Geometry in Computer Vision

Yaroslava Lochman

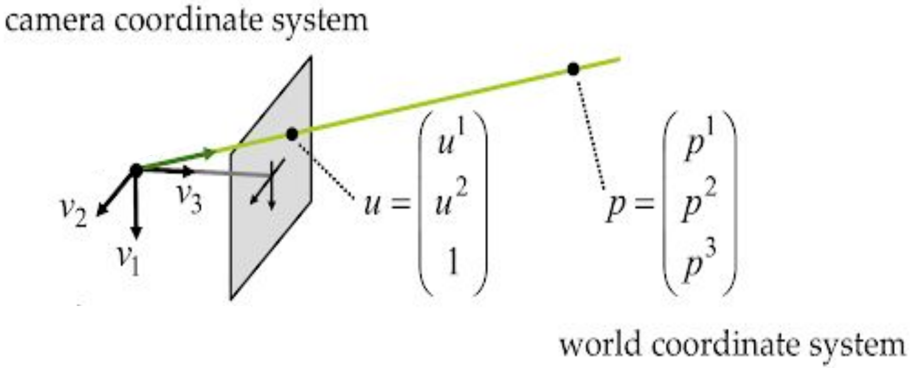
Dec 2020

Upon module completion...

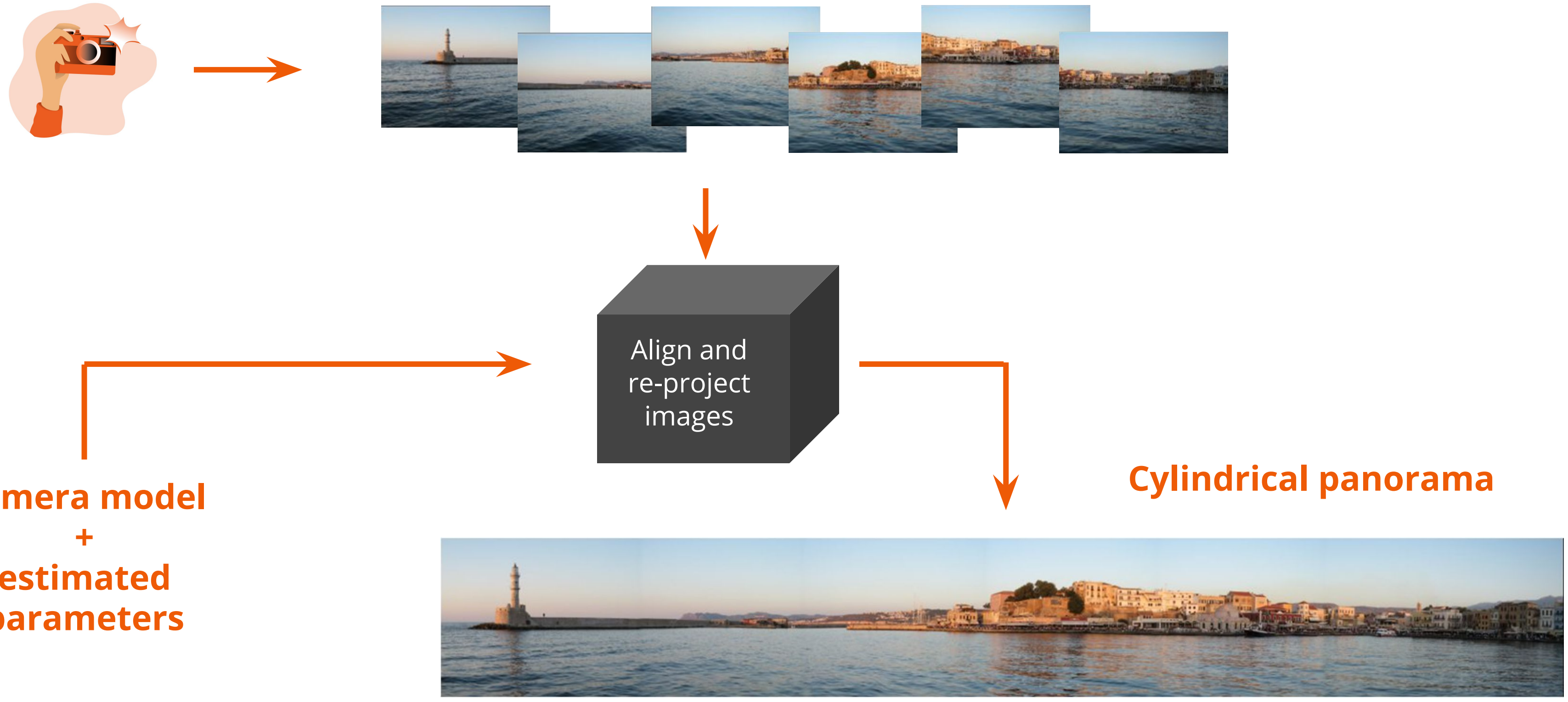
Calibration
or
Auto-Calibration

Assumptions on camera

camera model
+
estimated parameters



Upon module completion...



Panoramic Image Projections

Cube Projection

Stereographic Projection

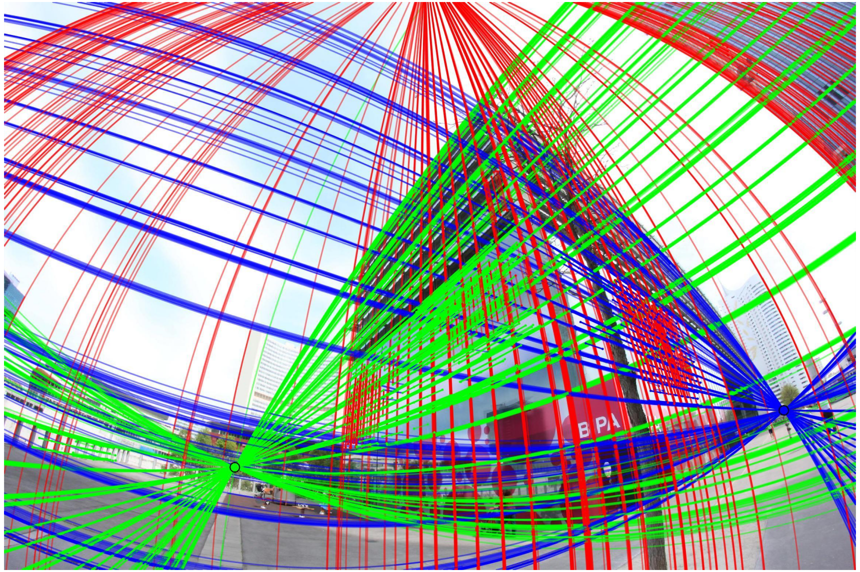
Single Image Correction

Undistortion

Single Image Correction

Scene Plane Rectification

Scene Geometry Parsing



Lines of orthogonal directions

Points on orthogonal planes

Augmented Reality

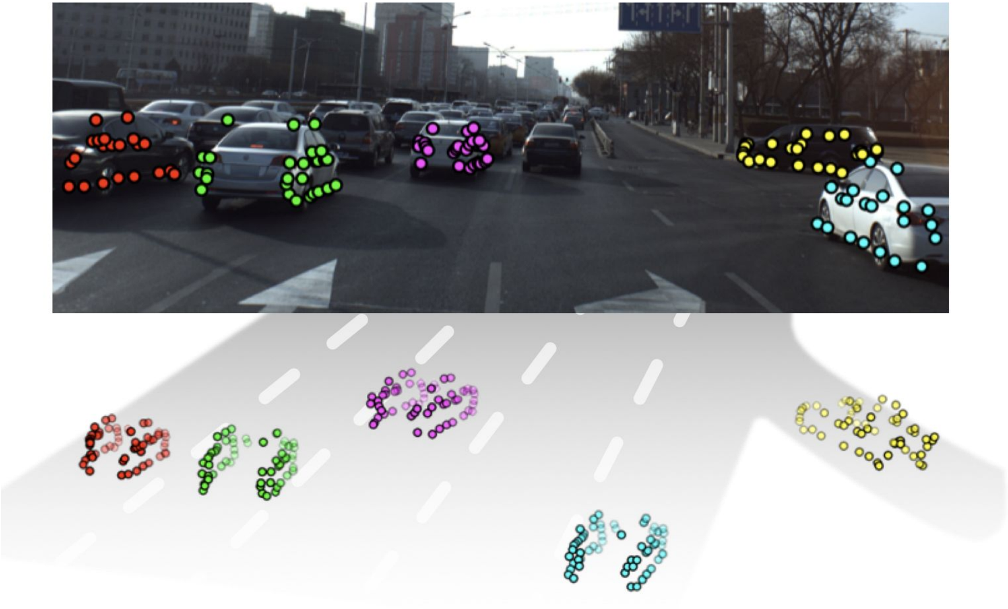
Project artificial objects onto the image

