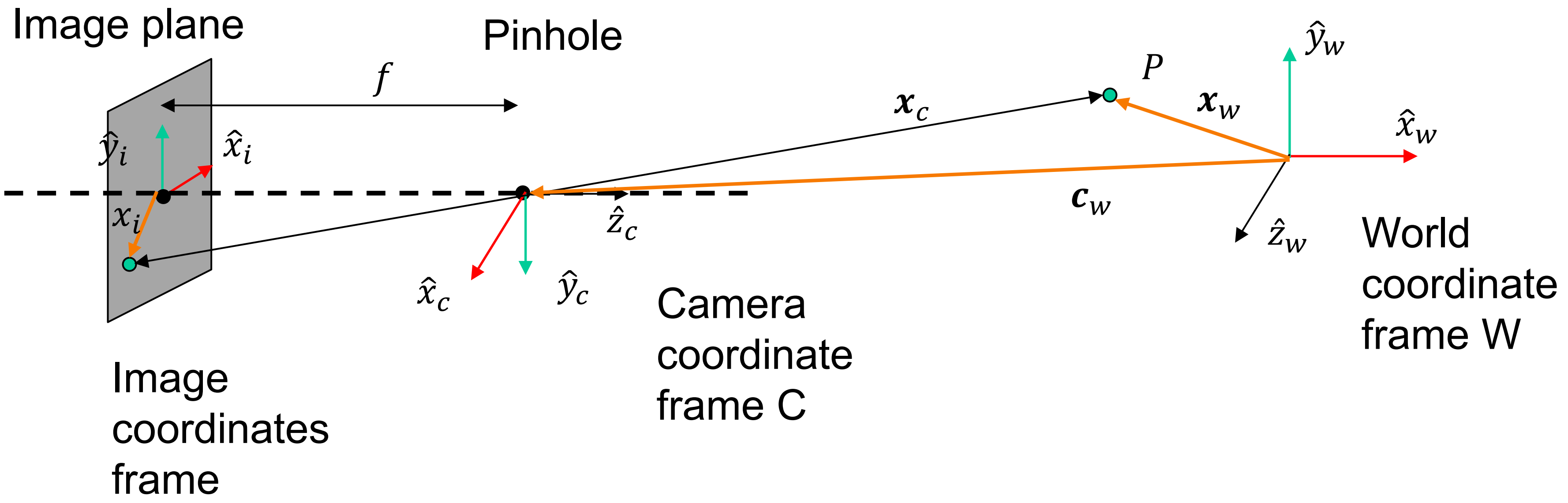
ECE 484
Lecture 7
Camera Calibration
Visual Odometry

Prof. Huan Zhang

Fall 2026

Slides adapted in part from materials by Prof. Sayan Mitra.

The forward imaging model has two stages



$$\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

3D-2D

$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

3D-3D

$$\boldsymbol{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Forward Camera Model

Camera to pixel

$$\boxed{\begin{bmatrix}\tilde{u}\\\tilde{v}\\\tilde{w}\end{bmatrix} = \begin{bmatrix}f_x & 0 & o_x & 0\\0 & f_y & o_y & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_c\\y_c\\z_c\\1\end{bmatrix}}$$

$$\tilde{\boldsymbol{u}} = M_{int}\ \tilde{\boldsymbol{x}}_c = [K|0]\tilde{\boldsymbol{x}}_c$$

World to camera

$$\boxed{\begin{bmatrix}x_c\\y_c\\z_c\\1\end{bmatrix} = \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}}$$

$$\tilde{\boldsymbol{x}}_c = M_{ext}\ \tilde{\boldsymbol{x}}_w$$

$$\boxed{\tilde{\boldsymbol{u}} = M_{\text{int}}\, M_{\text{ext}}\, \tilde{\boldsymbol{x}}_w = P\,\tilde{\boldsymbol{x}}_w = K[R\,|\,\boldsymbol{t}]\,\tilde{\boldsymbol{x}}_w}$$

$$\begin{bmatrix}\tilde{u}\\\tilde{v}\\\tilde{w}\end{bmatrix} = \begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\p_{21} & p_{22} & p_{23} & p_{24}\\p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}$$

P: **Projection matrix**

Outline

Forward camera model

Calibration

Visual Odometry

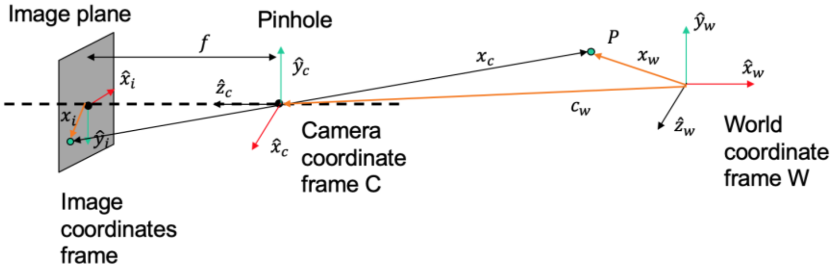
Camera Calibration

The extrinsic / intrinsic parameters of a camera is often unknown

So even after the derivation we still cannot easily do the conversion

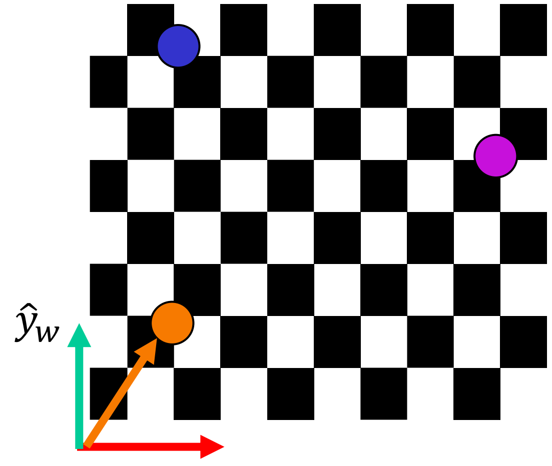
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

