Vision and Image Processing: Camera Geometry etc

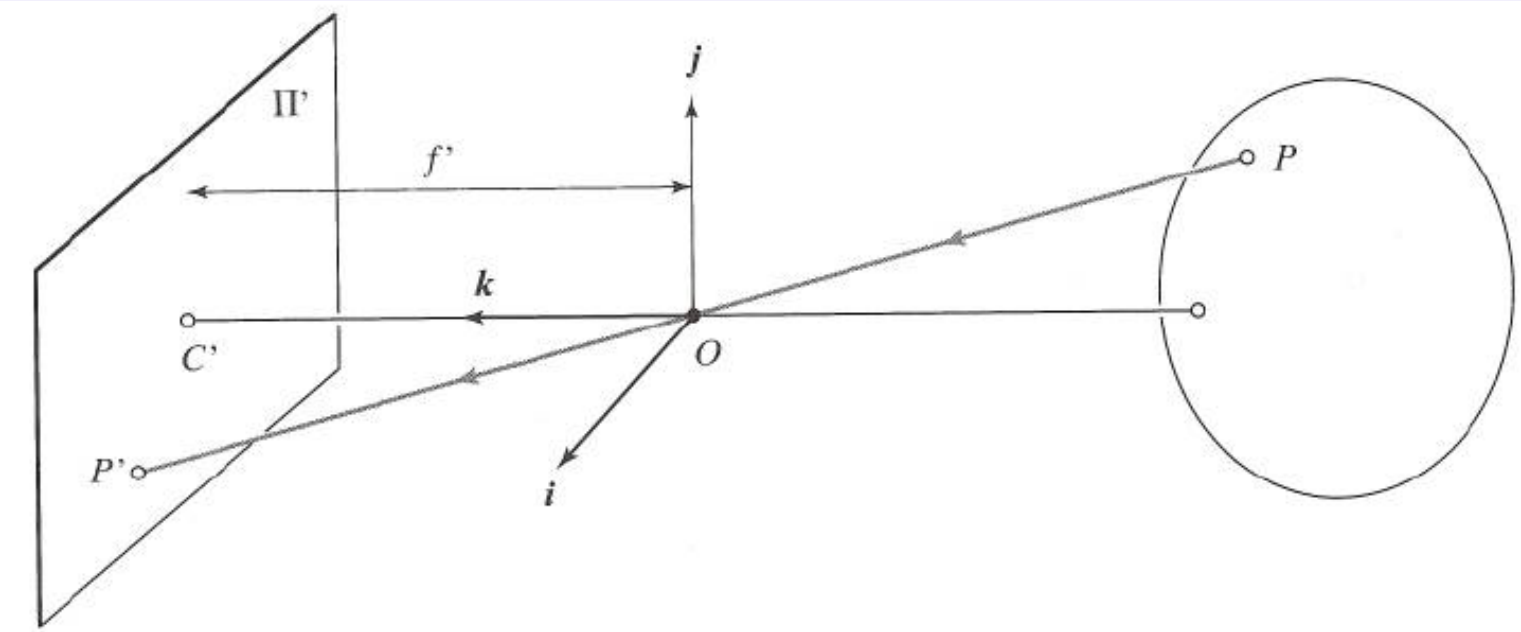
Søren Olsen

Department of Computer Science
University of Copenhagen

Plan for today

- Recap of Camera geometry and the perspective transformation
- Vanishing points and vanishing lines
- Camera matrices and parameters, Camera calibration
- 3D reconstructions
- Lens distortion
- Shape from focusing, Depth of field