Rigid Structure from Motion

Sequence of 2D images of an object from N viewpoints

3D structure of an object

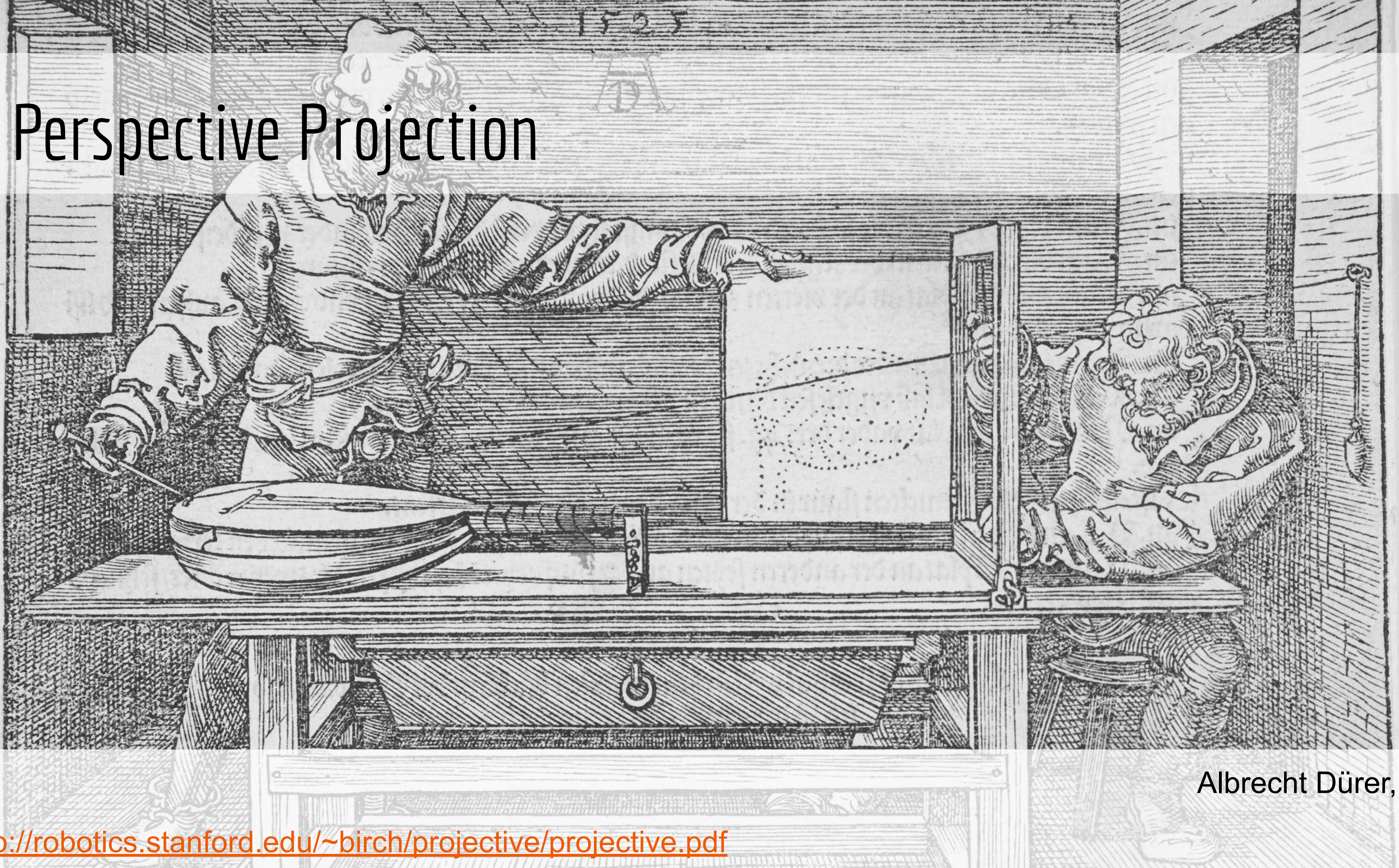
Non-Rigid Structure from Motion



3D reconstruction from a sequence of frames
where the object can deform

Camera Calibration

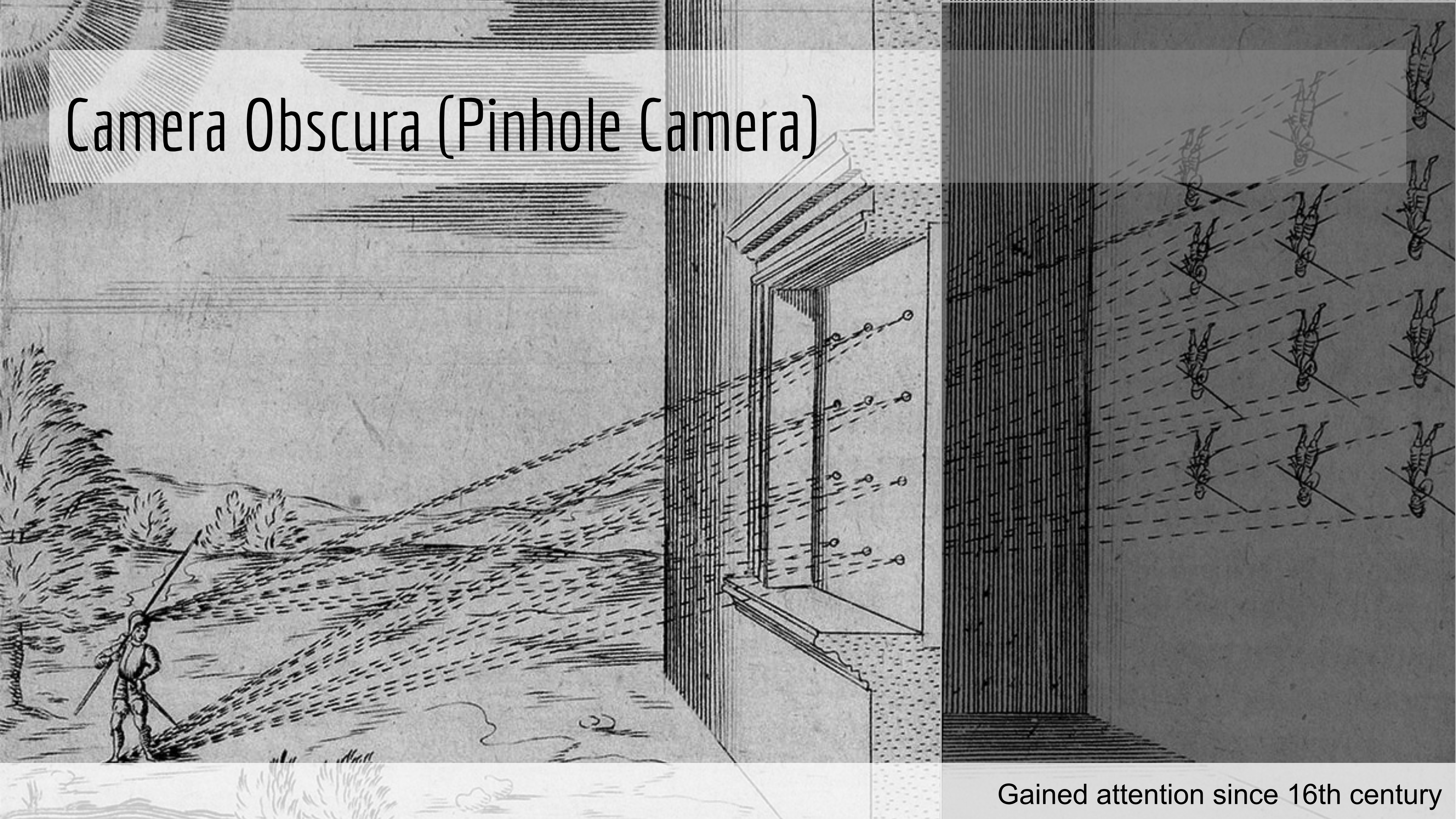
Perspective Projection



Albrecht Dürer, 1525

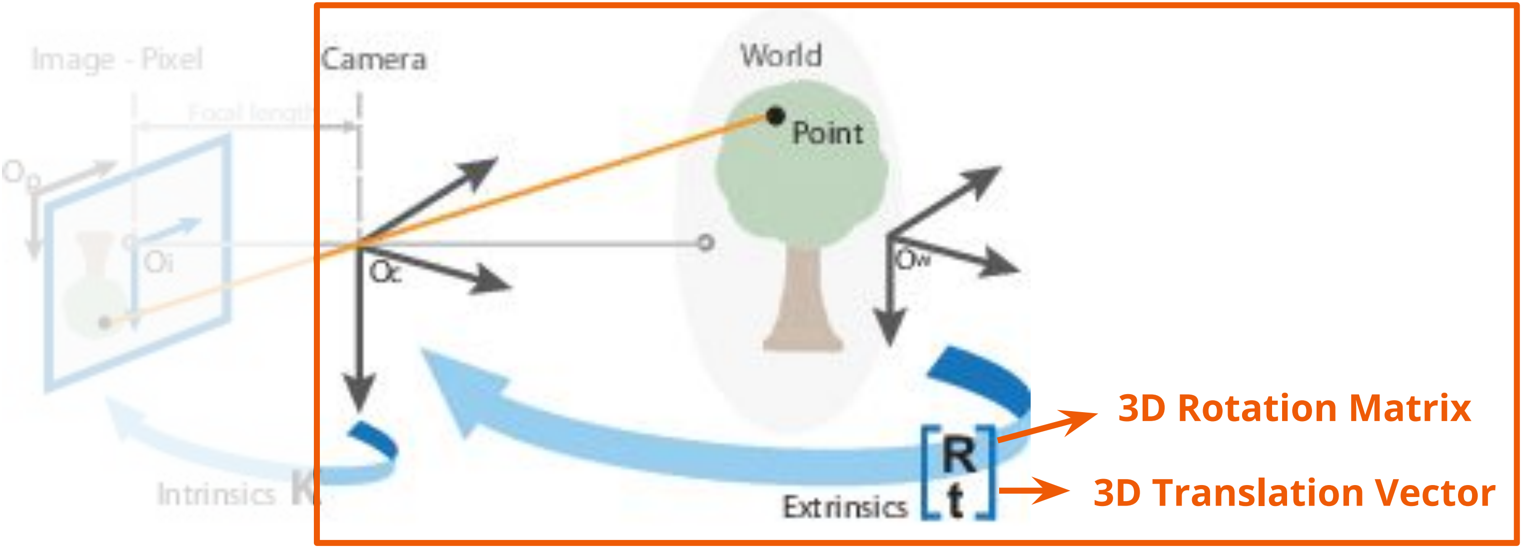
http://robotics.stanford.edu/~birch/projective/projective.pdf

Camera Obscura (Pinhole Camera)



Gained attention since 16th century

Extrinsic Parameters of the Camera

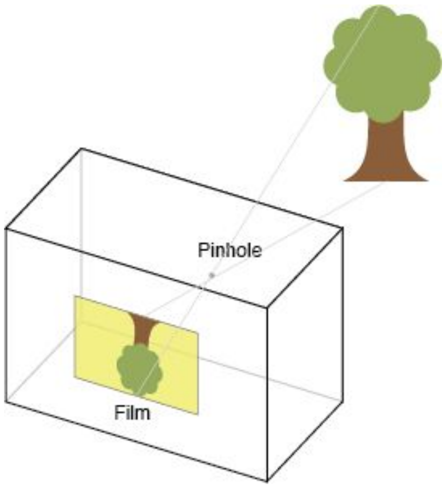


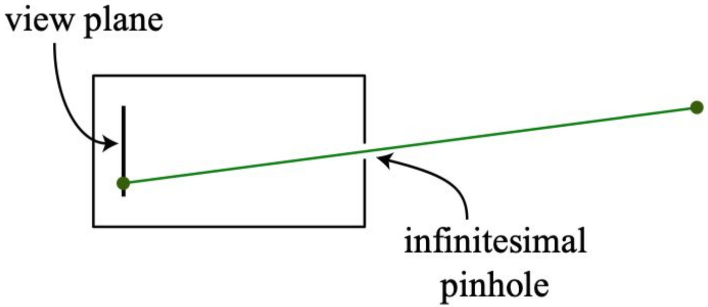
Can be seen as **a change of basis** from the ("global") world coordinate system to the ("local") camera coordinate system

Change of Basis in 3D

$$\mathbf{X}_{cam} = \mathrm{R}\mathbf{X} + \mathbf{t} = [\mathrm{R}|\mathbf{t}] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

Pinhole Camera Model

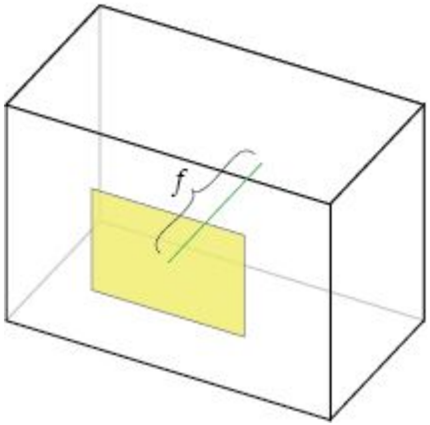




A pinhole camera is an idealization of the thin lens as aperture shrinks to zero.
Pinhole camera model is the simplest we can assume.