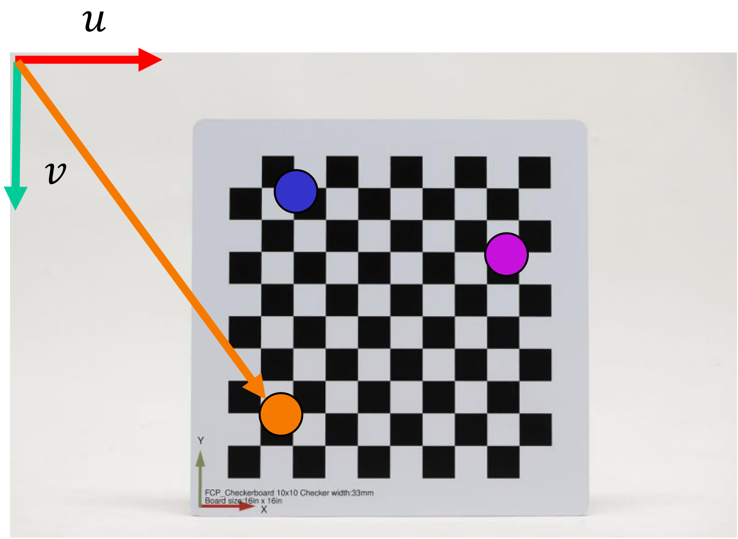


Camera Calibration Procedure

Step 1. Capture image of object with known geometry



known geometry object



captured image

$$\boldsymbol{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

$$\boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Camera Calibration

Step 2. For each point i with $(x_w^{(i)}, y_w^{(i)}, z_w^{(i)})$ we get the projection relation (up to scale)

$$\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix} \equiv \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix}$$

Step 3. Divide row 1 by row 3: $u^{(i)} = \frac{p_{11}x_w^{(i)} + p_{12}y_w^{(i)} + p_{13}z_w^{(i)} + p_{14}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$ (row 2 over row 3 gives $v^{(i)}$), and

rearranging $\boldsymbol{p} = [p_{11}\ p_{12} \cdots p_{34}]^T \in \mathbb{R}^{12}$ we get $A\boldsymbol{p} = 0$ with two rows of A per point:

$$A_{u,i} = [\, x_w^{(i)},\ y_w^{(i)},\ z_w^{(i)},\ 1,\ 0,\ 0,\ 0,\ 0,\ -u^{(i)}x_w^{(i)},\ -u^{(i)}y_w^{(i)},\ -u^{(i)}z_w^{(i)},\ -u^{(i)}\,]$$

$$A_{v,i} = [\, 0,\ 0,\ 0,\ 0,\ x_w^{(i)},\ y_w^{(i)},\ z_w^{(i)},\ 1,\ -v^{(i)}x_w^{(i)},\ -v^{(i)}y_w^{(i)},\ -v^{(i)}z_w^{(i)},\ -v^{(i)}\,]$$

Step 4. Solve for $\boldsymbol{p}$: $A \in \mathbb{R}^{2n \times 12}$; need $n \geq 6$ points not all in one plane (then rank $A = 11$)

$A\boldsymbol{p} = 0$

Find $\boldsymbol{p}$ such that $A\boldsymbol{p} = 0$

Find a vector in the null space of A

With noisy measurements A has full column rank, so the only exact solution is $\boldsymbol{p} = 0 \rightarrow$ minimize $||A\boldsymbol{p}||_2$ instead (next slides)

Also, remember we can only find $\boldsymbol{p}$ up to scale

- Fix one element, e.g. $p_{34} = 1$ (fails if the true p_{34} happens to be 0), OR
- Add $||\boldsymbol{p}||_2 = 1$ as a constraint (always works — used from here on)

Projection matrix scale

Since projection matrix works on homogeneous coordinates

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix} \equiv k\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix}$$

Therefore

$$\begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\ p_{21} & p_{22} & p_{23} & p_{24}\\ p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}x_w\\ y_w\\ z_w\\ 1\end{bmatrix} \equiv k\begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\ p_{21} & p_{22} & p_{23} & p_{24}\\ p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}x_w\\ y_w\\ z_w\\ 1\end{bmatrix}$$

Therefore P and kP produce the same homogeneous pixel coordinates

The projection matrix is defined only up to a scale factor

(Separately: scaling the whole scene and the camera by a common factor also gives the same image — an ambiguity of 3D reconstruction, not of P)

We fix the scale by choosing $\|\boldsymbol{p}\|_2 = 1$, so P has 12 − 1 = 11 degrees of freedom

Least Squares Solution for Projection Matrix

We want $A\boldsymbol{p}$ as close to 0 as possible, with $\left||\boldsymbol{p}|\right|^2 = 1$

$\min_{\boldsymbol{p}} \|A\boldsymbol{p}\|_2^2$ such that $\|\boldsymbol{p}\|_2^2 = 1$

$\min_{\boldsymbol{p}} \boldsymbol{p}^T A^T A\boldsymbol{p}$ such that $\boldsymbol{p}^T\boldsymbol{p} = 1$ (because $\|A\boldsymbol{p}\|_2^2 = (A\boldsymbol{p})^T(A\boldsymbol{p}) = \boldsymbol{p}^T A^T A\boldsymbol{p}$)

$L(\boldsymbol{p}, \lambda) = \boldsymbol{p}^{\boldsymbol{T}} A^T A\boldsymbol{p} - \lambda\left(\boldsymbol{p}^{\boldsymbol{T}}\boldsymbol{p} - 1\right)$. Unconstrained optimization

Taking derivative $\frac{\partial L}{\partial \boldsymbol{p}} = 0$ gives $2A^T A\boldsymbol{p} - 2\lambda\boldsymbol{p} = \boldsymbol{0}$

$$\boxed{A^T A\boldsymbol{p} = \lambda\boldsymbol{p}}$$

$\boldsymbol{p}$ is the eigenvector corresponding to the smallest eigenvalue of $A^T A$

Rearrange $\boldsymbol{p}$ to get the projection matrix **P**

At a unit eigenvector $\|A\boldsymbol{p}\|_2^2 = \boldsymbol{p}^T A^T A\boldsymbol{p} = \lambda$, so the minimum is attained at $\lambda_{\min}$ ($A^T A$ is symmetric positive semidefinite: all $\lambda \geq 0$)

From the projection matrix P to K, R and t

Camera to pixel

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

World to camera

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\tilde{\boldsymbol{u}} = M_{int}\ \tilde{\boldsymbol{x}}_c = [K|0]\tilde{\boldsymbol{x}}_c$$

$$\tilde{\boldsymbol{x}}_c = M_{ext}\ \tilde{\boldsymbol{x}}_w$$

$$\boxed{\tilde{\boldsymbol{u}} = M_{\text{int}}\, M_{\text{ext}}\, \tilde{\boldsymbol{x}}_w = P\,\tilde{\boldsymbol{x}}_w = K[R\,|\,\boldsymbol{t}]\,\tilde{\boldsymbol{x}}_w}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