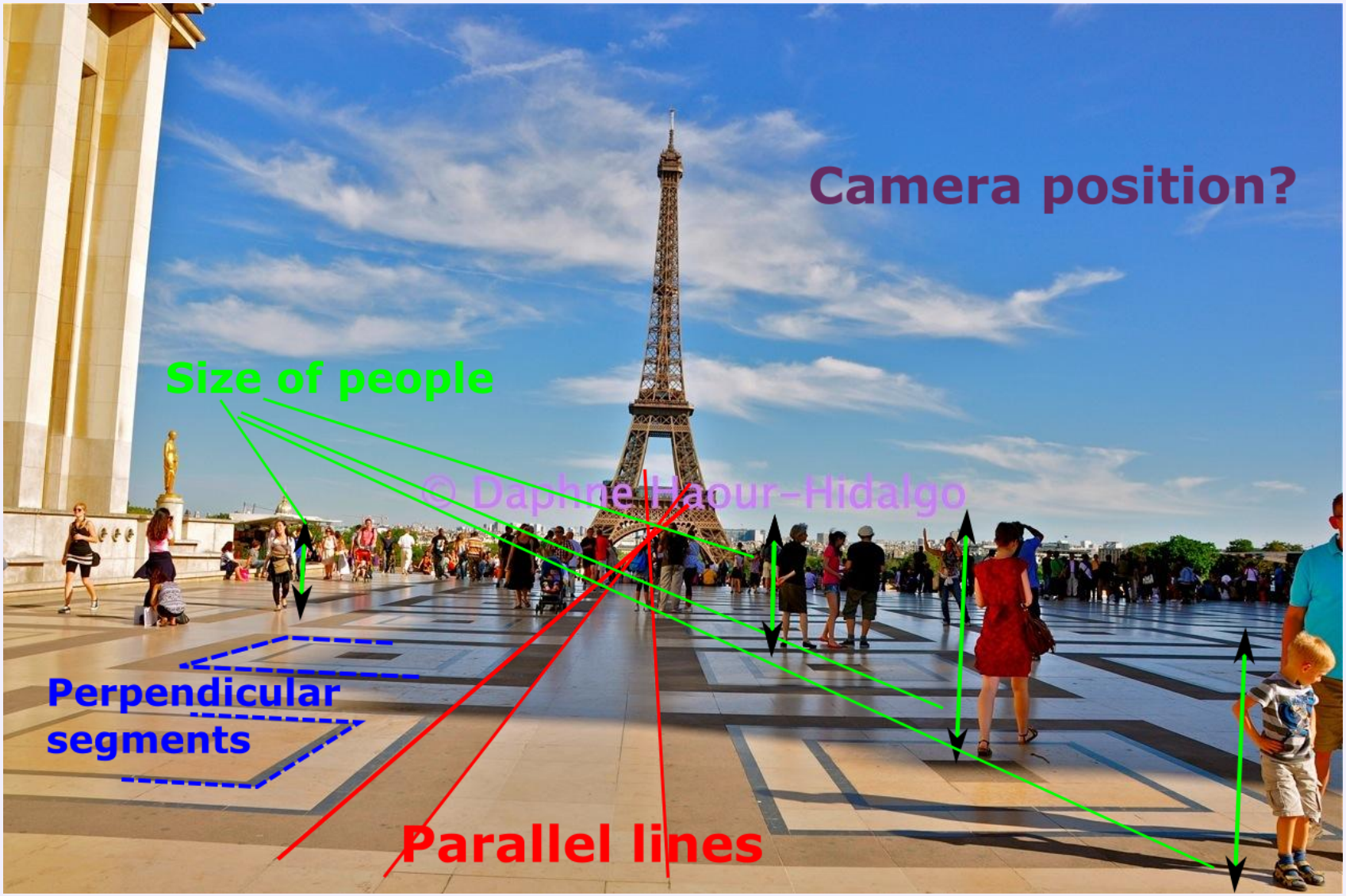
Projection Equations



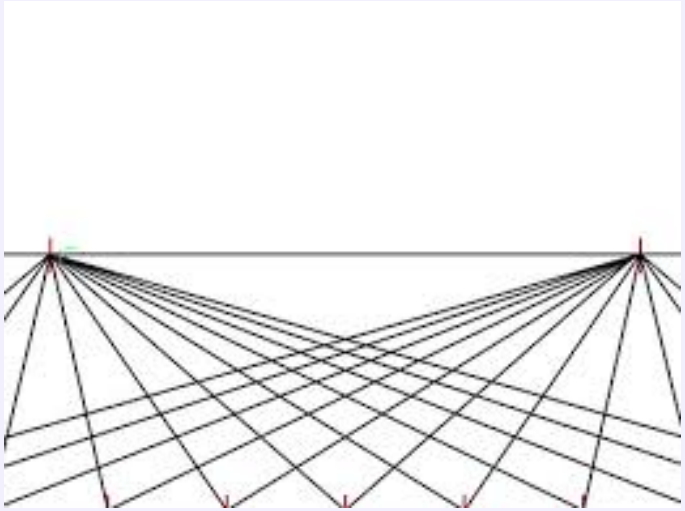
- $P(x, y, z)$, $P'(x', y', z')$. P' in the image plane $\Longrightarrow$ $z' = f$
- Remember: similar triangles:

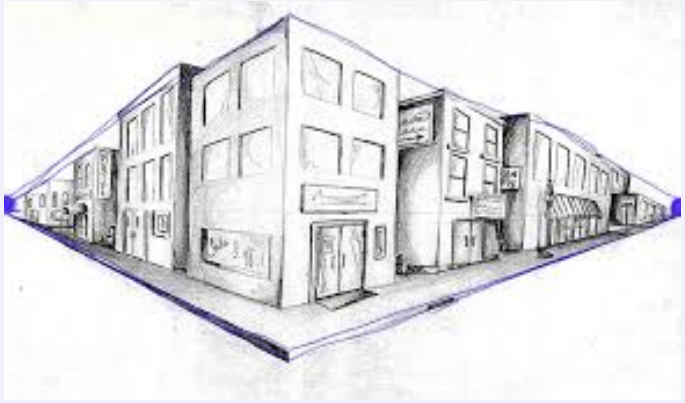
$$x' = f\frac{x}{z}, \quad y' = f\frac{y}{z}$$

Projections, Vanishing points



Vanishing line





Exercise

1. How many vanishing points can there be in an image ?

Exercise

1. How many vanishing points can there be in an image ?

   As many as there are planar sufaces with parallel linear texture

2. Can we compute the vanishing line of a planar surface from images of 1, 2 or, 3 different sets of parallel lines on the surface?

Exercise

1. How many vanishing points can there be in an image ?

   As many as there are planar sufaces with parallel linear texture

2. Can we compute the vanishing line of a planar surface from images of 1, 2 or, 3 different sets of parallel lines on the surface?

   Yes, we need 2 sets of parallel lines. This gives 2 VPs and the VL connecting them.

3. May we compute 3D surface orientation (the surface normal) from a vanishing line?

Exercise

1. How many vanishing points can there be in an image ?

   As many as there are planar sufaces with parallel linear texture

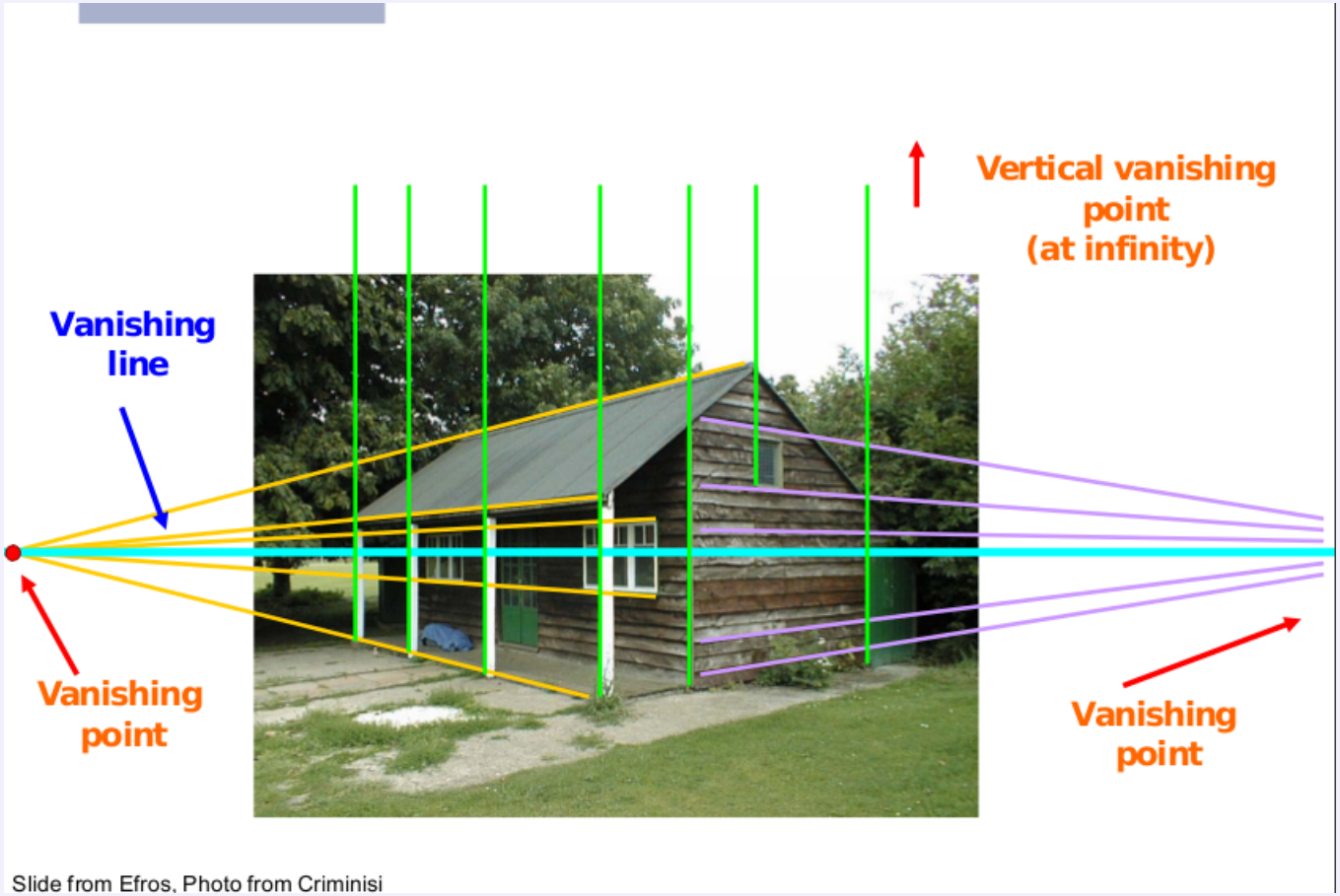
2. Can we compute the vanishing line of a planar surface from images of 1, 2 or, 3 different sets of parallel lines on the surface?