Check out this awesome lecture on pinholes and lenses:
http://graphics.cs.cmu.edu/courses/15-463/lectures/lecture3.pdf

Focal Length in Pinhole Camera



Focal Length & Perspective

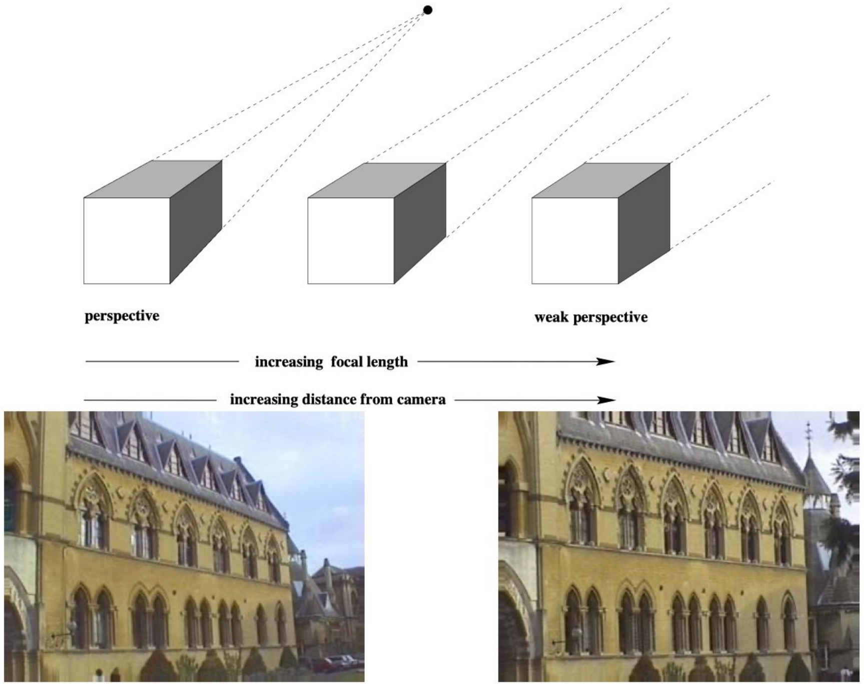
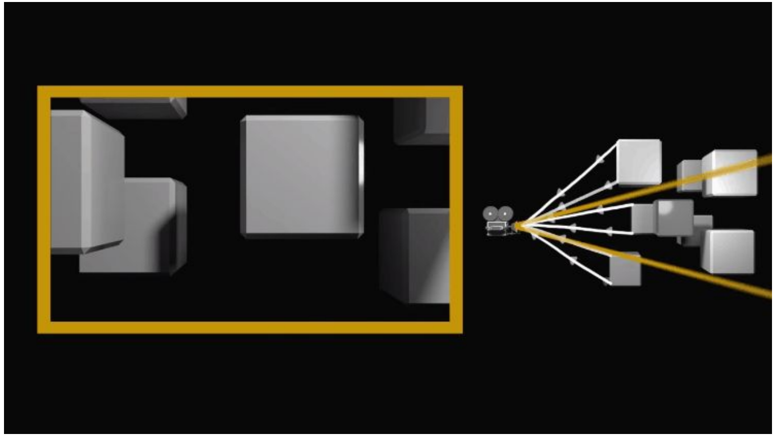
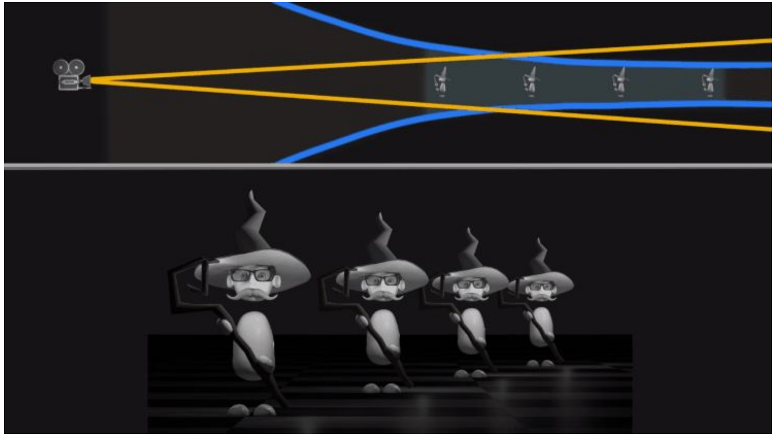
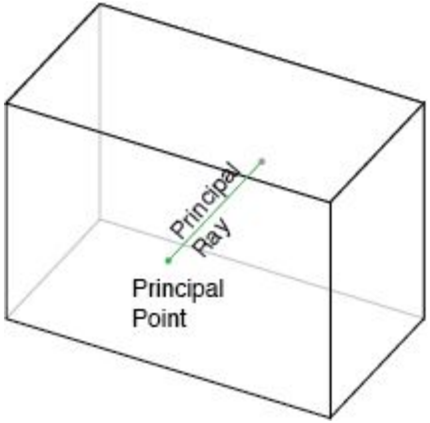


Fig. 6.7. *As the focal length increases and the distance between the camera and object also increases, the image remains the same size but perspective effects diminish.*

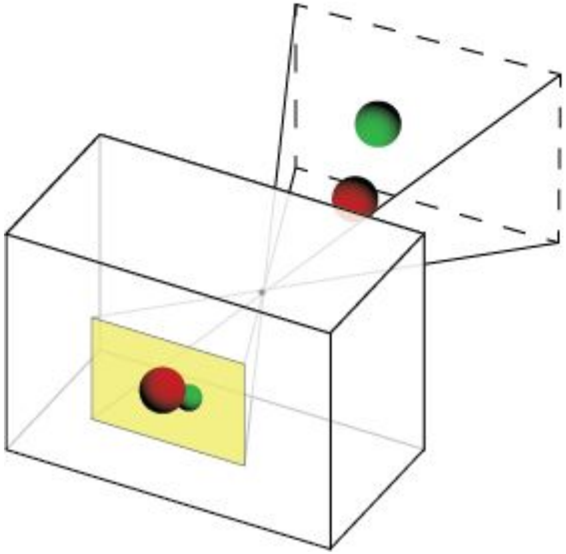




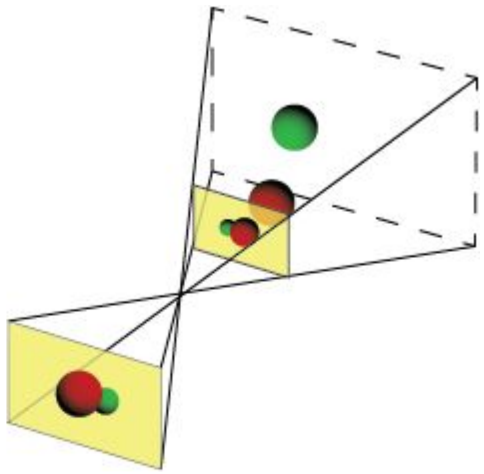
Principal Point (PP)



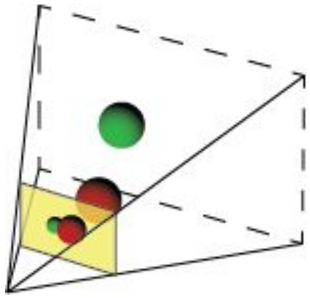
Viewing Frustrum



Viewing Frustrum



Viewing Frustrum

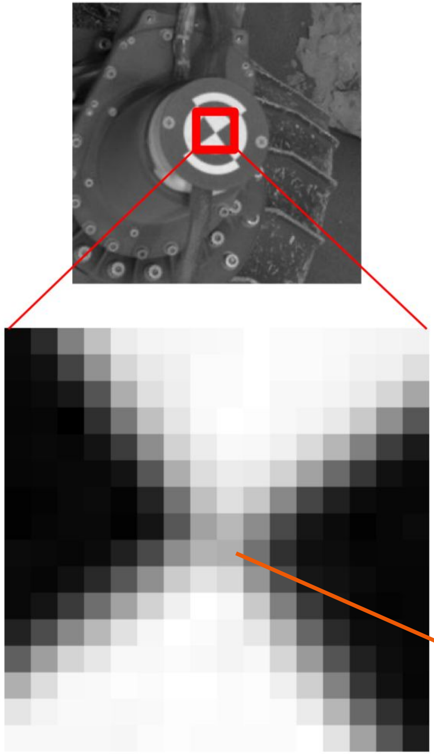


Pinhole Camera Model

$$\mathbf{x} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathtt{R}|\mathbf{t}] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

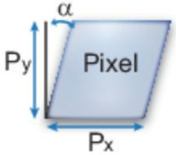
$$\begin{cases} u = x/z \\ v = y/z \end{cases}$$

Image Units



Film
(continuous, mm) → Image
(discrete, pixels)

Orthogonal Raster, Unit Aspect (ORUA)
or **square pixels**



non-square pixels
with **skew** and **non-unit aspect**

Anamorphic Format

CCD (Perspective) Camera Model

$$\mathbf{x} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \left[\mathbf{R} | \mathbf{t} \right] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

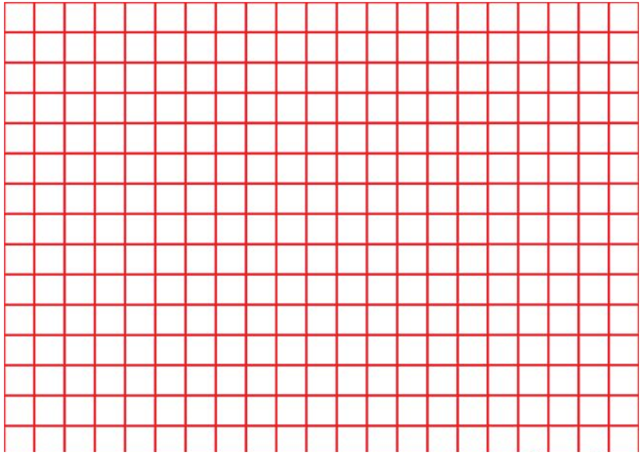
$$\begin{cases} u = x/z \\ v = y/z \end{cases}$$

Finite Projective Camera Model

$$\mathbf{x} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \left[\mathtt{R}|\mathbf{t}\right] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