After we know P, calculate K, R, t

From the projection matrix P to K, R and t

1. Reshape $P \in \mathbb{R}^{3\times4}$ $P = [M \mid \boldsymbol{p_4}]$ with $M \in \mathbb{R}^{3\times3}$
2. Factor M into intrinsic K and rotation R using RQ factorization:
   $M = K\,R$ where K is upper triangular (choose its diagonal positive by flipping the sign of a column of K and of the matching row of R) and R is orthogonal; if det R = −1 replace (R, t) by (−R, −t), i.e. P by −P
3. Solve for translation
   From $P = K\,[R \mid \boldsymbol{t}] = [K\,R \mid K\,\boldsymbol{t}]$ gives:
   $$\boldsymbol{p_4} = K\boldsymbol{t} \Rightarrow \boldsymbol{t} = K^{-1}\boldsymbol{p_4}$$

From data we have derived the intrinsic $M_{int} = [K|0]$ and the extrinsic $M_{ext} = [R|t]$ matrices and we have a fully calibrated camera

Summary: Monocular vision

- From pixel and 3D coordinate data we can solve the eigenvalue problem to get the camera's projection matrix
- From projection matrix and RQ decomposition we obtain the camera's intrinsic and extrinsic matrices
- The intrinsic and extrinsic matrices define the camera's forward model: transformation from world $W \rightarrow C \rightarrow i$ (pixel coordinates)
- Going back from pixels to the world is the next step: a pixel fixes only a ray through the camera centre, so depth needs a second view (stereo, visual odometry), motion, or a scene constraint (e.g. the ground plane): pose estimation, structure from motion

Outline

Forward camera model

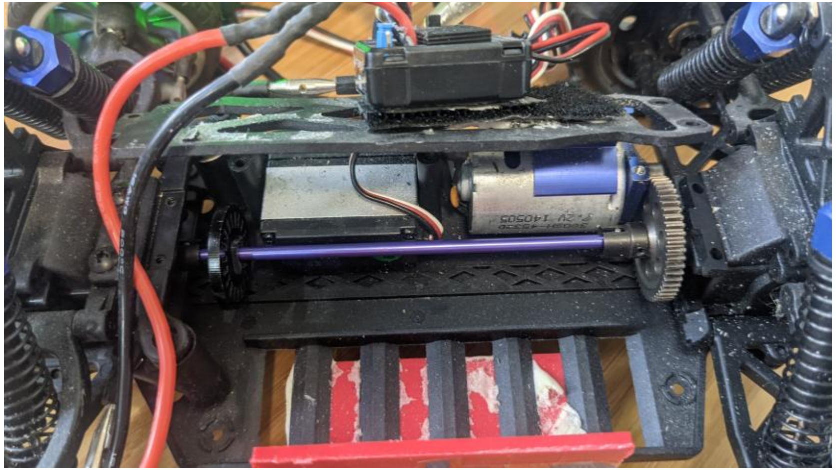
Calibration

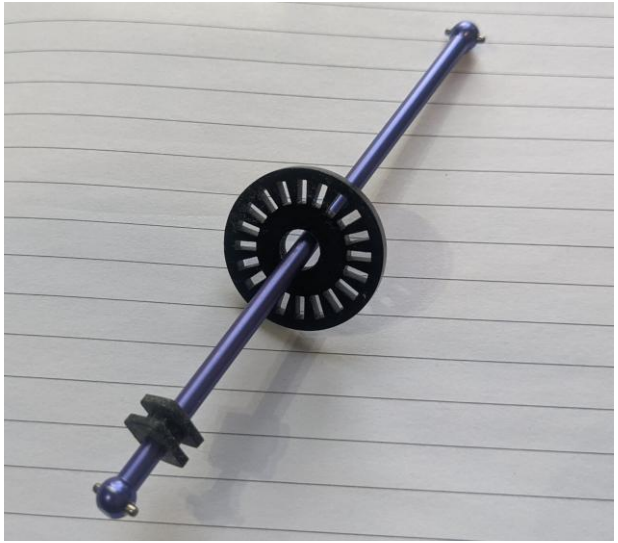
Visual Odometry

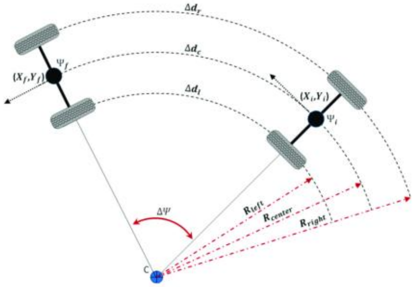
Problem: Odometry

a method that uses onboard motion sensors to estimate how a robot or vehicle's position and orientation (pose) change over time relative to a starting point