   Yes, we need 2 sets of parallel lines. This gives 2 VPs and the VL connecting them.

3. May we compute 3D surface orientation (the surface normal) from a vanishing line?

   Yes, if we know the focal length f of the camera. If the VL-equation is $Ax + By + C = 0$ then the 3D surface normal will be $(-\frac{fA}{C}, -\frac{fB}{C}, -1)$.

Vanishing line



Slide from Efros, Photo from Criminisi

VPs from texture

If we can measure a texture density, then we may estimate the directions of minimal and maximal density change. These points at the VPs thus giving the VL and the 3D surface normal.

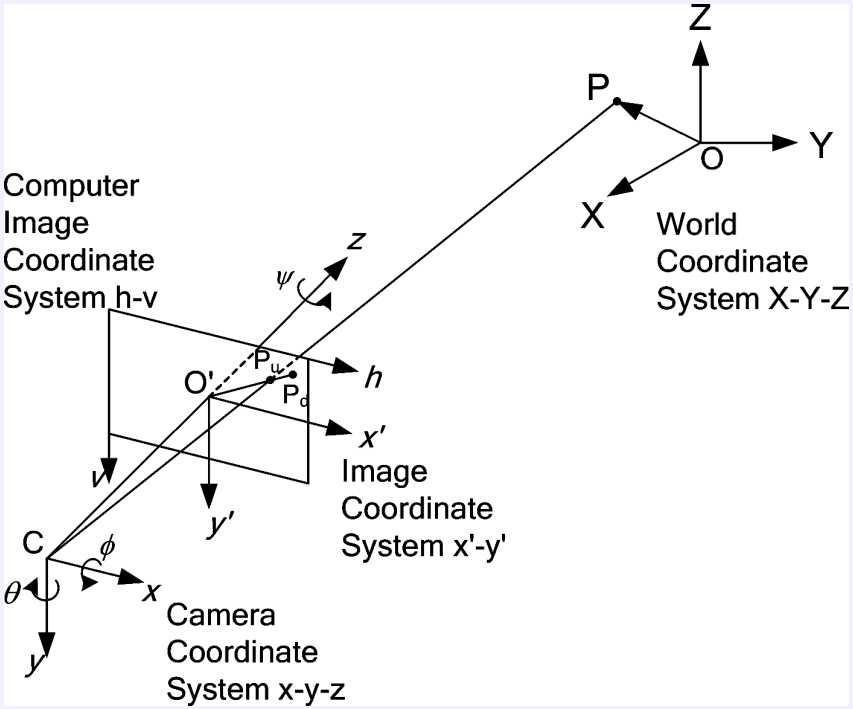
World, Camera and Image Coordinates

In the previous slides we were not precise on the many coordinate systems are implicitly used:

- 3D World Coordinates: Coordinate system of the 3D world.
- Camera Coordinates: 3D coordinate system attached to the camera.
- Image Coordinates: 2D Coordinate system attached to the image plane.
- The coordinate system for the sampled and digitised image

Not that simple in practice!

From World through Camera to Image coordinates



- The camera and image coordinate axes are coinciding.
- Pixels values are sampled at the image plane according to an offset (the principal point) and horizontal and vertical sampling rates. Pixels may not be square.

Intrinsic vs Extrinsic Camera Parameters

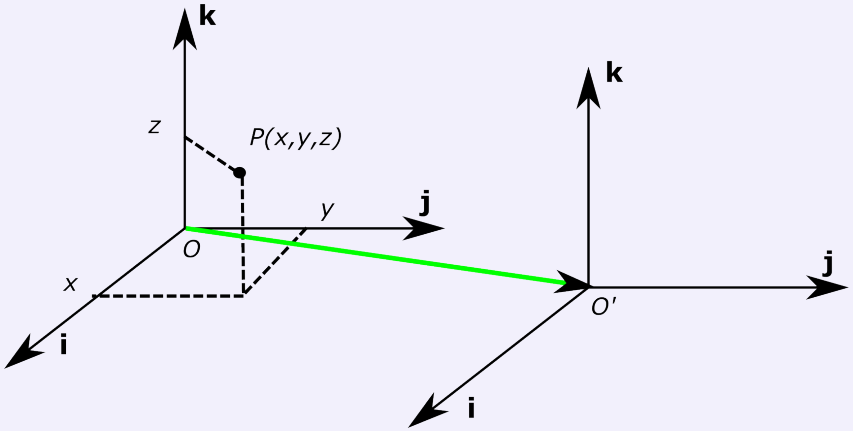
Intrinsic parameters refer to internal parameters:

- The principal point: 2 parameters
- Scale factors (or sampling frequencies) for the pixels sizes in both x and y directions: 2 parameters, multiplied by the focal length, resulting in the effective focal length $(f_x, f_y) = f \cdot (s_x, s_y)$.
- Skewness of pixels: 1 parameter.

Extrinsic Camera parameters:

- Position of the camera coordinate system with respect to the world coordinates system: translation: 3 parameters,
- Orientation of the camera coordinate system with respect to the world coordinates system: rotation: 3 parameters.