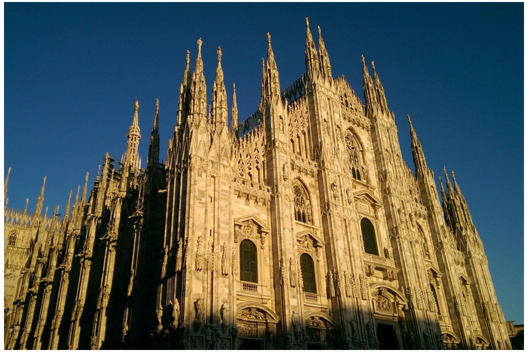
$$\begin{cases} u = x/z \\ v = y/z \end{cases}$$

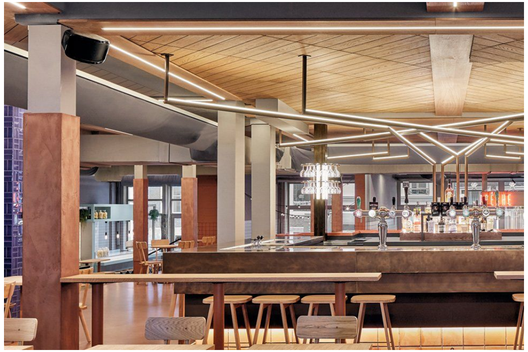
Rectilinear Images



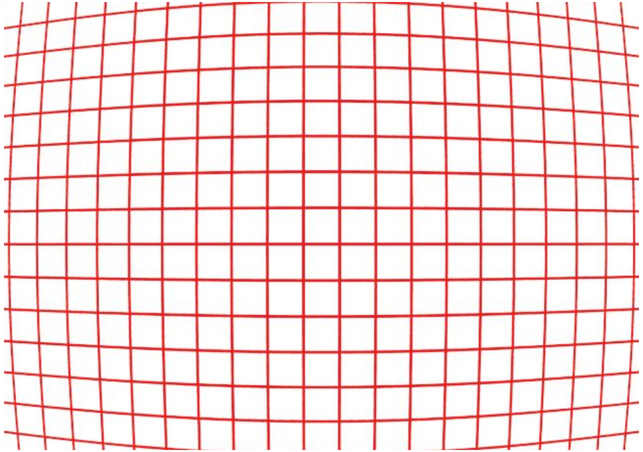
NO DISTORTION

Pretty unrealistic
Still, often assumed when the lens system introduces insignificant amount of distortion





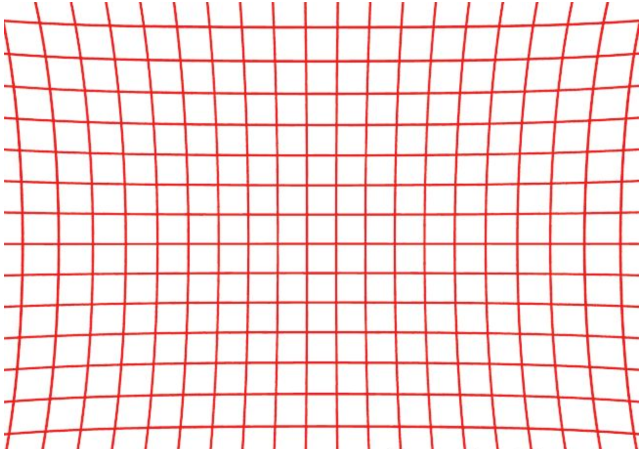
(Radial) Barrel-Distorted Images



"BARREL" RADIAL DISTORTION

Arises mostly when wide-angle or fisheye lens system is used

(Radial) Pincushion-Distorted Images



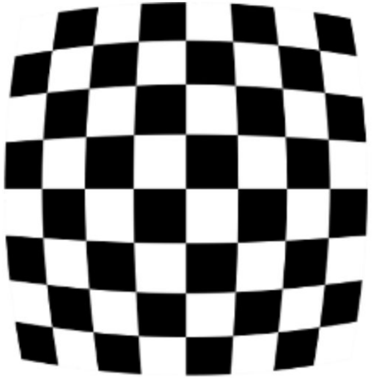
“PINCUSHION” RADIAL DISTORTION

Rarely arises alone but usually compensates barrel-distortion
This leads to less distortions but it usually becomes more complex

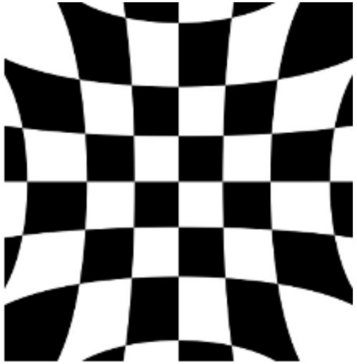
Brown-Conrady Model (radial part)

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

where $r^2 = x^2 + y^2$



Barrel if
$k_1 > 0$
$k_2 = 0$
$k_3 = 0$
(e.g.)



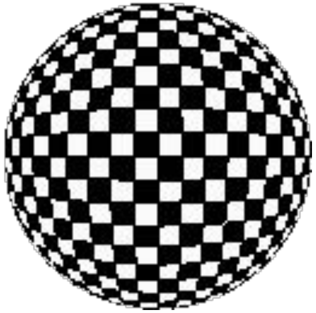
Pincushion if
$k_1 < 0$
$k_2 = 0$
$k_3 = 0$
(e.g.)

Widely used for near-rectilinear cameras

Doesn't model severe radial distortion like fisheye

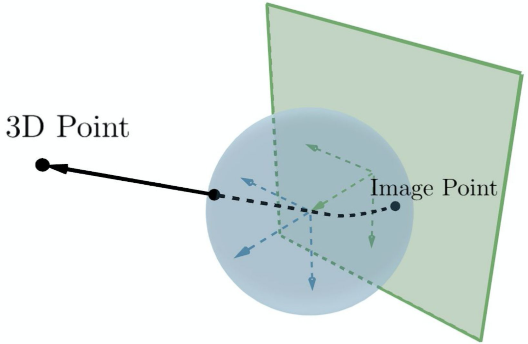
* all points are distortion center subtracted

Kannala-Brandt (Fisheye) Model



$$\theta_d = \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8)$$

$$\gamma\tilde{\mathbf{x}} = (x, y, \frac{r}{\theta_d})^\top$$



Can model various radial distortions including fisheye

Difficult to use in estimation frameworks

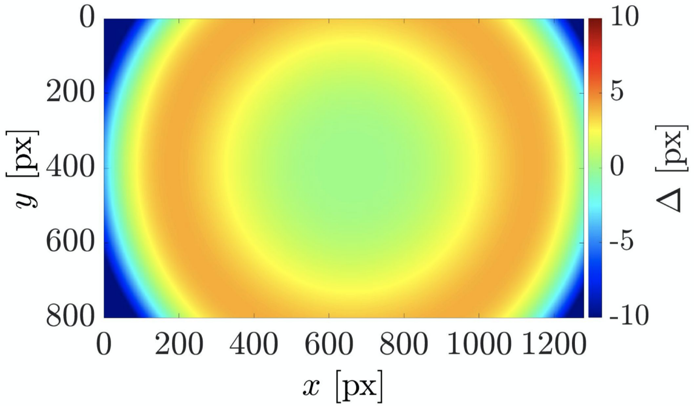
* all points are distortion center subtracted

One-parameter Division Model

$$\gamma \mathbf{x} = (\tilde{x}, \tilde{y}, 1 + \lambda(\tilde{x}^2 + \tilde{y}^2))^\top$$

It has only 1 parameter — very attractive in automatic estimation frameworks especially in ill-posed settings

Can model various radial distortions including fisheye but is less accurate



Per pixel error of division model fit to fisheye

* all points are distortion center subtracted

Scaramuzza's Model

DIV model

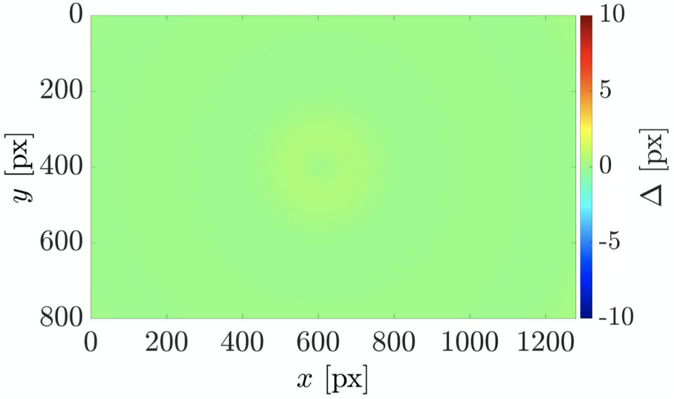
$$\gamma \mathbf{x} = (\tilde{x}, \tilde{y}, \boxed{1} + \boxed{\lambda}(\tilde{x}^2 + \tilde{y}^2))^\top$$

$a_0 \quad a_2$

Sc model

$$\gamma \mathbf{x} = (\tilde{x}, \tilde{y}, a_0 + a_2\tilde{r}^2 + a_3\tilde{r}^3 + a_4\tilde{r}^4))^\top$$

$$\tilde{r} = \sqrt{\tilde{x}^2 + \tilde{y}^2}$$

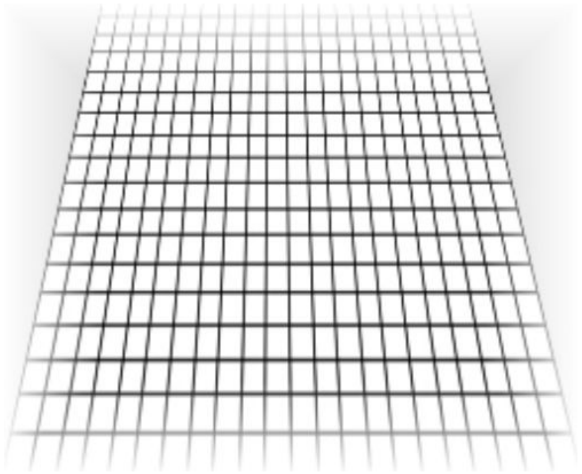


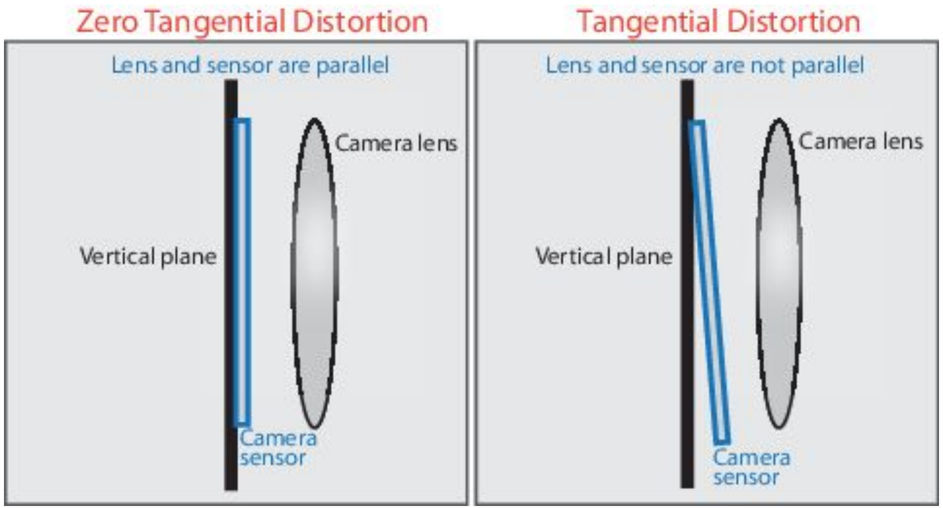
Per pixel error of Scaramuzza's model fit to fisheye