Wheel odometry: using optical encoders to measure the shaft turns



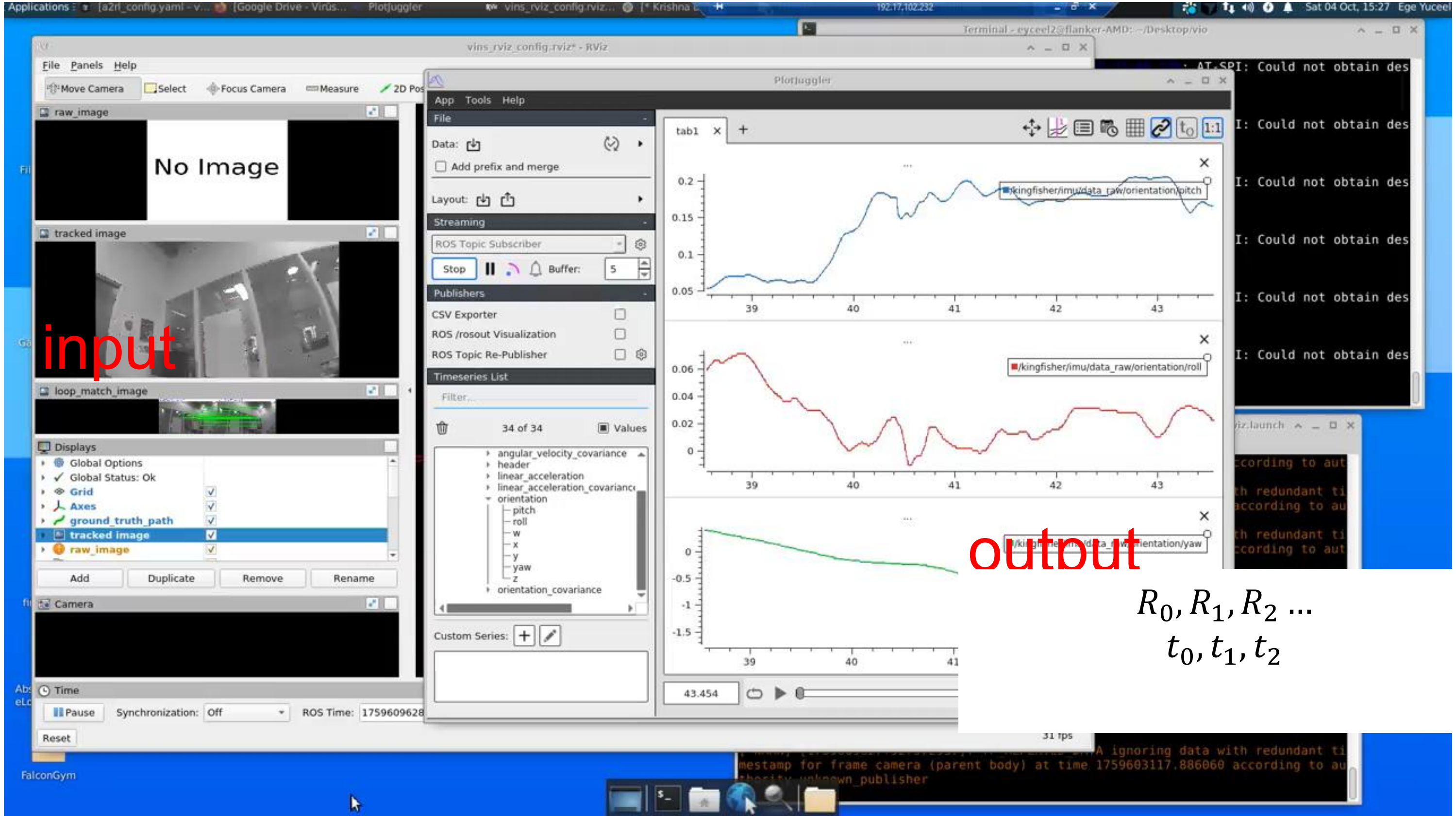




https://docs.donkeycar.com/parts/odometry/

Problem: Visual Odometry (VO)

Incrementally estimate pose of the vehicle from onboard camera images



$$R_0, R_1, R_2 \dots$$
$$t_0, t_1, t_2$$

Why VO ?

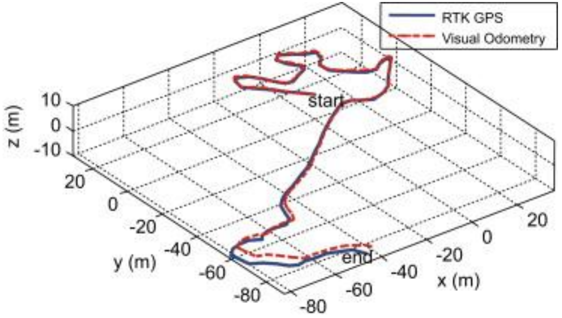
Unlike wheel odometry, VO is not affected by wheel slip in uneven terrain or other adverse conditions.

More accurate trajectory estimates compared to wheel odometry (relative position error 0.1% − 2%)