Translating Coordinate System



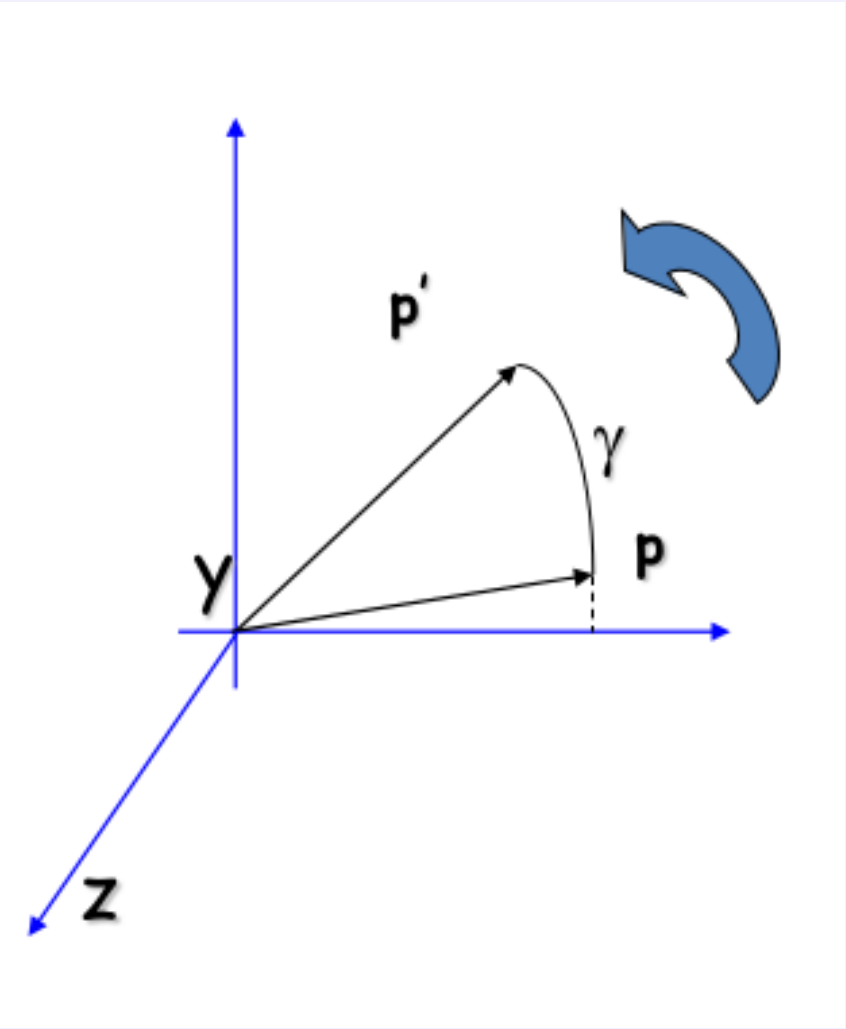
- Subtract the translation vector:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} x_{O'} \\ y_{O'} \\ z_{O'} \end{pmatrix} = \begin{pmatrix} x - x_{O'} \\ y - y_{O'} \\ z - z_{O'} \end{pmatrix}$$

- Transformation in homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & -x_{O'} \\ 0 & 1 & 0 & -y_{O'} \\ 0 & 0 & 1 & -z_{O'} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotating Coordinate System



Rotations along coordinate axes

$$R_{x\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$$

$$R_{y\beta} = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}$$

$$R_{y\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Can also be written in homogeneous coordinates

3D rotations

To obtain a 3D rotation apply all 3 single-axis rotations:

$$
\begin{aligned}
R &= R_{x\alpha} R_{y\beta} R_{y\gamma} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

Please notice that the order of the 3 rotation matrices do matter. However, no matter which order (α, β, γ) may be found such that the resulting 3D rotation is correct.

Camera Matrix

- Combine world vs camera coordinates with
- Simple Camera calibration matrix, now extended with Image plane transformation (axis scalings, shear, translation)

$$\mathbf{C} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}]$$

- $\mathbf{K}_{3\times 3}$ matrix encoding the homogeneous transformations inside the camera. $\mathbf{K}$ specifies the Intrinsic parameters.
- [**R t**] Concatenation of world coordinates rotation and origin translation to align camera and world coordinates.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \underbrace{\begin{pmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}}_{[\mathbf{R}\ \mathbf{t}]} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Exercise

- How many independent parameters are there to estimate in a camera calibration ?

Exercise

- How many independent parameters are there to estimate in a camera calibration ?

  11: 5 intrinsic and 6 extrinsic (sometimes simplified to 3+6)

- May all calibration parameters be found using Linear algebra ?

Exercise

- How many independent parameters are there to estimate in a camera calibration ?

  11: 5 intrinsic and 6 extrinsic (sometimes simplified to 3+6)

- May all calibration parameters be found using Linear algebra ?

  No, they don't appear in a linear combinations. Some are multiplied together.

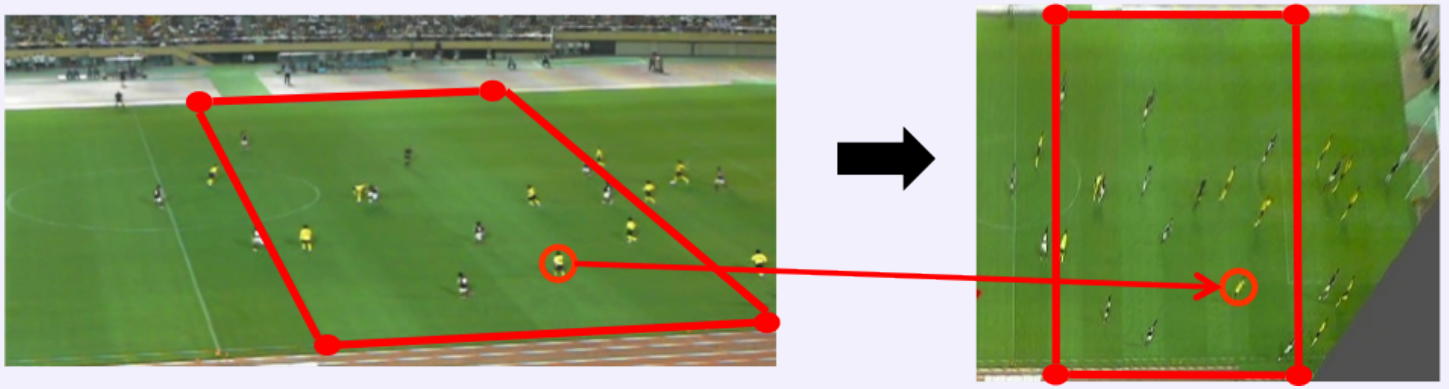
The geometric transformation between two images of a planar scene

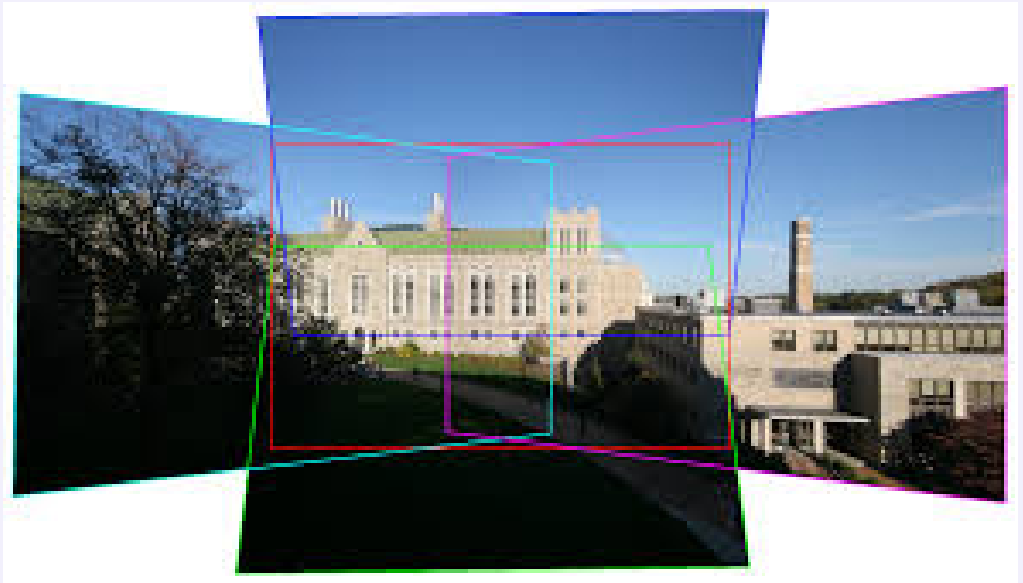
- The perspective projection of a planar scene surface is a transformation $(X, Y, Z) \longrightarrow (x, y)$ called an homography:

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = H \cdot \mathbf{X}$$

- The transformation between two images is an homography.
- Homographies conserves straight lines, but not parallelism.
- Two parallel lines intersect at infinity: after an homography they may intersect at finite distance.

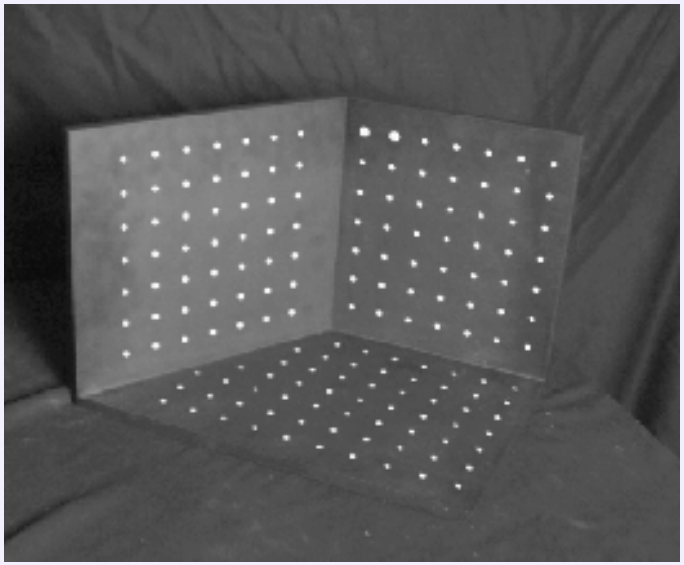
An homography can accurately describe the stitching of images of a planar scene. Non-planar scenes cannot be stitched correctly using homographies.





Geometric Calibration

- Computing the camera matrix is called geometric calibration.
- Extrinsic parameters: usually easy, but requires metric knowledge on scene features.
- Calibration focuses more on intrinsic parameters (calibration matrix **K**): Some parameters are easy (f) and some ((u_0, v_0)) are difficult to estimate correctly.



- Use an object with known geometry
- Use vanishing points / lines
- Use other cues...

Repetition: The camera matrix

Please recall the definition of the camera matrix:

$$M = K\,[R\ \mathbf{t}]$$

and the projection written out:

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}}_{[\mathbf{R}\ \mathbf{t}]} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

or

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{pmatrix} \mathbf{m}^1 \\ \mathbf{m}^2 \\ \mathbf{m}^3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{m}^1 U \\ \mathbf{m}^2 U \\ \mathbf{m}^3 U \end{pmatrix}$$

where $\mathbf{m}^i$ is the i'th row of the camera matrix.

Camera calibration

In Camera calibration the task is to recover the 12 unknowns $\mathbf{m}_{ij}$ in the camera calibration matrix. Converting the result before to image coordinates we have:

$$\begin{aligned} x\mathbf{m}^3 U &= \mathbf{m}^1 U \\ y\mathbf{m}^3 U &= \mathbf{m}^2 U \end{aligned}$$

Let $\mathbf{m} = (m_{11}, \ldots, m_{14}, \ldots, m_{34})^\top$ be the vector of the 12 unknowns. Isolating these in the equations above lead to: $A\mathbf{m} = 0$, where:

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 & -x_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 & -y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix}$$

Camera calibration continued

- Given 6 points (x, y) projected from 6 known 3D points (X, Y, Z) we can solve for the camera matrix elements m_{ij}.
- In practise many more than 6 points are preferred to cancel noise.
- From the 12 elements m_{ij} it is possible to recover all 11 parameters f_x, f_y, α, β, γ, etc.
- There exist more advanced calibration methods than the shown linear calibration.

Linear triangulation

Let $U = (X, Y, Z, 1)^\top$. For the first coordinate in the left camera we have:

$$x^L = \frac{\mathbf{m}_L^1 U}{\mathbf{m}_L^3 U}$$

and similar for the y-coordinate and the right camera. Multiplying by the denominator we get:

$$\begin{aligned} \mathbf{m}_L^3 x_L U &= \mathbf{m}_L^1 U \\ \mathbf{m}_L^3 y_L U &= \mathbf{m}_L^2 U \\ \mathbf{m}_R^3 x_R U &= \mathbf{m}_R^1 U \\ \mathbf{m}_R^3 y_R U &= \mathbf{m}_R^2 U \end{aligned}$$

Subtracting the right side and putting U outside a parentesis we get the homogeneous equation $AU = 0$, where:

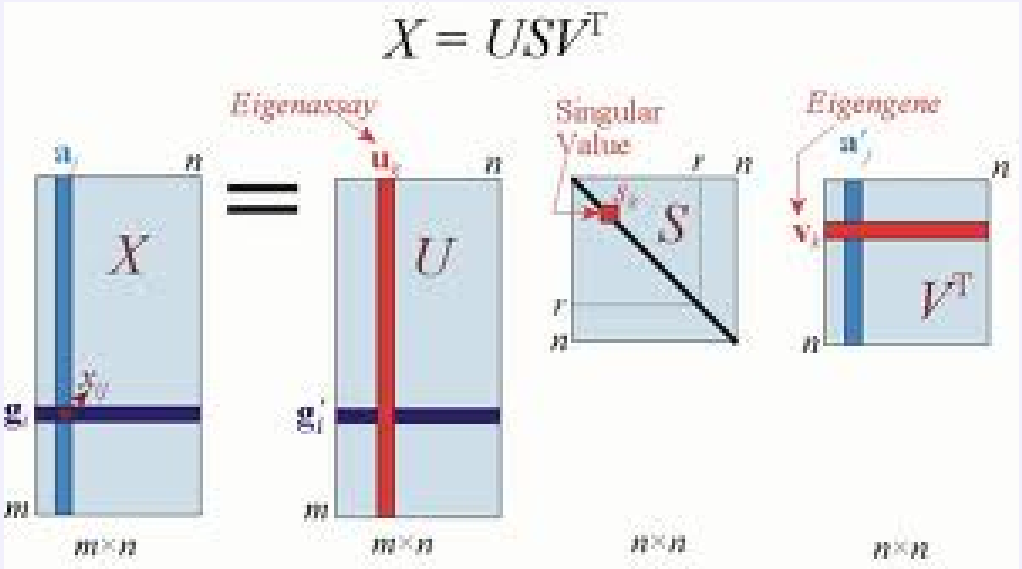
$$A = \begin{pmatrix} \mathbf{m}_L^3 x_L - \mathbf{m}_L^1 \\ \mathbf{m}_L^3 y_L - \mathbf{m}_L^2 \\ \mathbf{m}_R^3 x_R - \mathbf{m}_R^1 \\ \mathbf{m}_R^3 y_R - \mathbf{m}_R^2 \end{pmatrix}$$