Best of all worlds

* all points are distortion center subtracted

Non-Radial Distortion





TANGENTIAL DISTORTION

Caused by axes misalignments inside the camera
(e.g. between the sensor plane and thelens)

Brown-Conrady Model (tangential part)

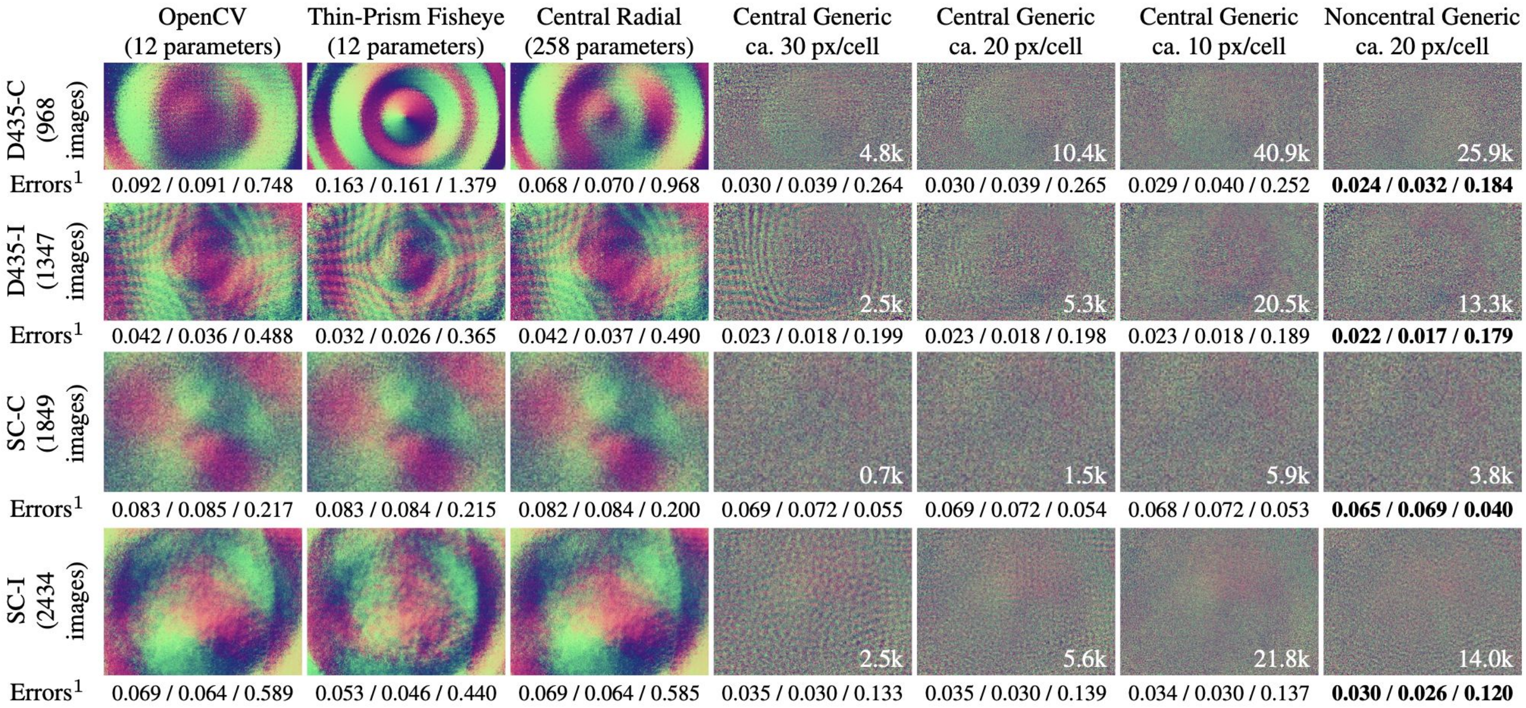
$$x_{distorted} = x + [2p_1xy + p_2(r^2 + 2x^2)]$$
$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2xy]$$

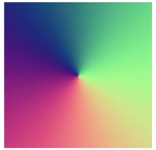
where $r^2 = x^2 + y^2$

Widely used along with radial part

* all points are distortion center subtracted

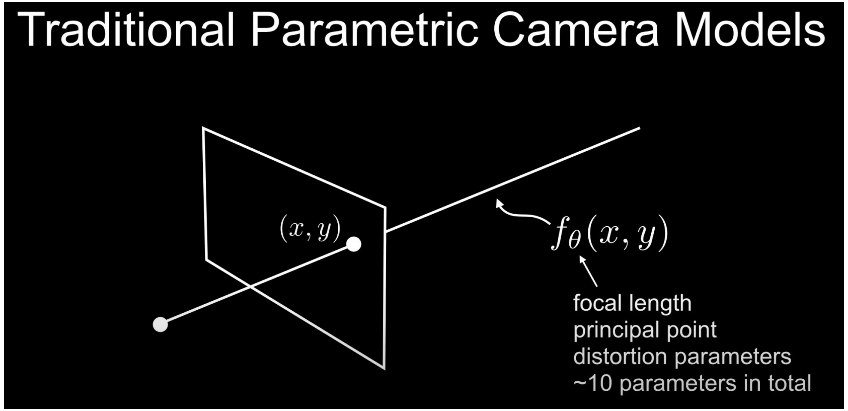
Real Lens Distortion

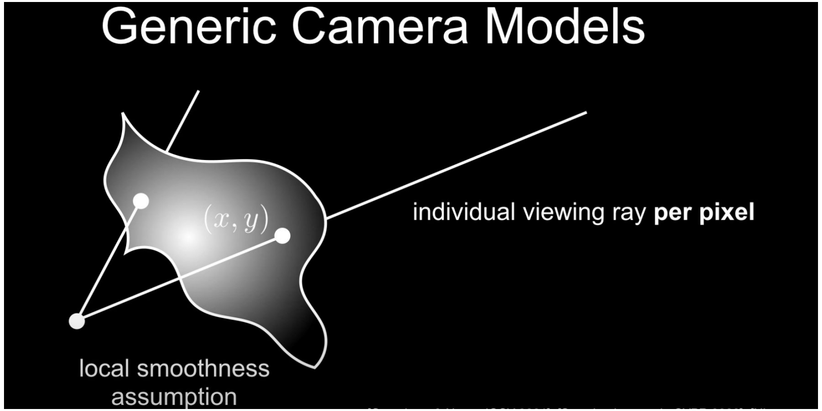




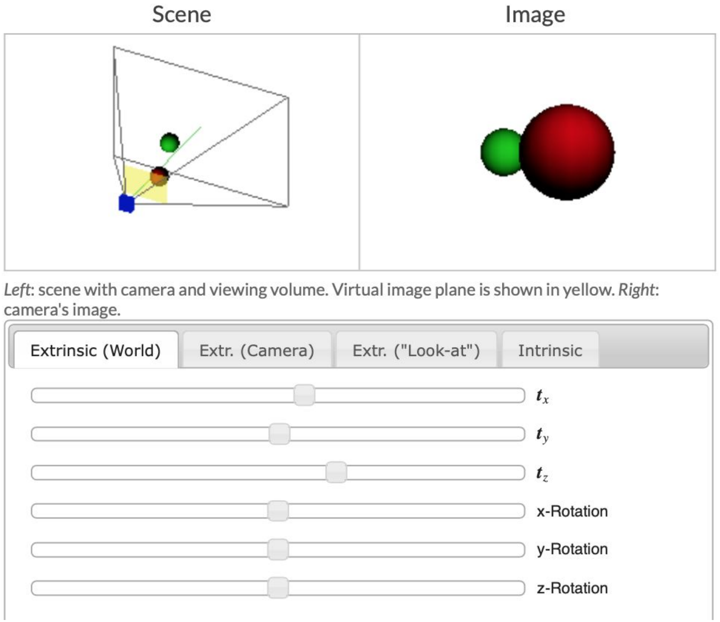
[1] Schöps et al. Why Having 10,000 Parameters in Your Camera Model is Better Than Twelve (2020)

Generic Camera Model





Demo: Simple Synthetic Camera



https://ksimek.github.io/2012/08/22/extrinsic

Real Projective Plane and Homogeneous Representation

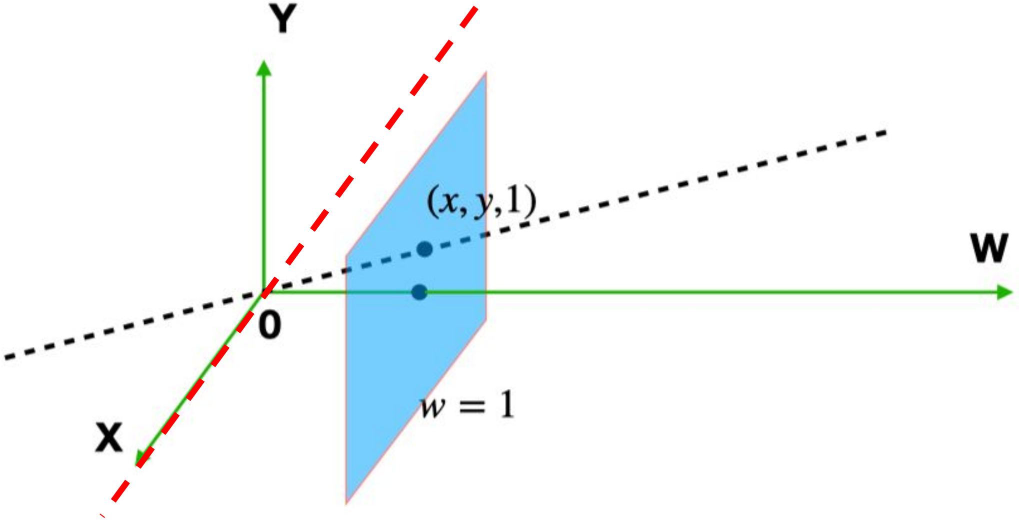
Every **point** in 2D **(x, y)** represents a ray in R^3 along which all points project the same, excluding **0**

Real Projective Plane (RP^2)

