DTU

Perception for Autonomous Systems 31392:

Camera Matrix and Camera Calibration

Lecturer: Evangelos Boukas—PhD

Outline

- Image formation
- Camera model – Project form World to sensor
- Camera calibration – The 3D case
- Camera calibration – The realistic case
- Camera radial distortion

Image Formation

- Let’s look at
  the most simplistic case.
  What’s weird about it?

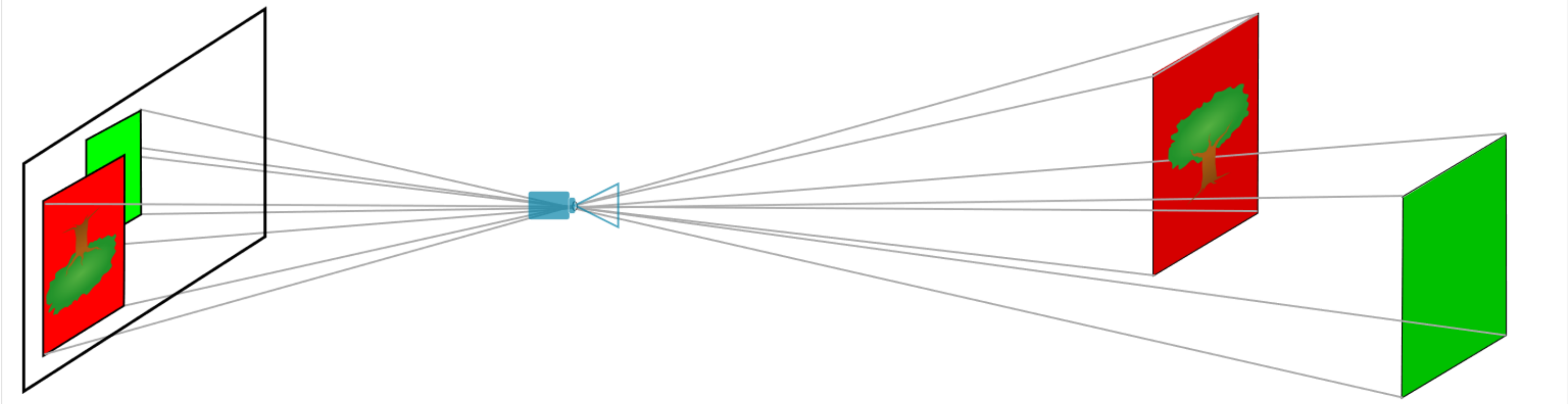


Image Formation

- Let’s look at
  the most simplistic case.
  What’s weird about it?
- Let’s Fix that

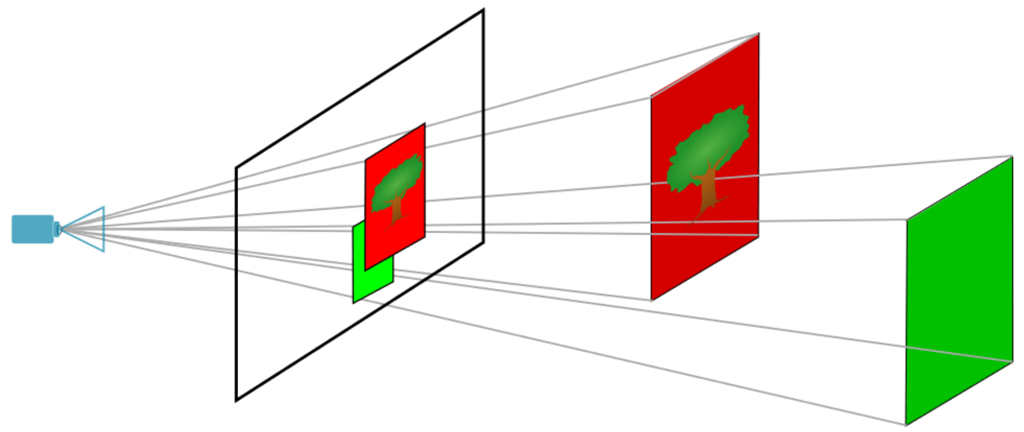


Image Formation

- Let’s look at
  the most simplistic case.
  What’s weird about it?
- Let’s Fix that