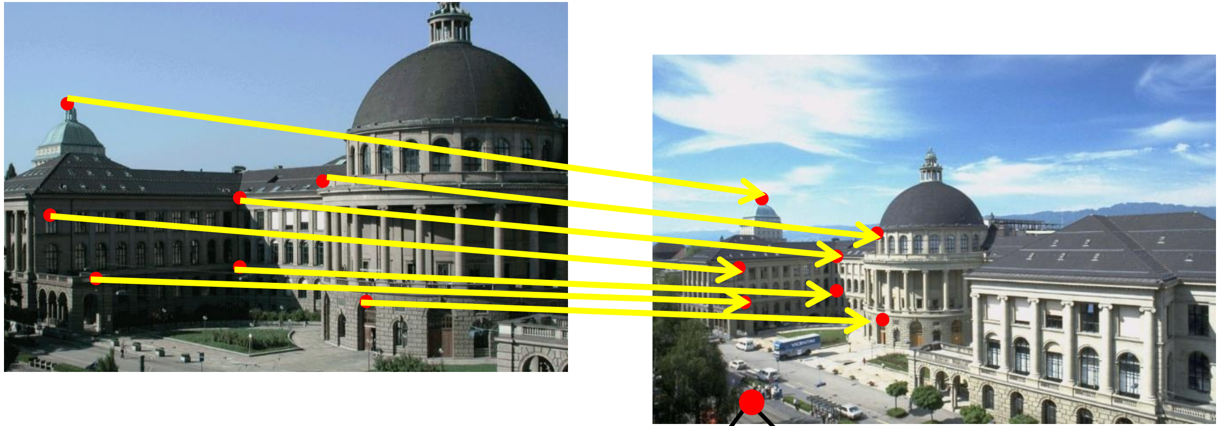
VO can be used as a complement to

- wheel odometry, GPS, IMUs, laser odometry

GPS-denied environments, e.g., underwater, Mars

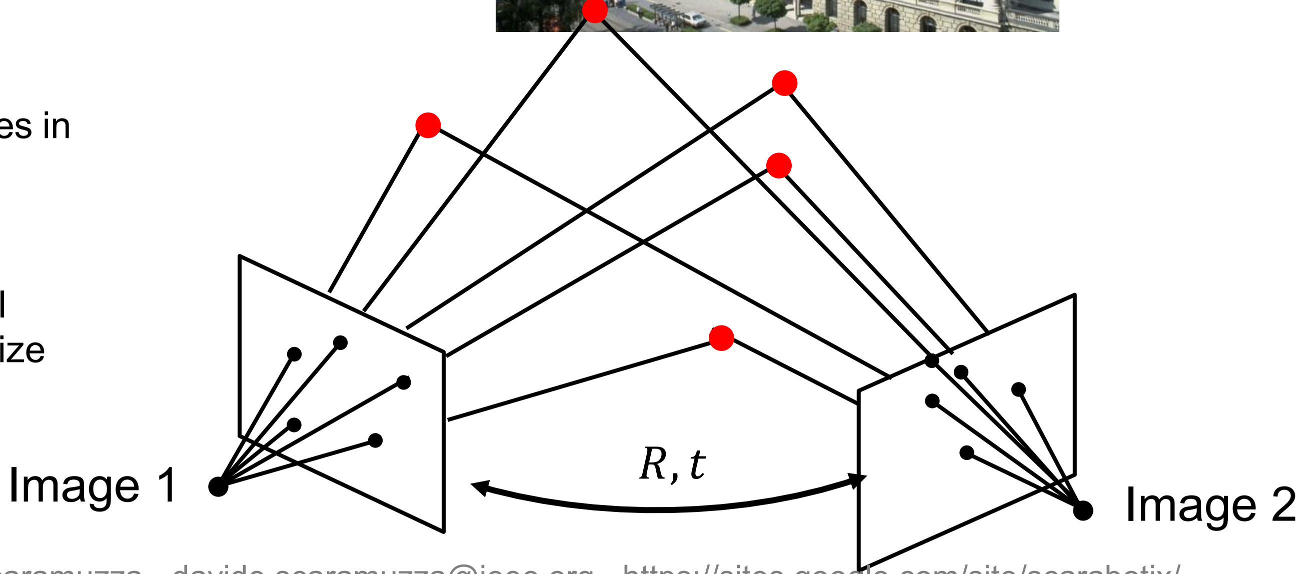


VO Principle



VO Solution Outline

1. Detect and match features in two successive images
2. Estimate motion R_k, t_k (Epipolar geometry)
3. RANSAC outlier removal
4. Repeat Steps 1-3, optimize



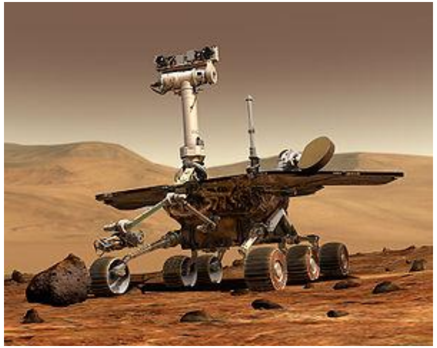
Brief history of VO

- 1996: The term VO was coined by Srinivasan to define motion orientation in honey bees.

- 1980: First known stereo VO real-time implementation on a robot by Moravec PhD thesis (NASA/JPL) for Mars rovers using a sliding camera. Moravec invented a predecessor of Harris detector, known as Moravec detector

- 1980 to 2000: The VO research was dominated by NASA/JPL in preparation of 2004 Mars mission (see papers from Matthies, Olson, etc. From JPL)

- 2004: VO used on a robot on another planet: Mars rovers Spirit and Opportunity



- 2004. VO was revived in the academic environment by Nister «Visual Odometry» paper. The term VO became popular.

Recall forward imaging with calibrated camera

Camera to pixel

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

World to camera

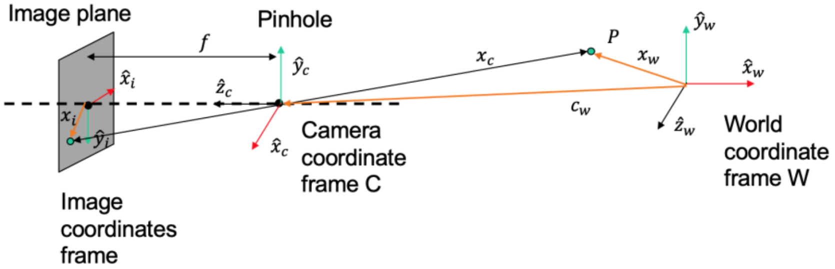
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$\tilde{u} = M_{int}\ \tilde{x}_c$ **Intrinsic**

$\tilde{x}_c = M_{ext}\ \tilde{x}_w$ **Extrinsic**

$$\boxed{\tilde{u} = M_{int} M_{ext}\ \tilde{x}_w = P\ \tilde{x}_w}$$

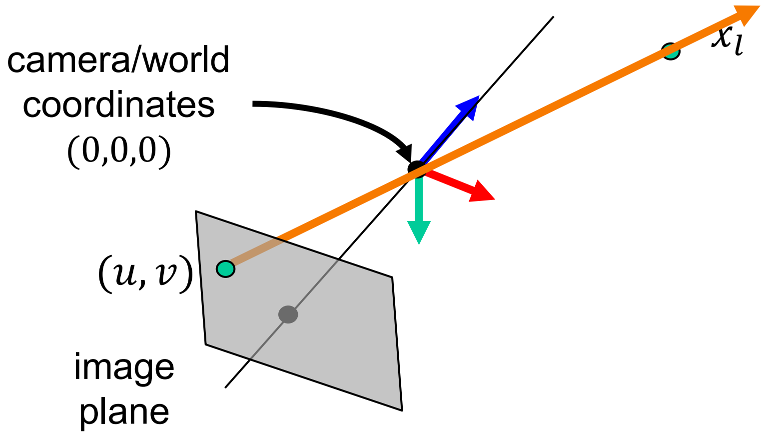
$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$



P: **Projection matrix** $P = K[R \mid \boldsymbol{t}]$, 3×4, up to scale $\Rightarrow$ 11 DoF

Slide ideas from First Principles of Computer Vision by Shree Nayar

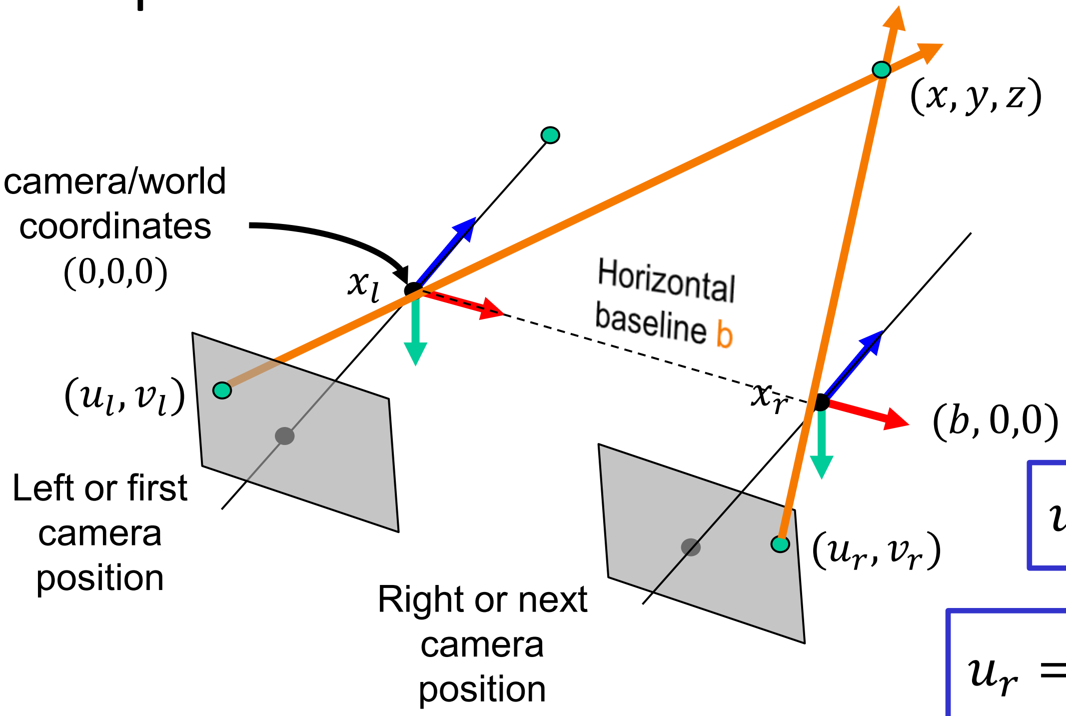
Backward projection from 2D pixel to 3D ray