Plane (R^2)

$$\begin{pmatrix} x \\ y \end{pmatrix} \leftrightarrow \{\alpha \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \alpha \neq 0\}$$

$$? \leftrightarrow \{\alpha \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}, \alpha \neq 0\}$$



Equivalence relation in $R^3 \setminus \{\mathbf{0}\}$

$$\begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} \sim \begin{pmatrix} x_2 \\ y_2 \\ w_2 \end{pmatrix} \Leftrightarrow \exists \alpha : \begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} = \begin{pmatrix} \alpha x_2 \\ \alpha y_2 \\ \alpha w_2 \end{pmatrix}$$

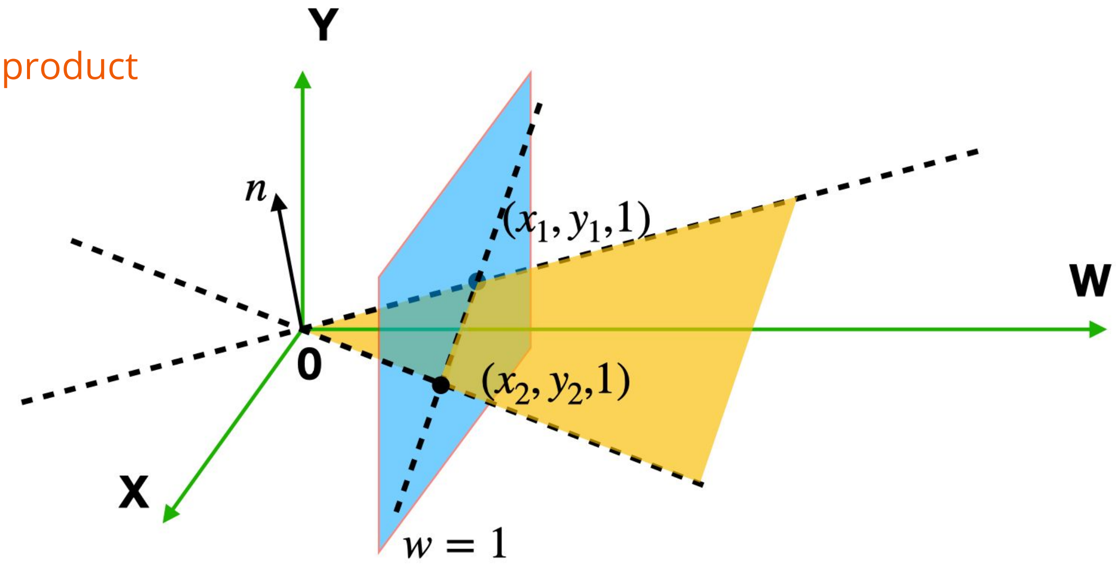
RP^2 — equivalence classes (partitioning) of **$R^3 \setminus \{0\}$**
3D points from one equivalence class project to the same 2D point

Homogeneous Representation of 2D Lines

Every **line** in 2D through **(x_1, y_1), (x_2, y_2)** represents a plane in 3D

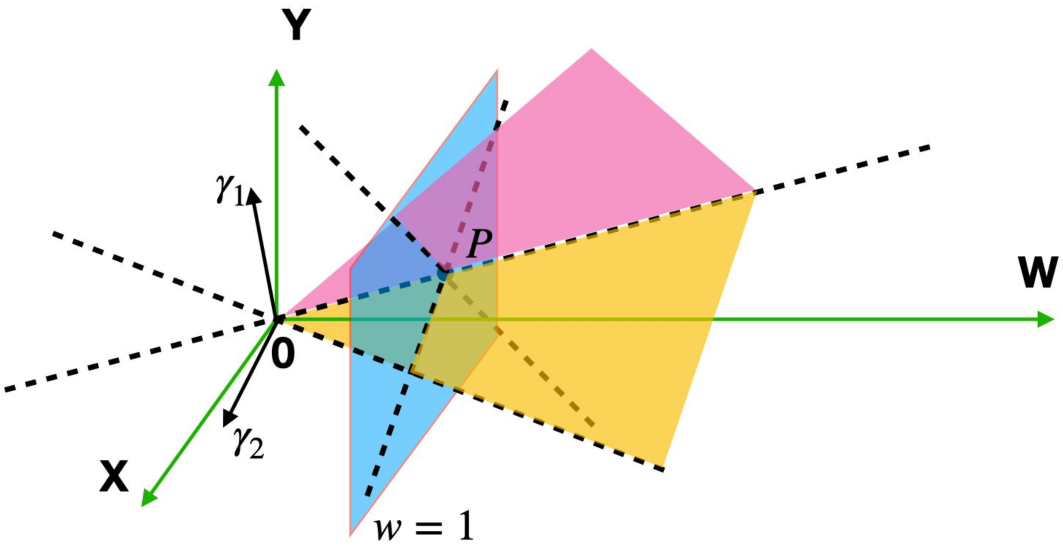
It's a join of **(x_1, y_1)** and **(x_2, y_2)**
mathematically expressed as cross product

$$n \sim \begin{pmatrix} x_1 & y_1 & 1 \end{pmatrix} \times \begin{pmatrix} x_2 & y_2 & 1 \end{pmatrix}$$



Meet of 2D Lines in Homogeneous Coordinates

Intersection of 2 lines $\gamma_1 = (a_1, b_1, c_1)$, $\gamma_2 = (a_2, b_2, c_2)$ is their cross product



$$P \sim \gamma_1 \times \gamma_2 = (b_1c_2 - c_1b_2, c_1a_2 - a_1c_2, a_1b_2 - b_1a_2)$$

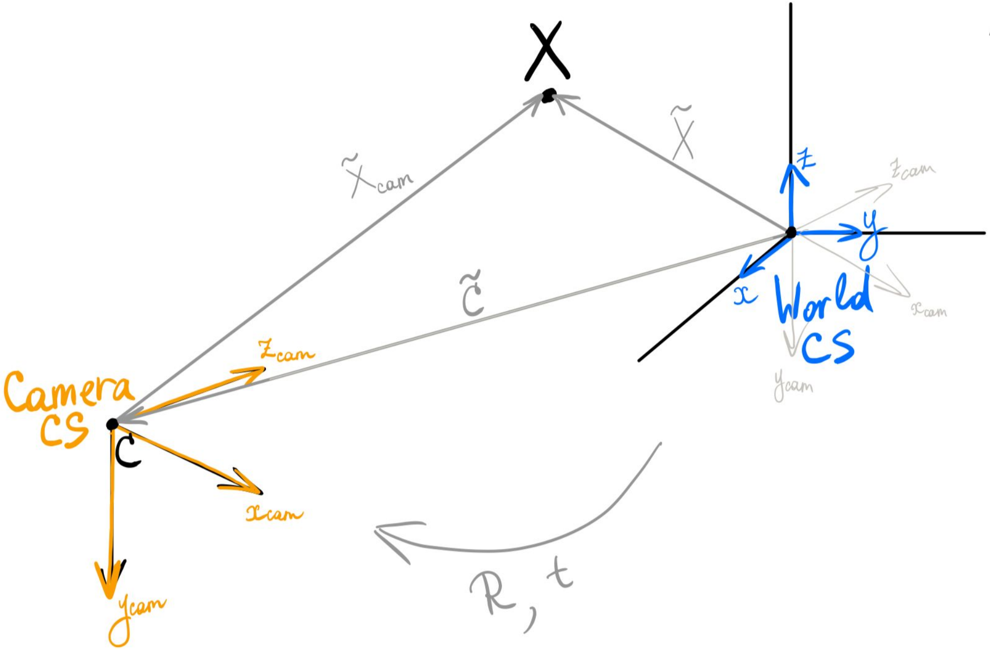
Summary on Homogeneous Coordinates

point	$\mathbf{p} = (X, Y, W)$
incidence	$\mathbf{p}^T\mathbf{u} = 0$
collinearity	$\lvert\, \mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \,\rvert = 0$
join of 2 points	$\mathbf{u} = \mathbf{p}_1 \times \mathbf{p}_2$

line	$\mathbf{u} = (a, b, c)$
incidence	$\mathbf{p}^T\mathbf{u} = 0$
concurrence	$\lvert\, \mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3 \,\rvert = 0$
intersection of 2 lines	$\mathbf{p} = \mathbf{u}_1 \times \mathbf{u}_2$

Points ⟵ Duality ⟶ **Lines**

Camera Rotation and Translation



$$R = \begin{bmatrix} - x_{cam} - \\ - y_{cam} - \\ - z_{cam} - \end{bmatrix}$$

"look right"
"look down"
"look at"

$$t = -R\tilde{C}$$

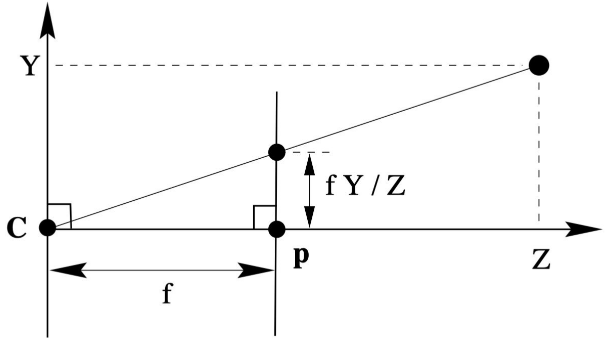
Inhomogeneous coordinates:

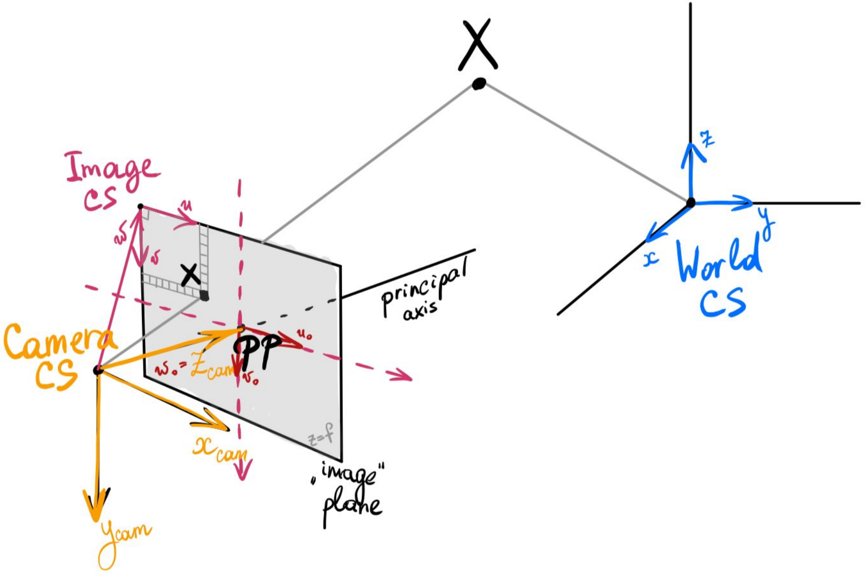
$$\tilde{\mathbf{X}}_{\text{cam}} = \mathrm{R}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$$