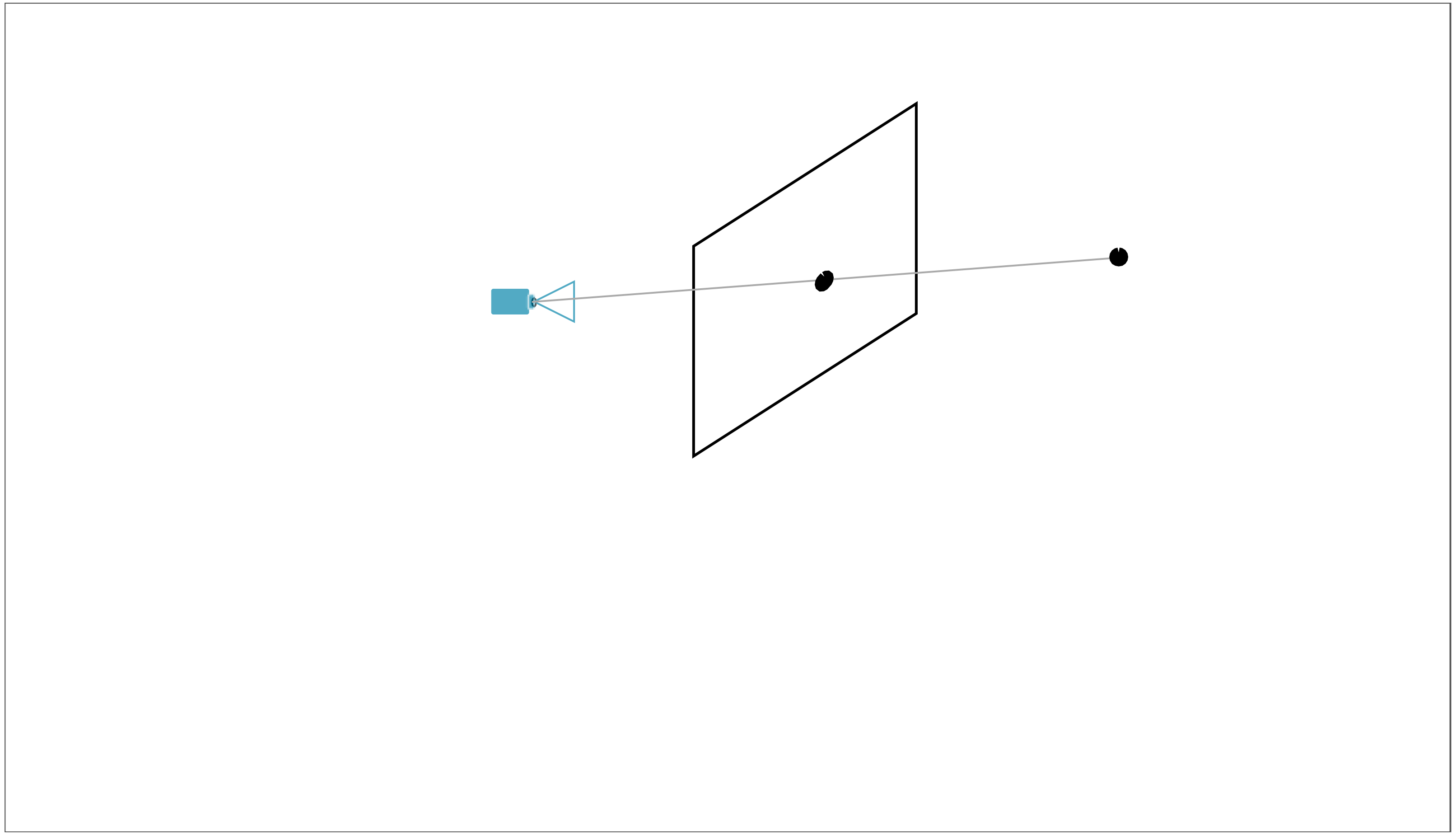
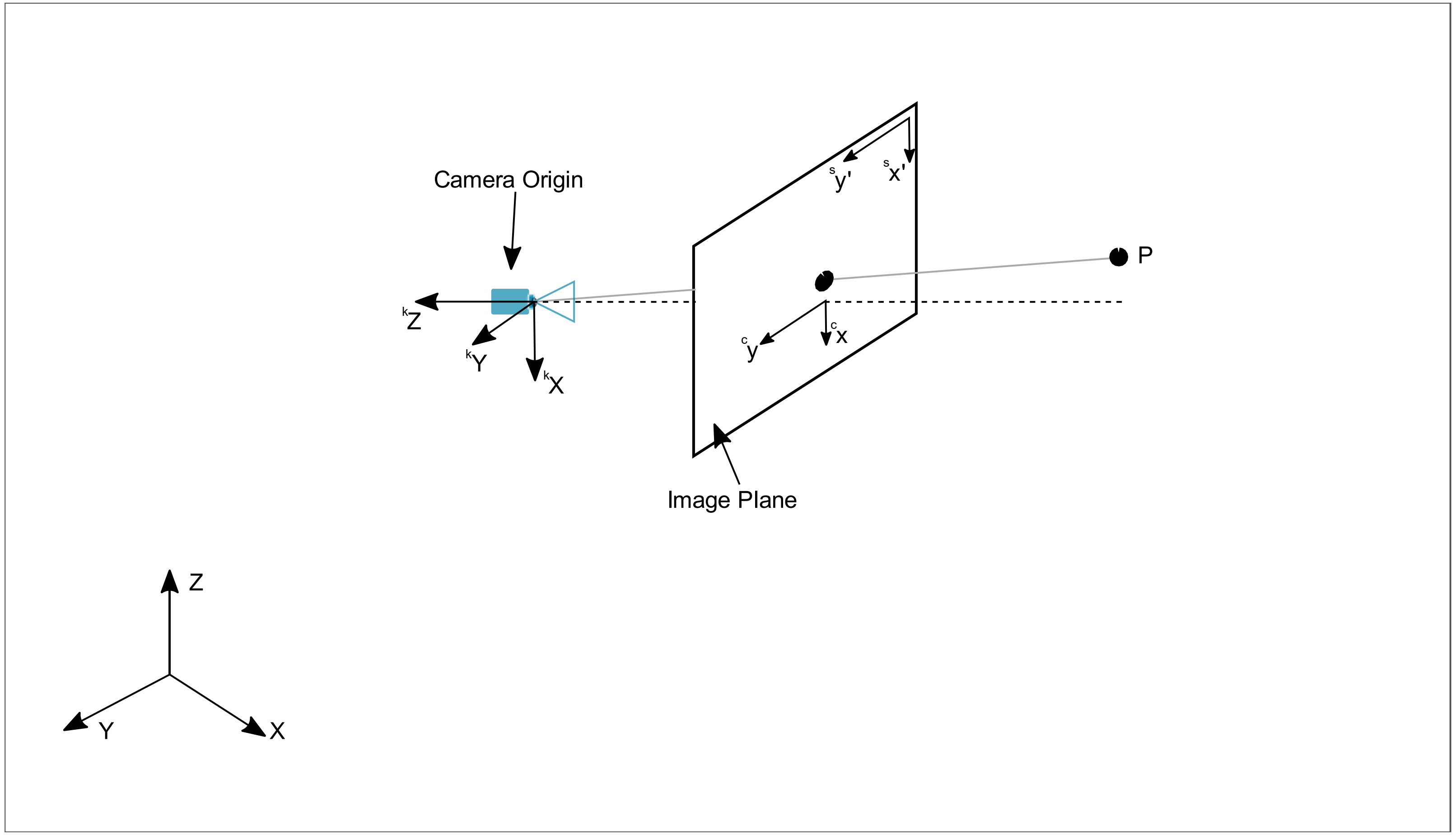