Solving $AU = 0$

Many different numerical approaches. Singular Value Decomposition (SVD) is based on decomposition of the matrix A into a product of 3 matrices:

$$A = UDV^\top$$

where A and U is $m \times n$ and $U^\top U = I$,
D is a diagonal $n \times n$ matrix with non-negative singular values,
and V is a $n \times n$ unitary matrix $VV^\top = V^\top V = I$.



$$A\,\mathbf{U} \;=\; UDV^{\top}\,\mathbf{U} \;=\; 0$$

We are not interested in the trivial solution $\mathbf{U} = \mathbf{0}$ but rather the solution lying in the (right) Null-space of the matrix A.

Due to noise all singular values we may have $\sigma_i > 0$. If we know that A should be singular may find the closest such matrix by zeroing out the smallest singular value.

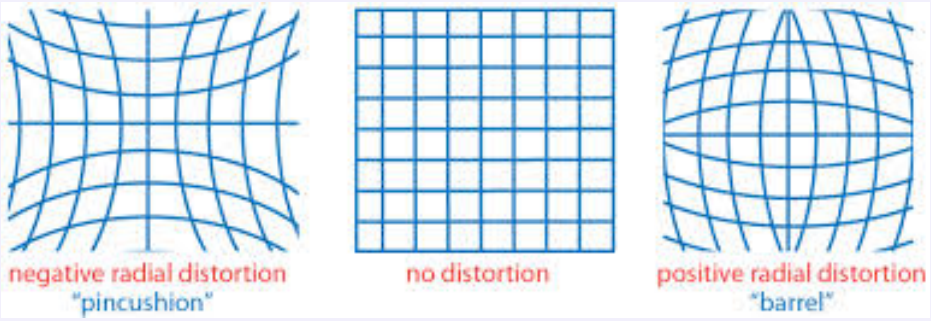
The singular values in the diagonal matrix D specifies the energy contained in each of the dimensions. Usually, these are sorted in decreasing order.

we get the solution to $A\mathbf{U} = \mathbf{0}$ as the last column of V. In MATLAB:

```
[U,D,V] = svd(A); solution = V(:,end);
```

Lens distortion

Cheap or wide-angle lenses causes geometric lens distortion. The most common distortion is (barral) radial distortion:

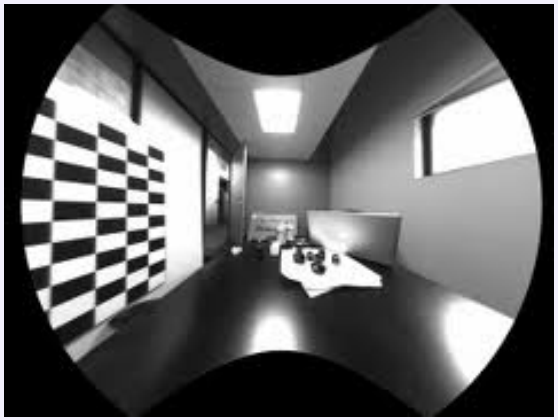


To comply with the assumptions of the pin-hole camera it may be necessary to correct for lens distortion before doing anything else.

Lens distortion

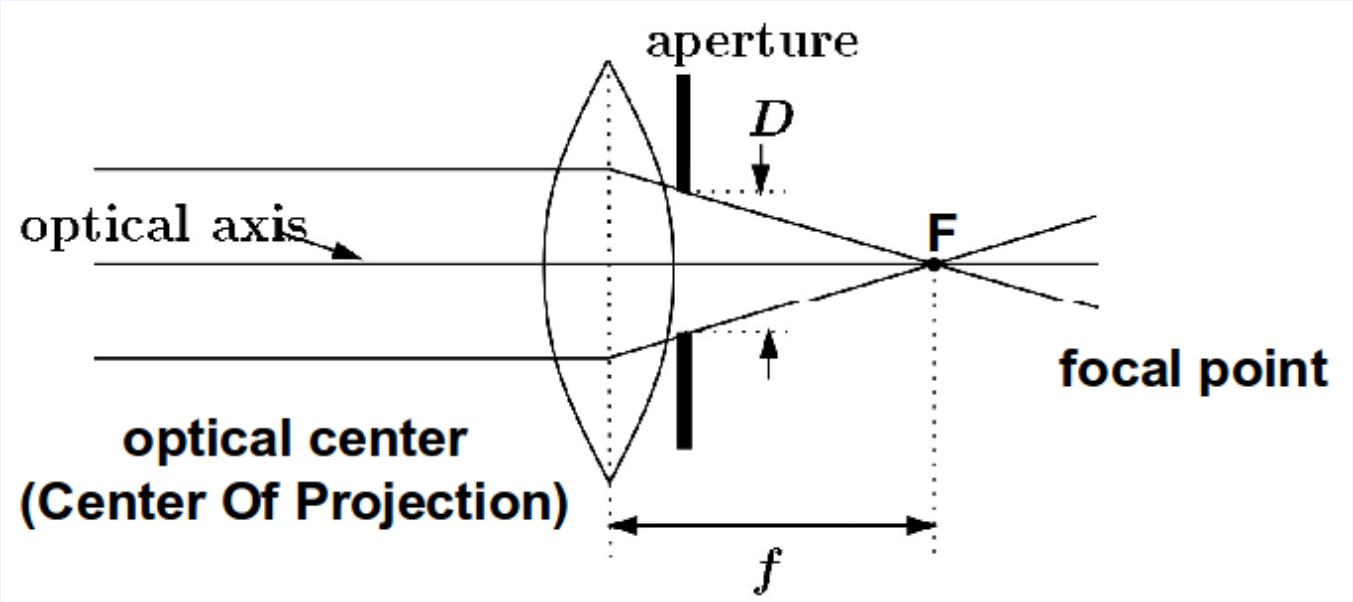
Lens distortion is non-linear. The displacement of pixels may (for radial distortion) be modeled as multiplicative even ordered polynomium in radial distance from the principal point.

$$\begin{aligned} \Delta x &= x\left(\kappa_1 r^2 + \kappa_2 r^4 + \cdots\right) \\ \Delta y &= y\left(\kappa_1 r^2 + \kappa_2 r^4 + \cdots\right) \\ r^2 &= x^2 + y^2 \end{aligned}$$