3D to 2D $u = f_x \frac{x_c}{z_c} + o_x$ and $v = f_y \frac{y_c}{z_c} + o_y$

2D to 3D $x_c = \frac{z_c}{f_x}(u - o_x),\ y_c = \frac{z_c}{f_y}(v - o_y);\ z_c > 0$ unknown

Simple Stereo Problem: Reconstructing scene from two images



- Simple (rectified) stereo assumes:
  - both cameras have the SAME intrinsics f_x, f_y, o_x, o_y
  - the second camera is the first TRANSLATED by $(b, 0,0)$ — no rotation, so the image planes stay coplanar
  - (x, y, z) are coordinates in the LEFT camera frame, $z > 0$
  - the correspondence $(u_l, v_l) \leftrightarrow (u_r, v_r)$ is already known

$$\boxed{u_l = f_x \frac{x}{z} + o_x \text{ and } v_l = f_y \frac{y}{z} + o_y}$$

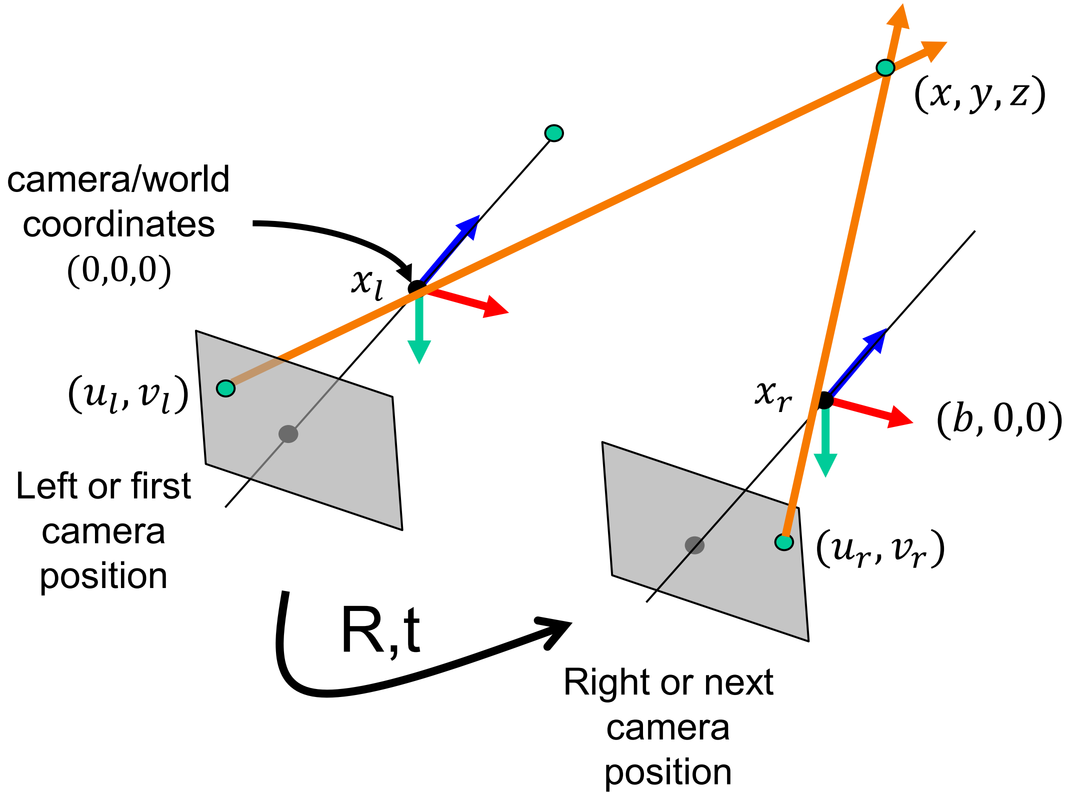
$$\boxed{u_r = f_x \frac{x-b}{z} + o_x \text{ and } v_r = f_y \frac{y}{z} + o_y}$$

Subtract: $u_l - u_r = \frac{f_x b}{z}$; with the disparity $d := u_l - u_r > 0$: $z = \frac{b f_x}{d}$, $x = \frac{b\,(u_l - o_x)}{d}$, $y = \frac{b f_x (v_l - o_y)}{f_y\, d}$