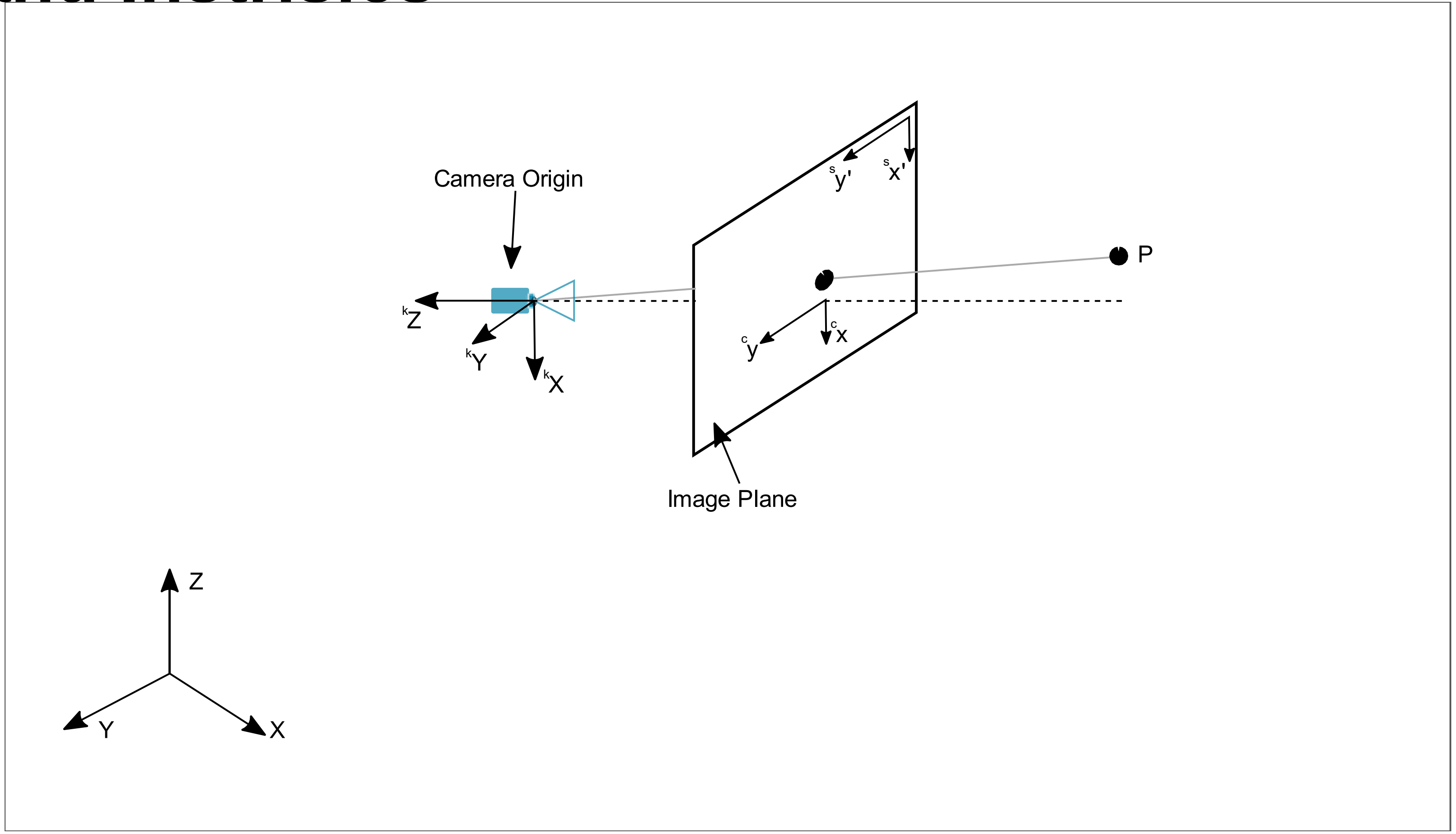
Image Formation

- Let’s look at
  the most simplistic case.
  What’s weird about it?
- Let’s Fix that
- Let’s get some notation



Camera Extrinsics and Instrisics

- The orientation of the camera wrt Word coordinates is called extrinsics
- The rest of Parameters are called intrinsics
- We will see how points in 3D are projected on the sensor