Homogeneous coordinates:

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}$$

Central Projection





Inhomogeneous coordinates:

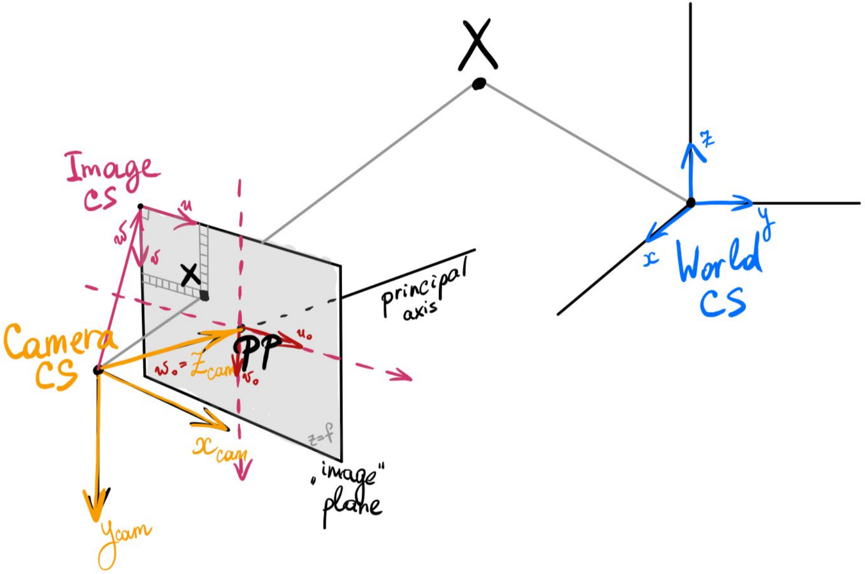
$$(X, Y, Z)^{\mathsf{T}} \mapsto (fX/Z, fY/Z)^{\mathsf{T}}$$

Homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \left[\begin{array}{ccc|c} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{array}\right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Camera Matrix **K**

Principal Point offset



Inhomogeneous coordinates:

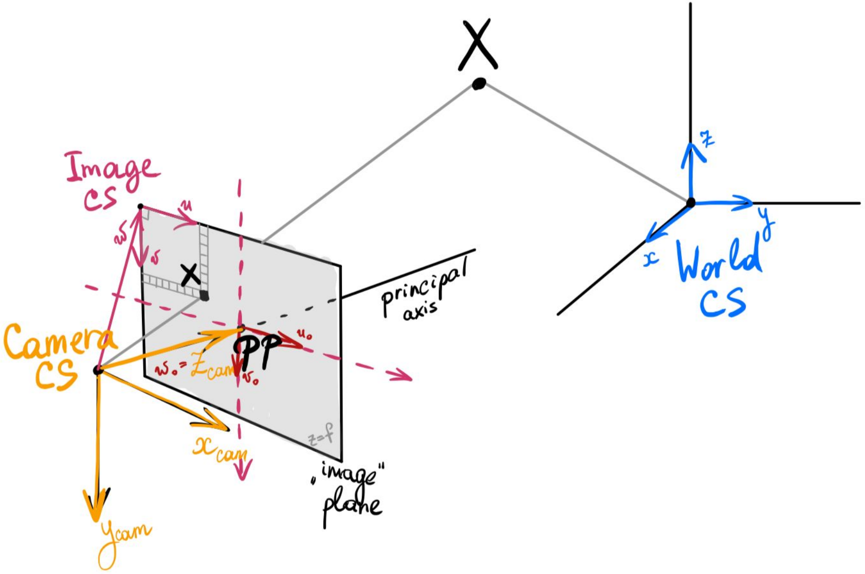
$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^{\mathsf{T}} \mapsto (f\mathrm{X}/\mathrm{Z} + p_x, f\mathrm{Y}/\mathrm{Z} + p_y)^{\mathsf{T}}$$

Homogeneous coordinates:

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} + \mathrm{Z}p_x \\ f\mathrm{Y} + \mathrm{Z}p_y \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$

Camera Matrix **K**

CCD (mm) to Image (px)



Homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \left[\boxed{\begin{matrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{matrix}} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Camera Matrix **K**

Homogeneous Representation of Imaging

A world point is *imaged* into an image point $\boxed{\mathbf{x} = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathtt{C}}]\mathbf{X}}$

1. *expressed w.r.t. camera basis*

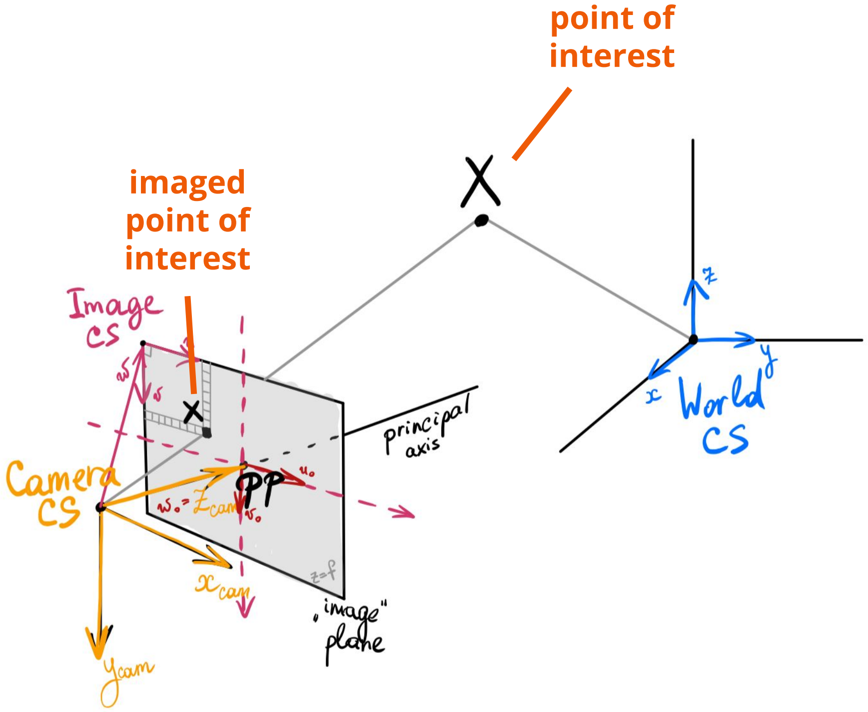
$$\mathbf{X}_{\mathrm{cam}} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\widetilde{\mathtt{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}$$

2. *projected along ray from 3D to 2D*

$$\mathbf{x} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

$$\mathtt{K} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

3. *distorted* $\boxed{\beta\tilde{\mathbf{x}} = f^d(\mathbf{x}, \lambda)}$



Restore Camera from Measurements

Camera Calibration (Offline Calibration)

Estimation of the parameters of the camera model from image(s) of a **known** structure — targets. Targets usually contain repeated patterns. The widely used targets are: checkerboards, deltille grids, circle grids. Grids can be augmented with AprilTags.

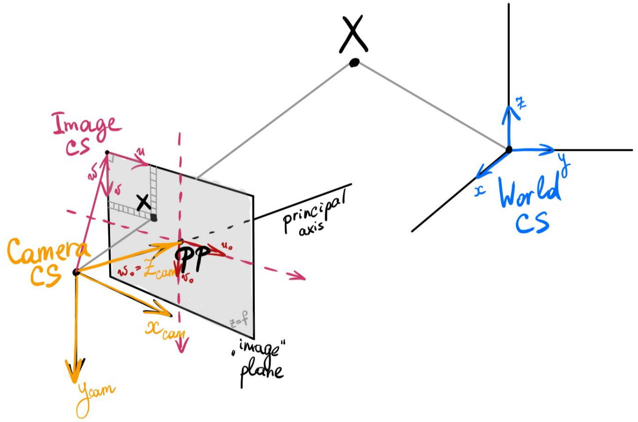
Tag36h11 TagStandard41h12 TagStandard52h13

Camera Auto-Calibration (Online Calibration)

Estimation of the parameters of the camera model from image(s) **in the wild**. Assumptions on the scene are usually made which **restricts** “the wilderness”. However, the class of an input images is **much broader** than in offline calibration.

Calibration from Point Correspondences

Given $\{ \mathbf{x}_i \leftrightarrow \mathbf{X}_i \}_{i=1}^{6}$ **compute** $\mathtt{P}$



For each pair: $\alpha \mathbf{x}_i = \mathtt{Q} \mathbf{X}_i$

Q is a 3x4 matrix, determined up to a non-zero scale:

$$\mathtt{Q} = \varepsilon \mathtt{P}$$

q$_i$ are 4x1 coordinate vectors

$$\mathtt{Q} = \begin{bmatrix} \mathrm{q}_1^\top \\ \mathrm{q}_2^\top \\ \mathrm{q}_3^\top \end{bmatrix} \qquad \begin{aligned} \alpha x &= \mathrm{q}_1^\top \mathbf{X} \\ \alpha y &= \mathrm{q}_2^\top \mathbf{X} \\ \alpha &= \mathrm{q}_3^\top \mathbf{X} \end{aligned} \longrightarrow \boxed{\begin{aligned} (\mathrm{q}_3^\top \mathbf{X})\, x &= \mathrm{q}_1^\top \mathbf{X} \\ (\mathrm{q}_3^\top \mathbf{X})\, y &= \mathrm{q}_2^\top \mathbf{X} \end{aligned}}$$

Calibration from Point Correspondences

$$\begin{aligned}(\mathrm{q}_3^\top \mathbf{X})\, x &= \mathrm{q}_1^\top \mathbf{X} \\ (\mathrm{q}_3^\top \mathbf{X})\, y &= \mathrm{q}_2^\top \mathbf{X}\end{aligned}$$

Introduce vector of parameters (which are elements of **Q**)

$$\mathrm{q} = \begin{bmatrix} \mathrm{q}_1^\top & \mathrm{q}_2^\top & \mathrm{q}_3^\top \end{bmatrix}^\top$$

… and express the two equations in matrix form:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{bmatrix} \mathrm{q} = 0$$

So each pair from $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}_{i=1}^{6}$ brings two rows into the matrix **M**.

Calibration from Point Correspondences

$$\begin{aligned}(\mathrm{q}_3^\top \mathbf{X})\,x &= \mathrm{q}_1^\top \mathbf{X}\\ (\mathrm{q}_3^\top \mathbf{X})\,y &= \mathrm{q}_2^\top \mathbf{X}\end{aligned}$$

Introduce vector of parameters (which are elements of **Q**)

$$\mathrm{q} = \begin{bmatrix}\mathrm{q}_1^\top & \mathrm{q}_2^\top & \mathrm{q}_3^\top\end{bmatrix}^\top$$

... and express the two equations in matrix form:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{bmatrix} \mathrm{q} = 0$$

So each pair from $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}_{i=1}^{6}$ brings two rows into the matrix **M**.