and $v_l = v_r$: the match lies on the SAME image row, so the search is 1-D

Slide ideas from First Principles of Computer Vision by Shree Nayar

In General: Uncalibrated Stereo Problem



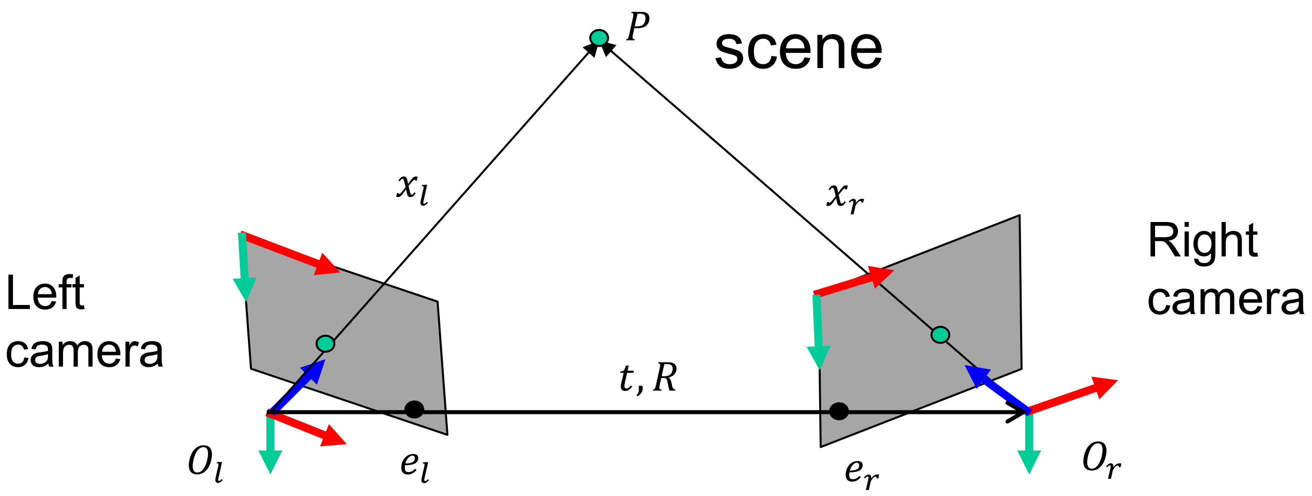
The two cameras' intrinsic matrices K_l, K_r are known; the rotation R and the translation $\boldsymbol{t}$ between the two camera poses are unknown

This relation between the two poses is described by **epipolar geometry**

Estimated from pixel matches as the fundamental matrix F, then converted to essential matrix E with the known K_l, K_r

Slide ideas from First Principles of Computer Vision by Shree Nayar

Epipolar Geometry: Epipoles



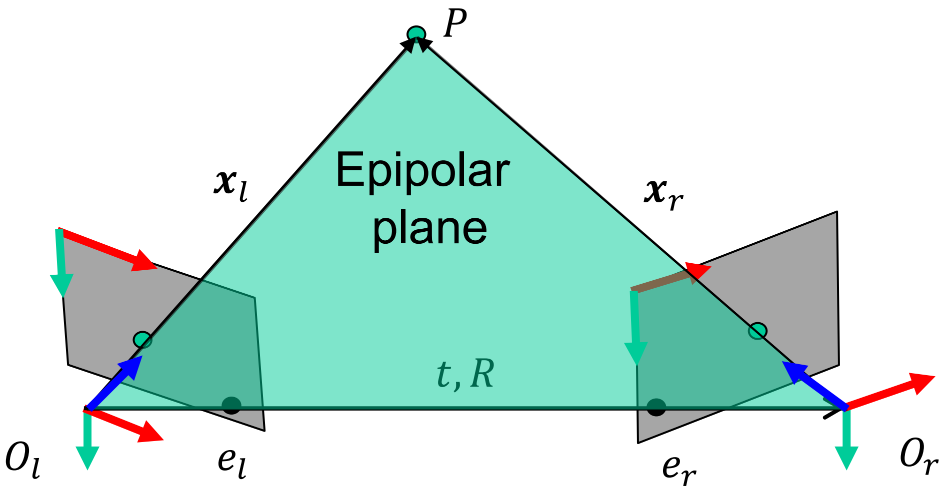
The intrinsic parameters (f_x, f_y, o_x, o_y) are given for each camera

Problem: Compute extrinsic parameters t, R relating the two camera positions

Epipoles: $\boldsymbol{e}_l$ = image of the RIGHT centre O_r in the left image, $\boldsymbol{e}_r$ = image of the LEFT centre O_l in the right image. Both lie on the baseline $O_l O_r$, so every epipolar line of an image passes through its epipole

This figure uses a virtual image plane in front of the optical center

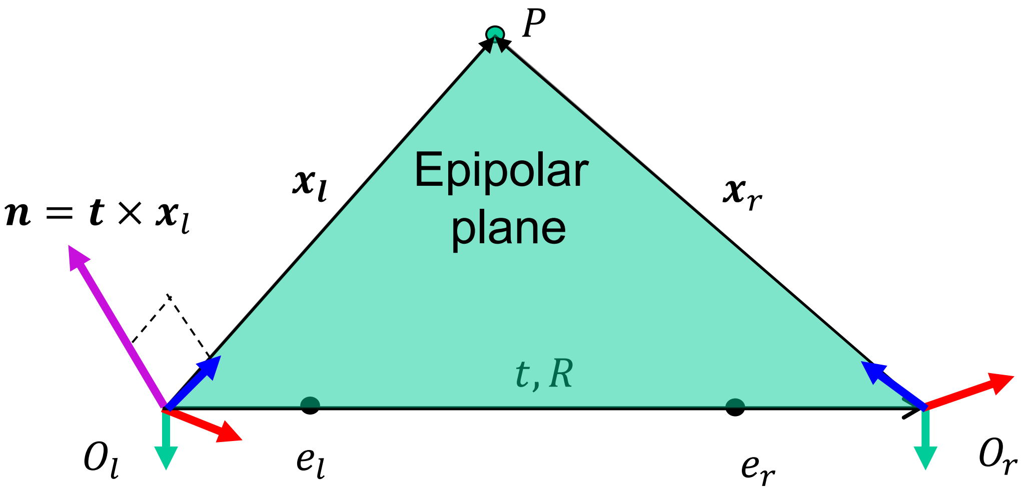
Epipolar Geometry: Epipolar Plane



Epipolar plane of a scene point P: the plane through the two camera centres O_l, O_r and P
(the epipoles lie on the baseline $O_l O_r$, hence in this plane automatically)

The plane is unique provided P is NOT on the baseline; if it is, every plane through $O_l O_r$ contains P and the constraint degenerates

Epipolar Constraints: derivations



Recall $a \times b = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$

Vector normal to the epipolar plane: $\boldsymbol{n} = \boldsymbol{t} \times \boldsymbol{x_l}$

$\boldsymbol{x}_l$ lies in that plane, so $\boldsymbol{x}_l \cdot \boldsymbol{n} = \boldsymbol{x}_l \cdot (\boldsymbol{t} \times \boldsymbol{x}_l) = 0$ — true for EVERY $\boldsymbol{t}$: the two views are coupled only after $\boldsymbol{x}_l$ is replaced by $R\boldsymbol{x}_r + \boldsymbol{t}$ (next slide)

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \begin{bmatrix} t_y z_l - t_z y_l \\ t_z x_l - t_x z_l \\ t_x y_l - t_y x_l \end{bmatrix} = 0 \qquad \begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \underbrace{\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}}_{T_X} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