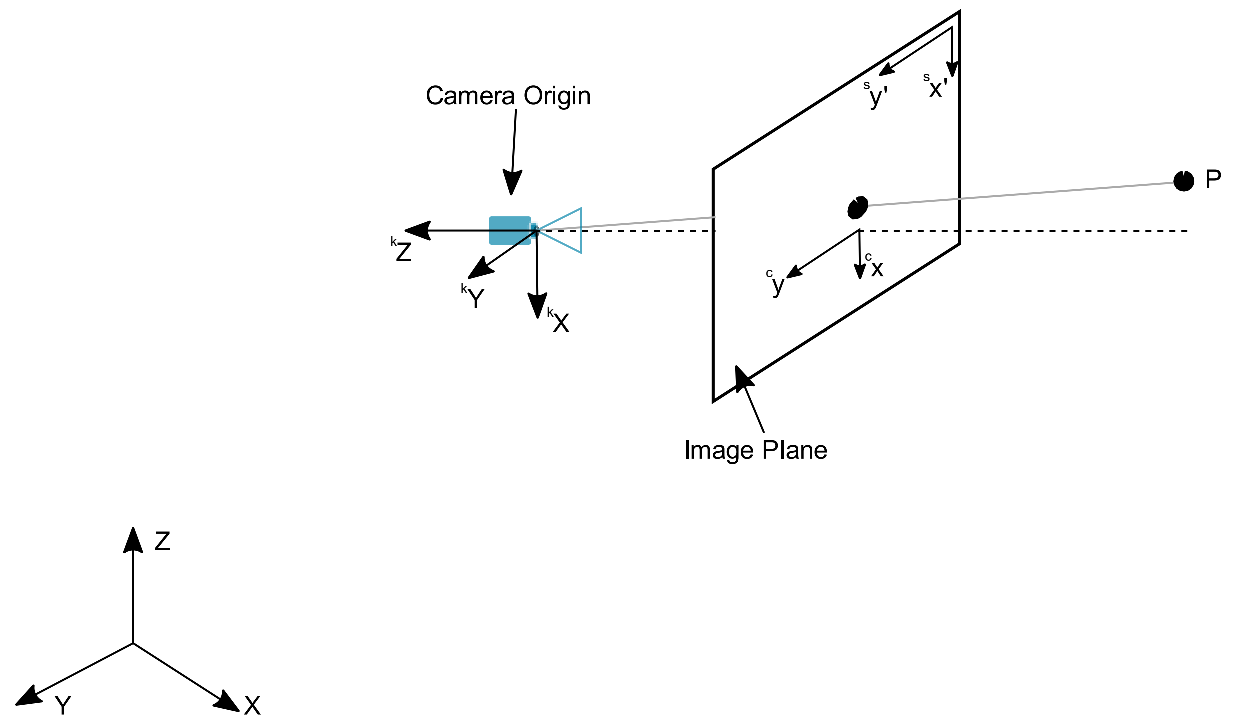


Camera Extrinsics

- We need to express a 3D point P
  $\boldsymbol{x}_p = [x_p \quad y_p \quad z_p]^T$
  in the camera coordinates
  $^k\boldsymbol{x}_p = [\,^k x_p \quad ^k y_p \quad ^k z_p]^T$
- To do so we need to use the camera Origin
  $^k\boldsymbol{x} = [\,^k x \quad ^k y \quad ^k z\,]^T$
- Then:

$$\begin{bmatrix} \boldsymbol{x_p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^w\boldsymbol{R}_c & {}^w\boldsymbol{t}_c \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} {}^k\boldsymbol{x_p} \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} {}^k\boldsymbol{x_p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^w\boldsymbol{R}_c{}^T & -{}^w\boldsymbol{R}_c{}^T\,{}^w\boldsymbol{t}_c \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_p} \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} {}^k\boldsymbol{x_p} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{3x3} & \boldsymbol{t} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_p} \\ 1 \end{bmatrix}$$



Camera Extrinsics

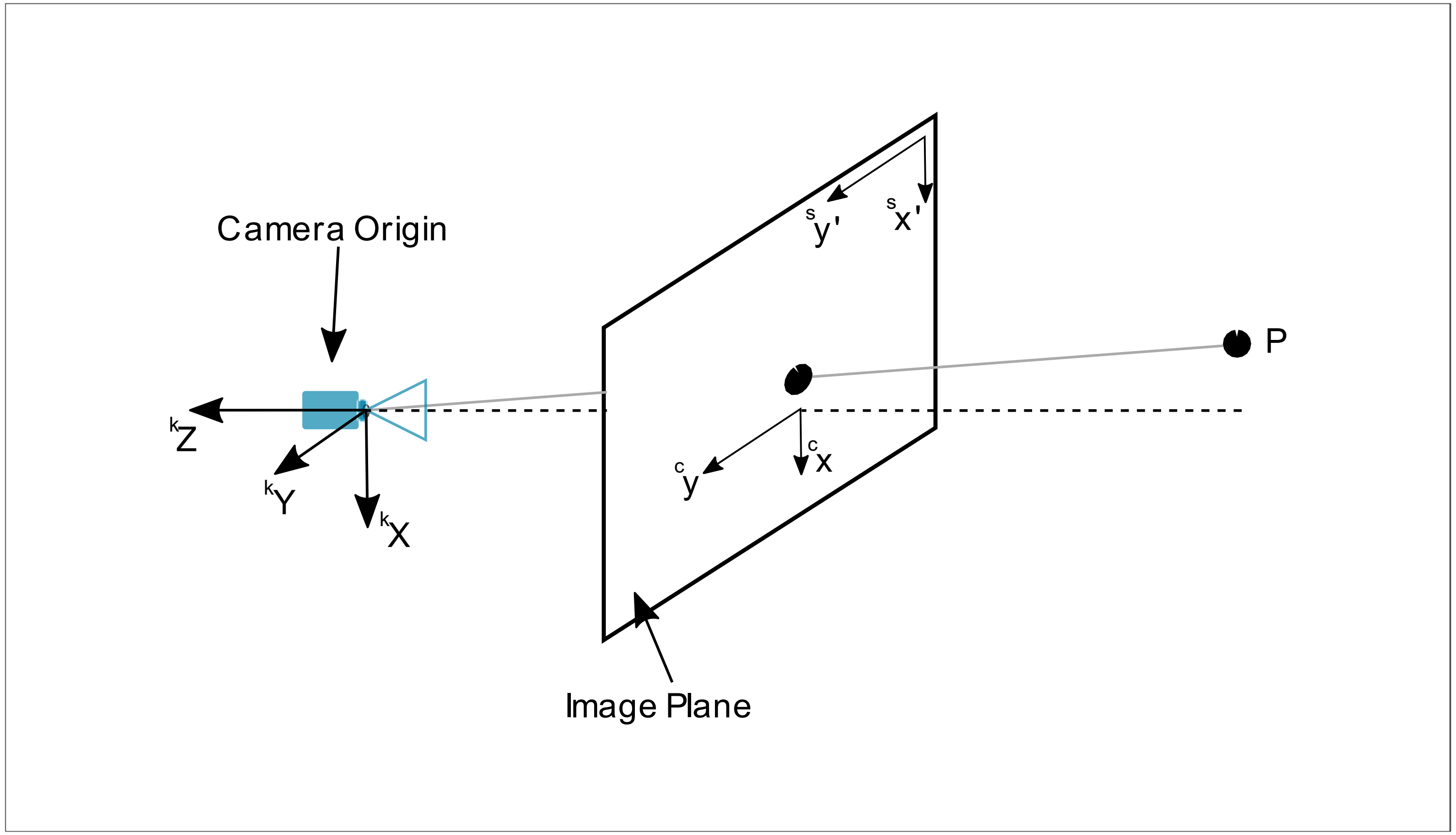
- So now we have a description for the 3D point in camera frame

