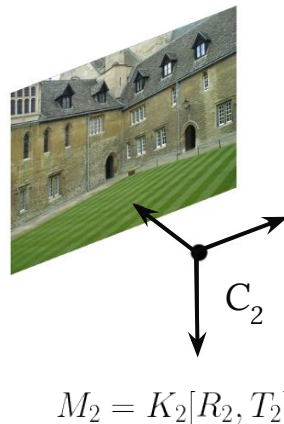
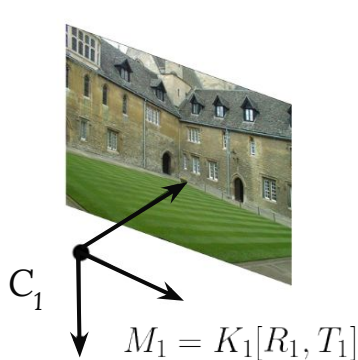
Triangulation

Calibrated Stereo System

$$\lambda_1 p'_1 = M_1 P_w$$



$$M_1 P_w - \lambda_1 p'_1 = 0$$



$$\lambda_2 p'_2 = M_2 P_w$$



$$M_2 P_w - \lambda_2 p'_2 = 0$$

Triangulation

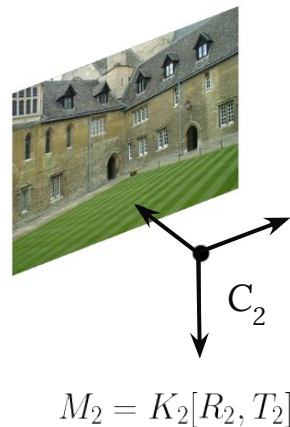
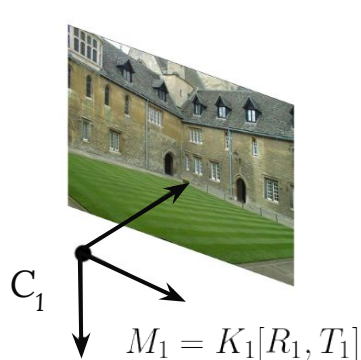
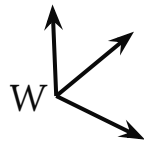
Calibrated Stereo System

$$M_1 P_w - \lambda_1 p'_1 = 0$$

$$M_2 P_w - \lambda_2 p'_2 = 0$$



$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$



Triangulation


Calibrated Stereo System

$$M_1 P_w - \lambda_1 p'_1 = 0$$

$$M_2 P_w - \lambda_2 p'_2 = 0$$

We consider a scale factor k because we are recovering the 3D point in homogeneous coordinates.

It will be estimated with $k \neq 1$.

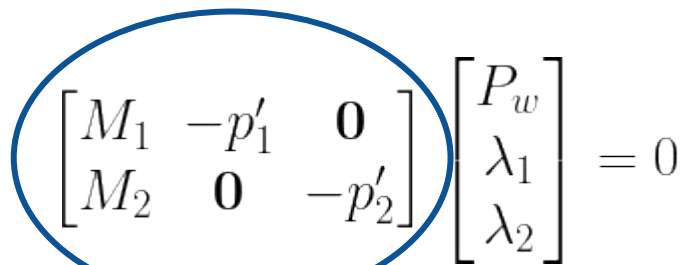

$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0 \quad \longrightarrow \quad \begin{bmatrix} [M_1]_{3 \times 4} & -x'_1 & 0 \\ & -y'_1 & 0 \\ & 1 & 0 \\ [M_2]_{3 \times 4} & 0 & -x'_2 \\ & 0 & -y'_2 \\ & 0 & 1 \end{bmatrix} \begin{bmatrix} kX_w \\ kY_w \\ kZ_w \\ k \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Triangulation

Calibrated Stereo System

There might not be an exact solution to these equations due to image noise, errors in the camera matrices, or other sources of errors.

Using SVD, we can get a least squares estimate of the 3D point.


$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$

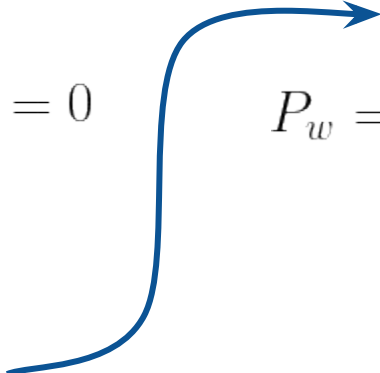
$$\chi \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$

Triangulation

Calibrated Stereo System

There might not be an exact solution to these equations due to image noise, errors in the camera matrices, or other sources of errors.

Using SVD, we can get a least squares estimate of the 3D point.

$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$

$$\chi \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$
$$SVD(\chi) = U\Sigma V^T$$
$$P_w = V[\text{first 4 lines, last column}]$$

Triangulation

Calibrated Stereo System

There might not be an exact solution to these equations due to image noise, errors in the camera matrices, or other sources of errors.

Using SVD, we can get a least squares estimate of the 3D point.

$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$
$$SVD(\chi) = U\Sigma V^T$$
$$P_w = V[\text{first 4 lines, last column}]$$
$$\chi \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$
$$P_w = \begin{bmatrix} kX_w \\ kY_w \\ kZ_w \\ k \end{bmatrix} \xrightarrow{\text{Divide by the last element to get cartesian coordinates}} P_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

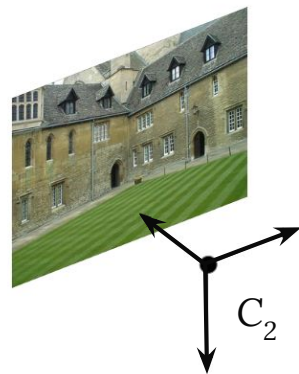
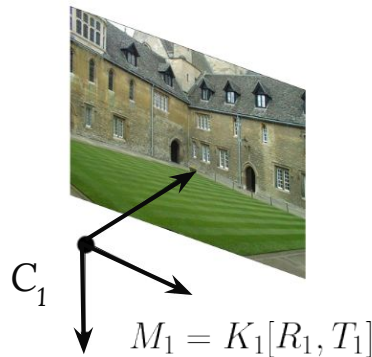
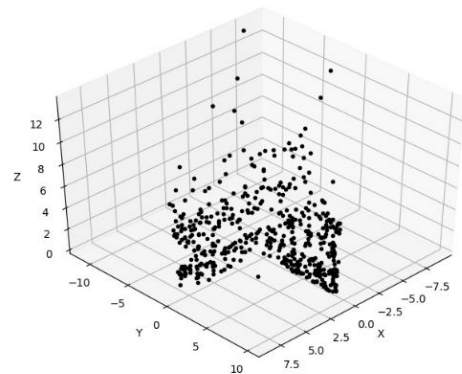
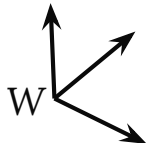
Triangulation

Calibrated Stereo System

So, by triangulating each pair of corresponding points in both images, we can recover the 3D information.

$$\begin{bmatrix} M_1 & -p'_1 & \mathbf{0} \\ M_2 & \mathbf{0} & -p'_2 \end{bmatrix} \begin{bmatrix} P_w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0$$

$$P_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



$$M_2 = K_2[R_2, T_2]$$

Credits



Yi Ma, Stefano Soatto, Jana Kosecka e S. Shankar Sastry. An Invitation to 3D Vision: From Images to Geometric Models.
Springer, ISBN 0387008934



Jan Erik Solem. Programming Computer Vision with Python.
Published by O'Reilly Media, ISBN: 978-1-449-31654-9