Epipolar Constraints

$\boldsymbol{x}_l$ lies in the epipolar plane: $\boldsymbol{x}_l \cdot (\boldsymbol{t} \times \boldsymbol{x}_l) = 0$

$$[x_l \quad y_l \quad z_l]\begin{bmatrix} t_y z_l - t_z y_l \\ t_z x_l - t_x z_l \\ t_x y_l - t_y x_l \end{bmatrix} = 0 \qquad [x_l \quad y_l \quad z_l]\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

Relating 3D coordinates of P in the left camera with that of the right $\boldsymbol{x}_l = R\boldsymbol{x}_r + \boldsymbol{t}$

Substituting $\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ in the second $\boldsymbol{x}_l$

Essential matrix

$$E = T_X R = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

$$[x_l \quad y_l \quad z_l]\left(\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \underbrace{\boxed{\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}}}_{0}\right) = 0$$

$$[x_l \quad y_l \quad z_l]\boxed{\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

$E = T_X R$ is a skew-symmetric matrix times a ROTATION ($R^T R = I$, $\det R = +1$).

SVD of E returns R and $\boldsymbol{t}$, but $\boldsymbol{t}$ only up to scale and sign, and in four combinations — see the next slides

How to find the Essential Matrix?

$$x_l^T E x_r = 0$$

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

Unfortunately, we do not have the 3D coordinates of the same scene point x_l and x_r

But we know the corresponding points in image coordinates

Incorporating image coordinates

$$u_l = f_x^{(l)} \frac{x_l}{z_l} + o_x^{(l)} \qquad v_l = f_y^{(l)} \frac{y_l}{z_l} + o_y^{(l)}$$

$$z_l u_l = f_x^{(l)} x_l + z_l o_x^{(l)} \qquad z_l v_l = f_y^{(l)} y_l + z_l o_y^{(l)}$$

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & 0 & o_x^{(l)} \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

Known intrinsic matrix K_l

Slide ideas from First Principles of Computer Vision by Shree Nayar

Incorporating image coordinates in Epipolar constraints

Left camera

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & 0 & o_x^{(l)} \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

$$\boldsymbol{x}_l^T = z_l [u_l \quad v_l \quad 1] K_l^{-T}$$

Right camera

$$z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(r)} & 0 & o_x^{(r)} \\ 0 & f_y^{(r)} & o_y^{(r)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}$$

$$\boldsymbol{x}_r = K_r^{-1} z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix}$$

Plug into the epipolar constraints

$$z_l z_r [u_l \quad v_l \quad 1] K_l^{-T} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

A visible point has $z_l, z_r > 0$, so both depths divide out

$$[u_l \quad v_l \quad 1] K_l^{-T} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

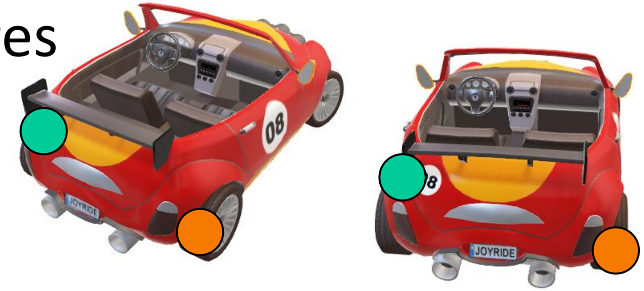
$$[u_l \quad v_l \quad 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

$F = K_l^{-T} E K_r^{-1}$ Fundamental matrix

$$E = K_l^T F K_r$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Finding Fundamental Matrix: Matching features



Find a set of corresponding features in the two images (e.g. using SIFT)

$(u_l^{(1)}, v_l^{(1)})$ $(u_r^{(1)}, v_r^{(1)})$

… …

$(u_l^{(m)}, v_l^{(m)})$ $(u_r^{(m)}, v_r^{(m)})$

Plugging into Epipolar Constraints

$$\begin{bmatrix} u_l^{(i)} & v_l^{(i)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r^{(i)} \\ v_r^{(i)} \\ 1 \end{bmatrix} = 0$$

known unknown known

One linear equation for each matched feature i

Stacking equations for all the features and the elements of F as a vector $\boldsymbol{f}$ we get

$$A\,\boldsymbol{f} = \boldsymbol{0}$$

With $\boldsymbol{f} = (f_{11}, \dots, f_{33})^T \in \mathbb{R}^9$ the i-th row of A is

$(u_l u_r,\ u_l v_r,\ u_l,\ v_l u_r,\ v_l v_r,\ v_l,\ u_r,\ v_r,\ 1)$, so $A \in \mathbb{R}^{m\times 9}$

F is defined up to scale, so $m \geq 8$ correspondences are needed — the EIGHT-POINT algorithm

Missing scale

$$[u_l \quad v_l \quad 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0 \;\Leftrightarrow\; [u_l \quad v_l \quad 1]\begin{bmatrix} kf_{11} & kf_{12} & kf_{13} \\ kf_{21} & kf_{22} & kf_{23} \\ kf_{31} & kf_{32} & kf_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0 \text{ for every } k \neq 0$$

The fundamental matrix works on homogeneous coordinates

Fundamental matrix F and $\mathrm{k}F$ describe the same epipolar geometry

F is defined only up to a scale factor

Fix the scale by $||\boldsymbol{f}||_2 = 1$

Solving for F up to Scale

$$\begin{bmatrix} u_l^{(i)} & v_l^{(i)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r^{(i)} \\ v_r^{(i)} \\ 1 \end{bmatrix} = 0$$

known unknown known

One linear equation for each matched feature i

Stacking equations for all the features and the elements of F as a vector $\boldsymbol{f}$ we get

$$A\,\boldsymbol{f} = \boldsymbol{0}$$

We want $A\boldsymbol{f}$ as close to 0 as possible and $||\boldsymbol{f}||^{\mathbf{2}} = 1$:

Constrained linear least squares $\min_{\boldsymbol{f}} \|A\boldsymbol{f}\|_2^2$ such that $\|\boldsymbol{f}\|_2 = 1 \;\Rightarrow\; \boldsymbol{f}$ = right singular vector of A for the smallest singular value

From $\boldsymbol{f}$ rearrange to get F, compute $E = K_l^T F K_r$ and extract $R, \boldsymbol{t}$ from $E = T_X R$ by singular value decomposition (SVD)
(read: https://www.robots.ox.ac.uk/~vgg/hzbook/hzbook1/HZepipolar.pdf)