$$\begin{bmatrix} {}^k x_p & {}^k y_p & {}^k z_p & 1 \end{bmatrix}^T = \underbrace{\begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{0}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \boldsymbol{I}_{3x3} & \boldsymbol{t}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}} \underbrace{\begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix}^T}_{\text{Homogeneous Point In World Coordinates}}$$

- To get to the image frame we will have to drop a dimension

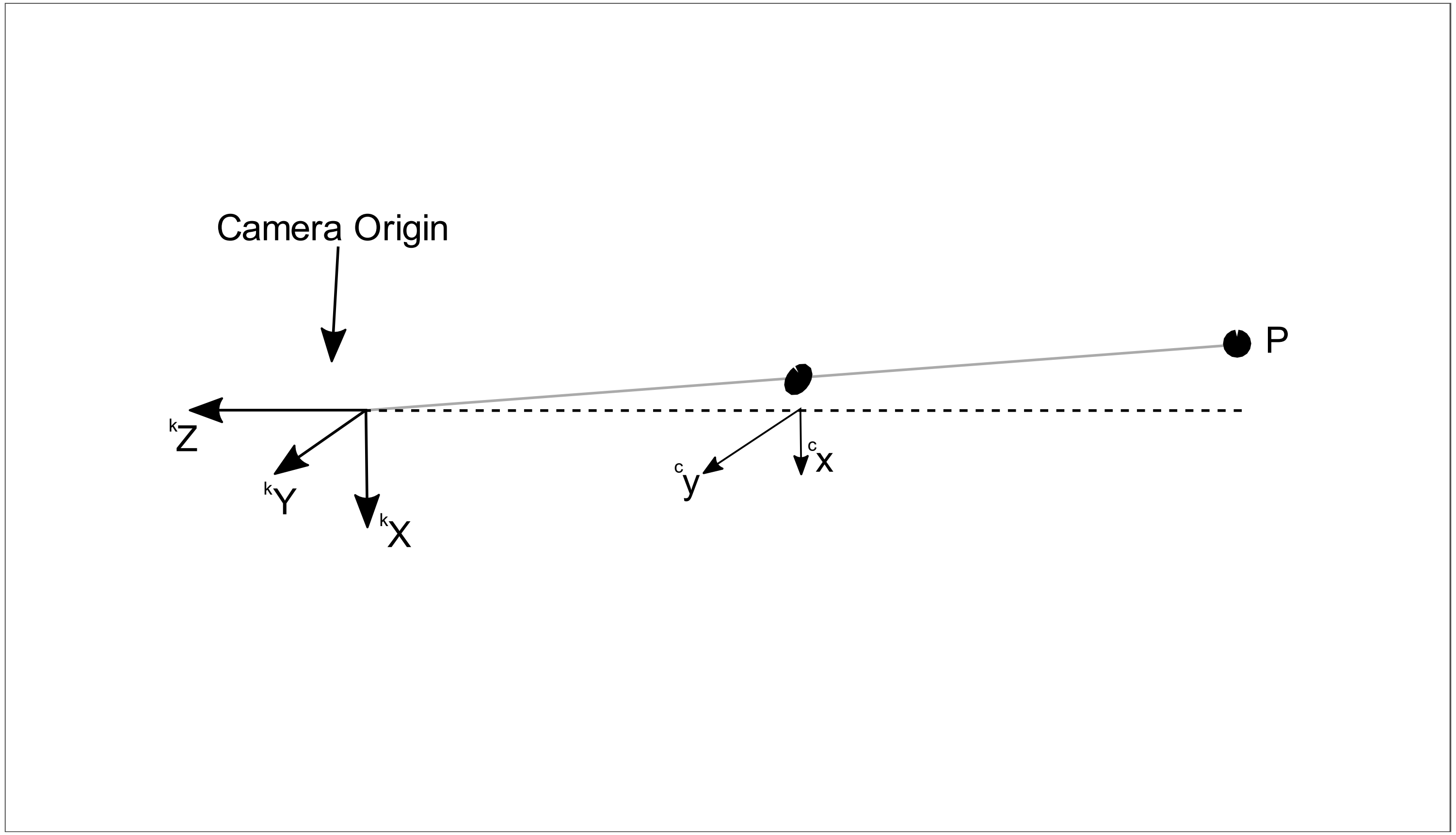
Projection

- Let's get rid of things we don't need



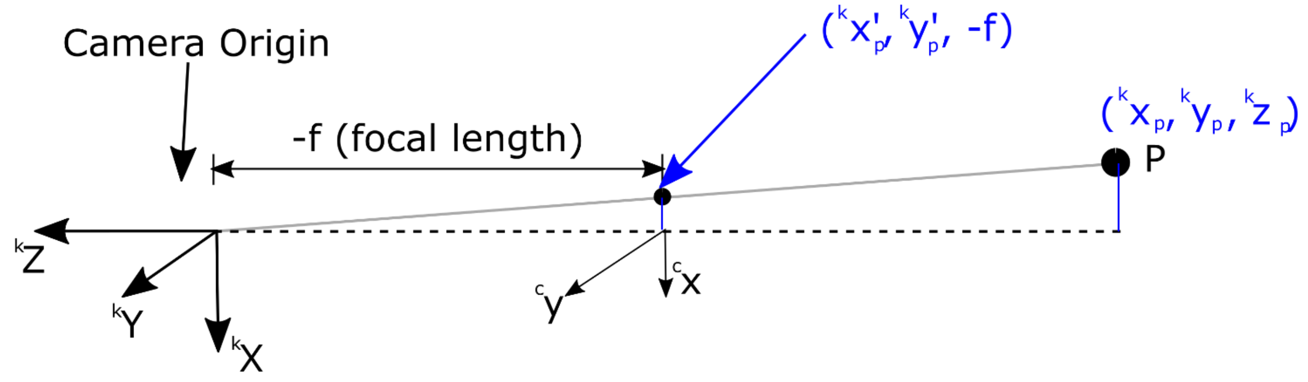
Projection

- Let's get rid of things we don't need



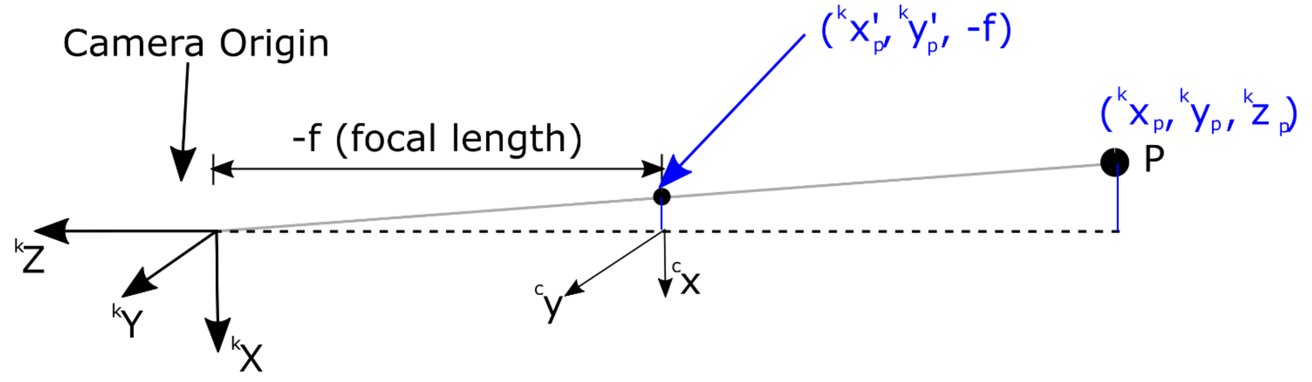
Projection

- Let's get rid of things we don't need



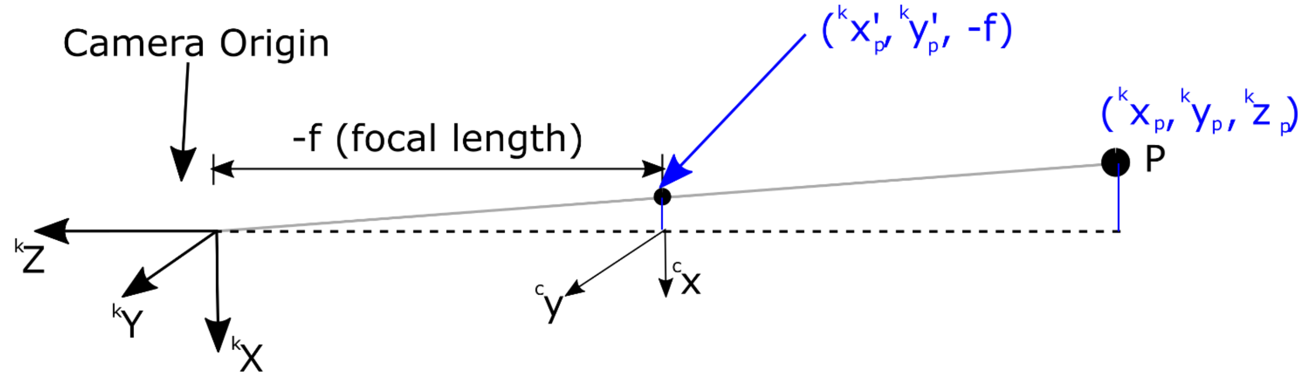
Projection

- Let's get rid of things we don't need
- So it is clear that the position of the projection on the image frame is dependent on the ratio x/z