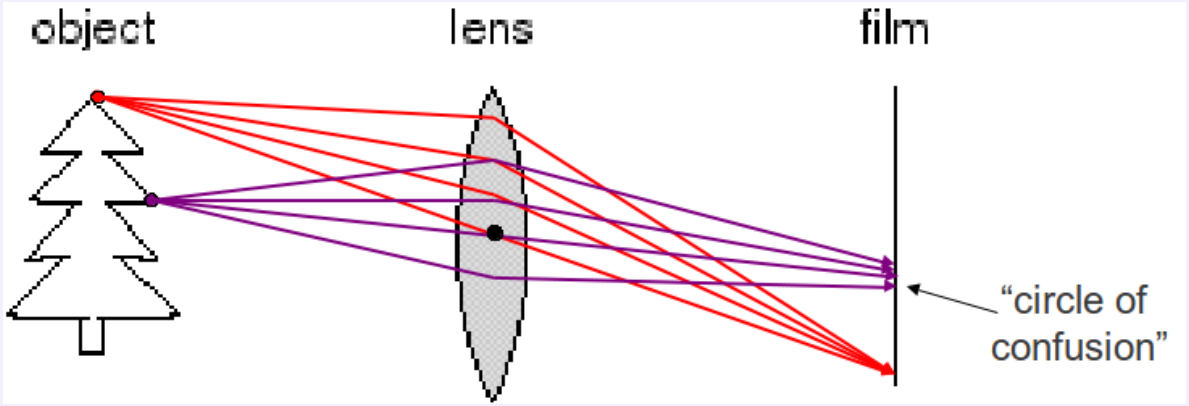
Often only one or two terms are used. The two konstants κ_1 and κ_2 are estimated using an iterative (nonlinear) optimisation approach.

Focal Length, Aperture



- Lens focuses parallel rays into a single point.
- Aperture restricts range of rays.
- Zoom by increasing *f*, make wide angle by decreasing *f*
- In practice zoom optics use a set of lenses moving in a non-linear way wrt. each other.

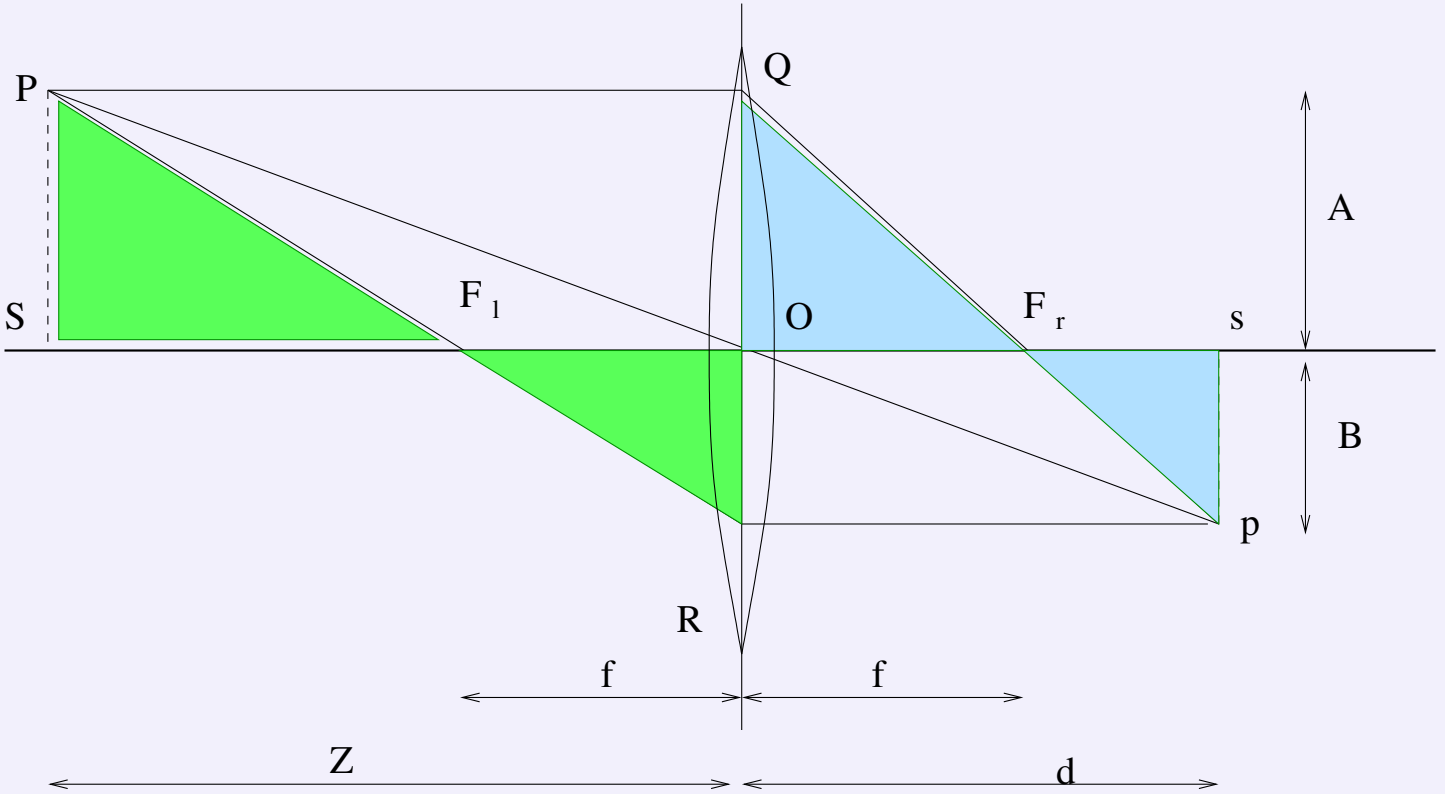
Focus, Depth of Field



- Specific distance for which objects are in focus
- Changing shape of lens changes the focus distance.
- Changing distance between lens and sensor changes the 3D points in focus.

Depth from Focusing

The figure below show the basic functioning of a thin lens.



Considering the similar triangles **SPF**$_l$ and **ORF**$_l$, and the triangles **OQF**$_r$ and **spF**$_r$ we get:

$$\frac{Z-f}{f} = \frac{A}{B} = \frac{f}{d-f} \qquad \text{or} \qquad Zd - Zf - df = 0$$

The thin lens equation

$$\frac{1}{Z} + \frac{1}{d} = \frac{1}{f}$$

is the simplest model, “better” than the pin-hole model, of the image projection.