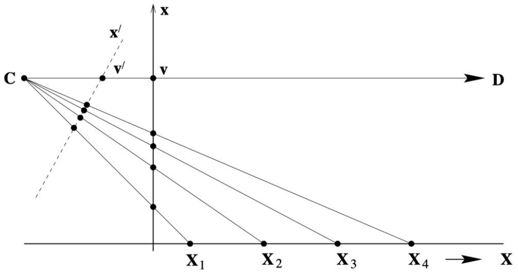
6 pairs in general position → **11** linearly independent rows → a one-dimensional space of solutions.

If **Q** is a solution, then **τQ** is also a solution and both determine the same projection for any non-zero **τ**

Vanishing Points

Point at Infinity or **Ideal Point**

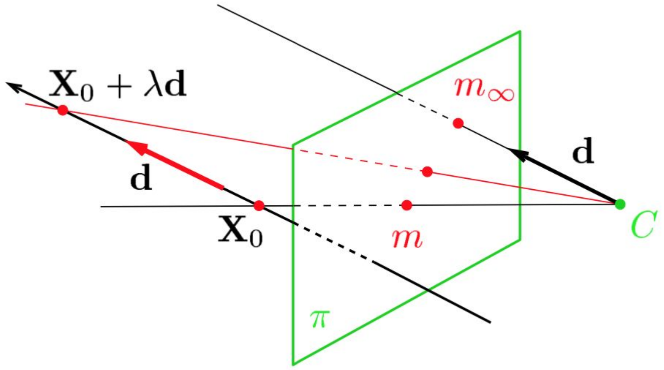
a point with the last coordinate **zero**



Vanishing Point (VP) of the line

the limit of the projection of a point that moves along a line in space infinitely in one direction or

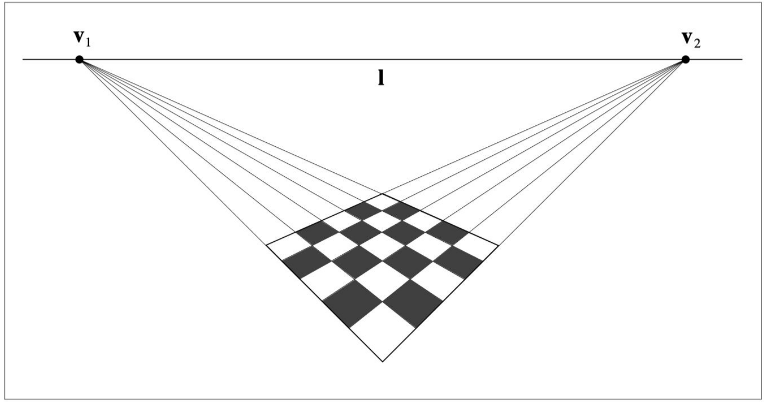
an image of the **point at infinity**



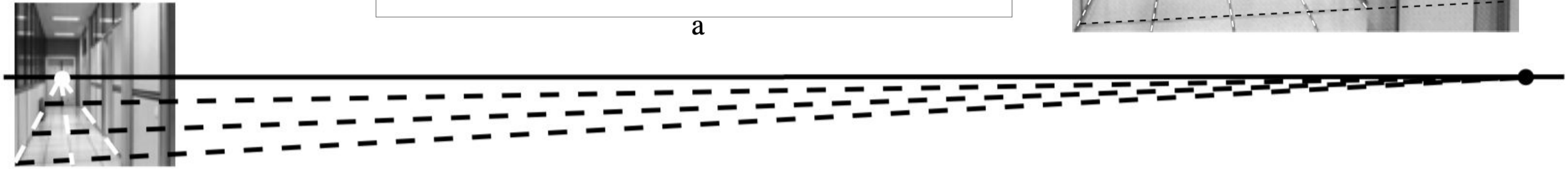
Vanishing Lines

Vanishing Line (VL) of the plane

The line through the vanishing points of lines on the scene plane.



a

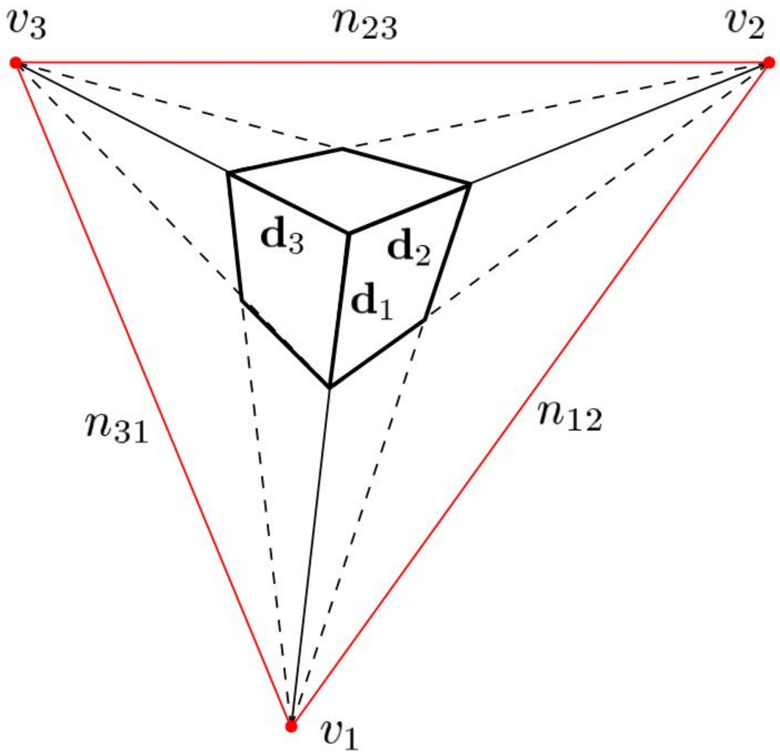


Auto-Calibration from Vanishing Points and Lines

Given 3 finite VPs, compute camera matrix **K**

$$\mathbf{d}_i \simeq \mathbf{Q}^{-1}\underline{\mathbf{v}}_i, \qquad\qquad i = 1, 2, 3$$

$$0 = \mathbf{d}_1^\top \mathbf{d}_2 = \underline{\mathbf{v}}_1^\top \mathbf{Q}^{-\top}\mathbf{Q}^{-1}\underline{\mathbf{v}}_2 = \underline{\mathbf{v}}_1^\top \underbrace{(\mathbf{K}\mathbf{K}^\top)^{-1}}_{\boldsymbol{\omega}\ \text{(IAC)}} \underline{\mathbf{v}}_2$$



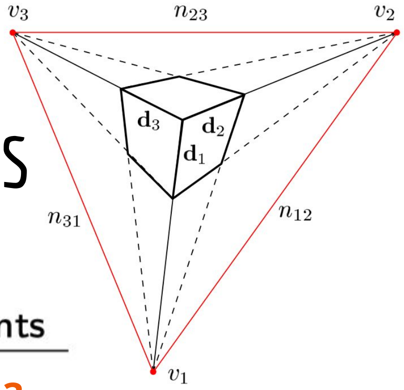
Auto-Calibration from Vanishing Points and Lines

configuration	equation	# constraints
orthogonal v.p.	$\underline{\mathbf{v}}_i^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_j = 0$	1
orthogonal v.l.	$\underline{\mathbf{n}}_{ij}^\top \boldsymbol{\omega}^{-1} \underline{\mathbf{n}}_{ik} = 0$	1
v.p. orthogonal to v.l.	$\underline{\mathbf{n}}_{ij} = \lambda\boldsymbol{\omega}\, \underline{\mathbf{v}}_k$	2
orthogonal raster $\theta = \pi/2$	$\omega_{12} = \omega_{21} = 0$	1
unit aspect $a = 1$ when $\theta = \pi/2$	$\omega_{11} - \omega_{22} = 0$	1
known principal point $u_0 = v_0 = 0$	$\omega_{13} = \omega_{31} = \omega_{23} = \omega_{32} = 0$	2

Estimate **ω** from at least 5 constraints
Get **K** from **ω**$^{-1}$ **= KK**$^{\mathbf{T}}$ by Choleski decomposition (e.g.)

Auto-Calibration from Vanishing Points and Lines

for example

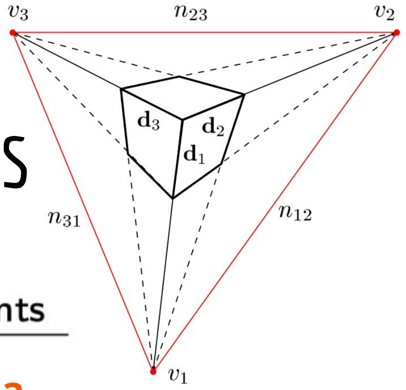


configuration	equation	# constraints
orthogonal v.p.	$\underline{\mathbf{v}}_i^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_j = 0$	1 ×3
orthogonal raster $\theta = \pi/2$	$\omega_{12} = \omega_{21} = 0$	1
unit aspect $a = 1$ when $\theta = \pi/2$	$\omega_{11} - \omega_{22} = 0$	1

$$\boxed{\boldsymbol{\omega} \simeq \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}}$$

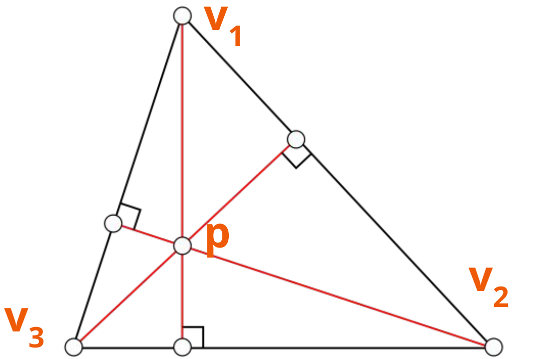
5 constraint linear equations for **5** parameters

Auto-Calibration from Vanishing Points and Lines



configuration	equation	# constraints
orthogonal v.p.	$\underline{\mathbf{v}}_i^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_j = 0$	1 ×3

$$\boxed{\boldsymbol{\omega} \simeq \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}}$$



The principal point **p** is computed as the triangle orthocenter

The focal length is computed as:

$$\underline{\mathbf{v}}_1^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_2 = 0 \quad \Rightarrow \quad f^2 = \left|(\mathbf{v}_1 - \mathbf{m}_0)^\top (\mathbf{v}_2 - \mathbf{m}_0)\right|$$

References

[1] Pajdla, Tomas. Elements of geometry for computer vision. FEE CTU, 2013

[2] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003

[3] Šára, Radim. TDV – 3D Computer Vision, Winter 2017

[4] Gkioulekas, Ioannis. Computational Photography, Fall 2019

[5] Kutulakos, Kyros. Computer Graphics, Fall 2010