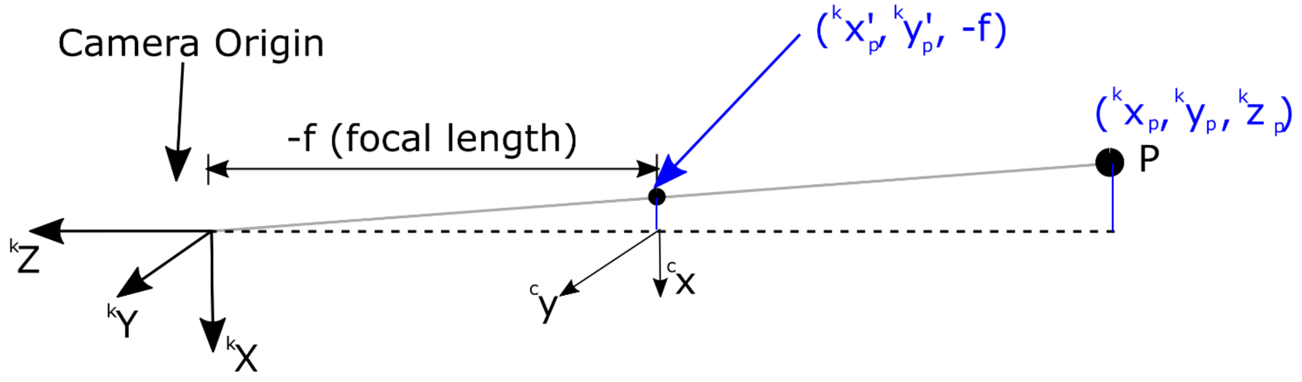
Projection

- Let's get rid of things we don't need
- So it is clear that the position of the projection on the image frame is dependent on the ratio x/z
- In fact ${}^{c}x_p = -f * {}^{k}x_p / {}^{k}z_p$ and ${}^{c}y_p = -f * {}^{k}y_p / {}^{k}z_p$



Projection

- Let's get rid of things we don't need
- So it is clear that the position of the projection on the image frame is dependent on the ratio x/z
- In fact ${}^{c}x_p = -f * {}^{k}x_p / {}^{k}z_p$ and ${}^{c}y_p = -f * {}^{k}y_p / {}^{k}z_p$
- Additionally we can add the transformation to the sensor frame



Projection

- From the previous

$$\begin{bmatrix} {}^{k}x_p & {}^{k}y_p & {}^{k}z_p & 1 \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{3x3} & \boldsymbol{t}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix}^T$$

rotation   translation   Homogeneous Point In World Coordinates

- We get to the following

Camera Matrix

$$[{}^{s}\mathrm{x}, {}^{s}\mathrm{y}, \mathrm{w}]^{\mathrm{T}} = \boxed{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{t}_{3x1} \end{bmatrix} \begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix}^T$$

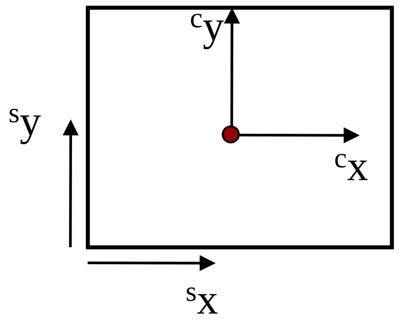
Intrinsics   exttrinsics   Homogeneous Point in World Coordinates

- Where W is the scale
- **This is the analytic version of the projection matrix**

Projection Matrix

- We can define the Projection Matrix as follows

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



- **This is great news for Calibration!**
- It means that our problem is a system of linear equations
- How many unknowns?
- 11: (5 + 6) Intrinsic+Extrinsic

Notation Disclaimer: In this slide X Y Z refer to 3D world points (not matrices) and x,y to 2D camera points

Calibration – Known 3D positions

- We defined our camera projection as follows:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- If we “sample” a set of points in 3D and 2D: [X, Y, Z] → [x,y] we get the following analytic equations:

$$\begin{aligned} \lambda x &= P_{11}X + P_{12}Y + P_{13}Z + P_{14} \\ \lambda y &= P_{21}X + P_{22}Y + P_{23}Z + P_{24} \\ \lambda &= P_{31}X + P_{32}Y + P_{33}Z + P_{34} \end{aligned}$$

- However, λ is used for scale and therefore does not provide a 3rd equation

Notation Disclaimer: In this slide X Y Z refer to 3D world points (not matrices) and x,y to 2D camera points

Calibration – Known 3D positions

- So for every 3D to 2D correspondence we get 2 equations:

$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14}$$
$$(P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = P_{21}X + P_{22}Y + P_{23}Z + P_{24}$$

- How many points do we need?
- 6 Points x 2 equations

Notation Disclaimer: In this slide X Y Z refer to 3D world points (not matrices) and x,y to 2D camera points

- For every point we get the following

$$0 = p_{11}X + p_{12}Y + p_{13}Z + p - p_{31}xX - p_{32}xY - p_{33}xZ - p_{34}x$$

$$0 = p_{21}X + p_{22}Y + p_{23}Z + p_{24} - p_{31}yX - p_{32}yY - p_{33}yZ - p_{34}y$$

- We can format all the linear equations in a single matrix: Ap=0

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_nX_n & -x_nY_n & -x_nZ_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_n & -x_nZ_n & -x_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Notation Disclaimer: In this slide X Y Z refer to 3D world points (not matrices) and x,y to 2D camera points

Calibration – Known 3D positions

- The solution is given by the following process which is itself called DLT:
  - Calculate the Singular Value decomposition (SVD)

    SVD : $A = UDV^T$

  - *Get the last column of V*

    $P = V_{smallest}$ *(column of V corr. to smallest singular value)*

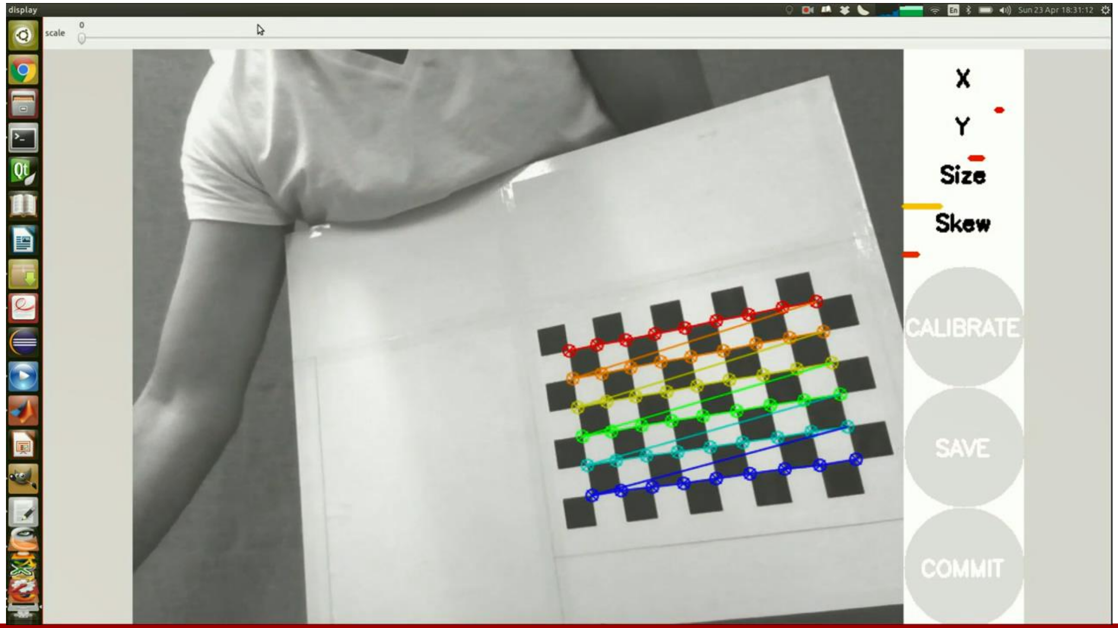
  -Reshape into the P matrix

Calibration – Known 3D positions

- To summarize:
  - Get min 6 3D – 2D correspondences
  - Form matrix A (AP=0)
  - Calculate SVD
  - Get Last column of V which corresponds to the values of P

Calibration in real world

- In Real World we do not have 3D points.
- Or do we?
- What if we could use some sort of predefined setup which would allow us to know the relative position of 3D points?
- It turns out that if we have a calibration pattern we can know their relative positions of all points as they are on a known configuration



Calibration using homography

- Since Z = 0 we lose one degree of freedom:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

- We can then redefine our P as:

$$\equiv \begin{bmatrix} P_{11} & P_{12} & P_{14} \\ P_{21} & P_{22} & P_{24} \\ P_{31} & P_{32} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

- This is called homography and allows us to project points from one plane to another

Notation Disclaimer: In this slide X Y Z refer to 3D world points (not matrices) and x,y to 2D camera points

Calibration using homography

- How to calculate the homography:

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \qquad \mathbf{x}' = \begin{bmatrix} w'x' \\ w'y' \\ w' \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

- The homography has 9 elements. However, similar to before due to the scale there are only 8 unkowns
- Therefore, we need minimum 4 correspondences to setup the problem as a DLT

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & xx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & yx' & yy' & y' \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \mathbf{0}$$

Calibration using homography

Steps

- With a static camera move the pattern and get images.
  - Make sure to not present only one orientation, as there will be a problem with the estimation of the homography
- Calculate correspondences using the pattern
- Solve using **Normalized DLT**
  - Normalize coordinates for each image
    - Translate for zero mean
    - Scale so that average distance to origin is ~sqrt(2)
  - Form Matrix A
  - Solve using SVD
    - Calculate SVD
    - Use last column of V to get the homography.
  - Denormalize

Calibration using homography

- However the DLT solution is known to be prone to outliers.
- Other approaches exist to solving the calibration:
  - Nonlinear least squares
  - RANSAC
- RANSAC:

1. Choose number of samples *N*
2. Choose 4 random potential matches
3. Compute **H** using normalized DLT
4. Project points from **x** to **x**’ for each potentially matching pair:
5. Count points with projected distance $<$ t
6. Repeat steps 2-5 *N* times

Choose **H** with most inliers

Calibration - Distortion

- Ok, but still, What about the lens Distortion?

Barrel

Pincushion

Fisheye

Calibration - Distortion

- So how do we model it:

30

Calibration - Distortion

- So how do we model it:
- Usually a quartic (biquadratic) polynomial is used:

$$\begin{aligned} \hat{x}_c &= x_c(1+\kappa_1 r_c^2+\kappa_2 r_c^4) \\ \hat{y}_c &= y_c(1+\kappa_1 r_c^2+\kappa_2 r_c^4) \end{aligned}$$

- OR higher order

30

Calibration - Distortion

- So how do we model it:
- Usually a quartic (biquadratic) polynomial is used:
- OR h

Distorted

Undistorted

Calibration - Distortion

- How to calibrate for the distortion coefficients:
  - First do DLT for the Projection Matrix
  - Then form a non linear least square problem that includes both the linear projection and the non-linear distortion using the Levenberg Marquardt algorithm.

Conclusion

- What did we learn?

Perception for Autonomous Systems 31392:

Camera Matrix and Camera Calibration

Lecturer: Evangelos Boukas—PhD