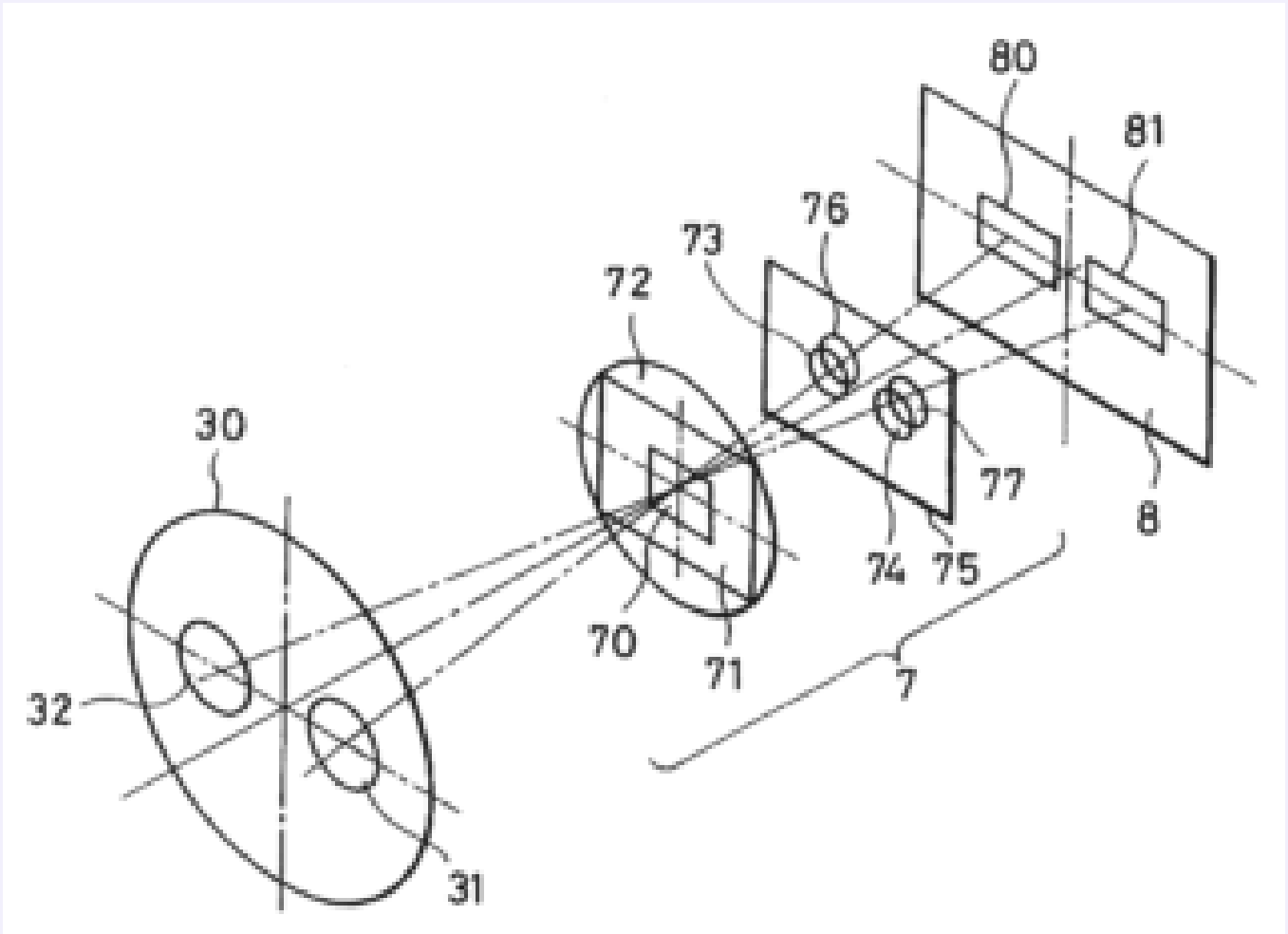
$$Z = \frac{fd}{d - f}$$

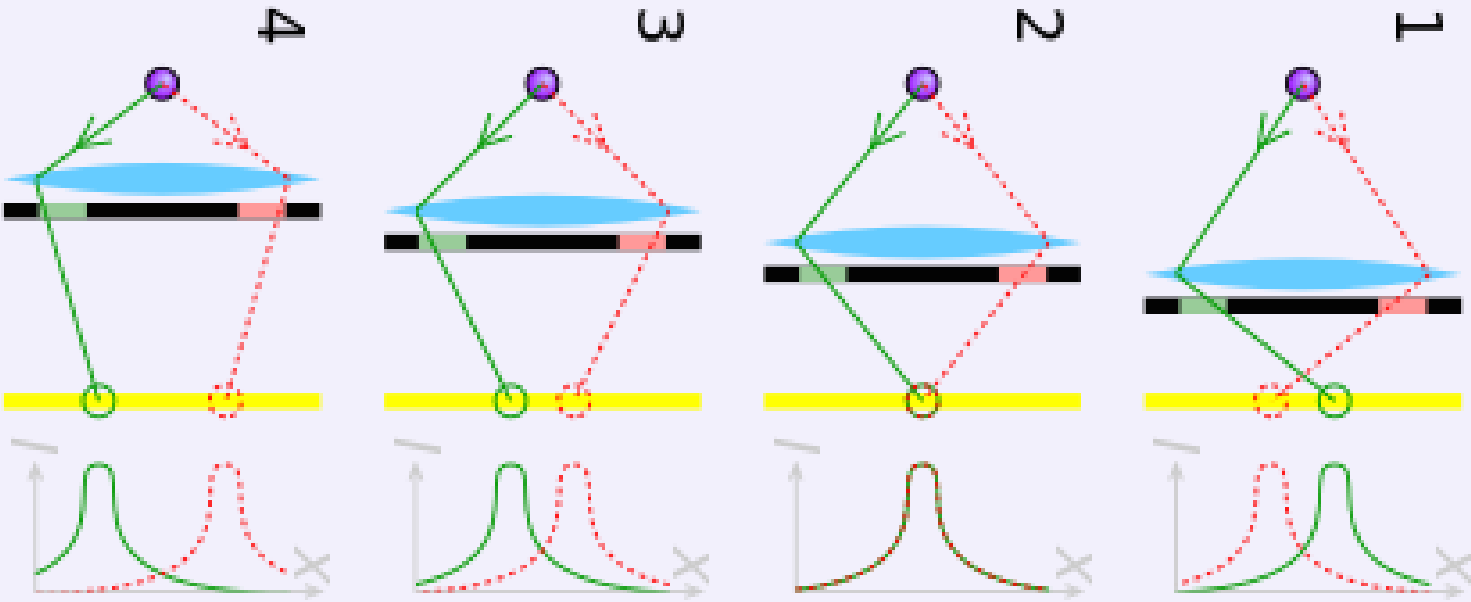
f is known from the producer of the lens. d depends on the focus adjustment and is in general not available.

To recover Z we must auto-focus. For the right value of d the image is in focus. In all other settings the images is more or less blurred.

Auto focus cameras

Adjust the focus ring position to make the projection of light through two special lens systems hit similar positions on two 1D sensor arrays.

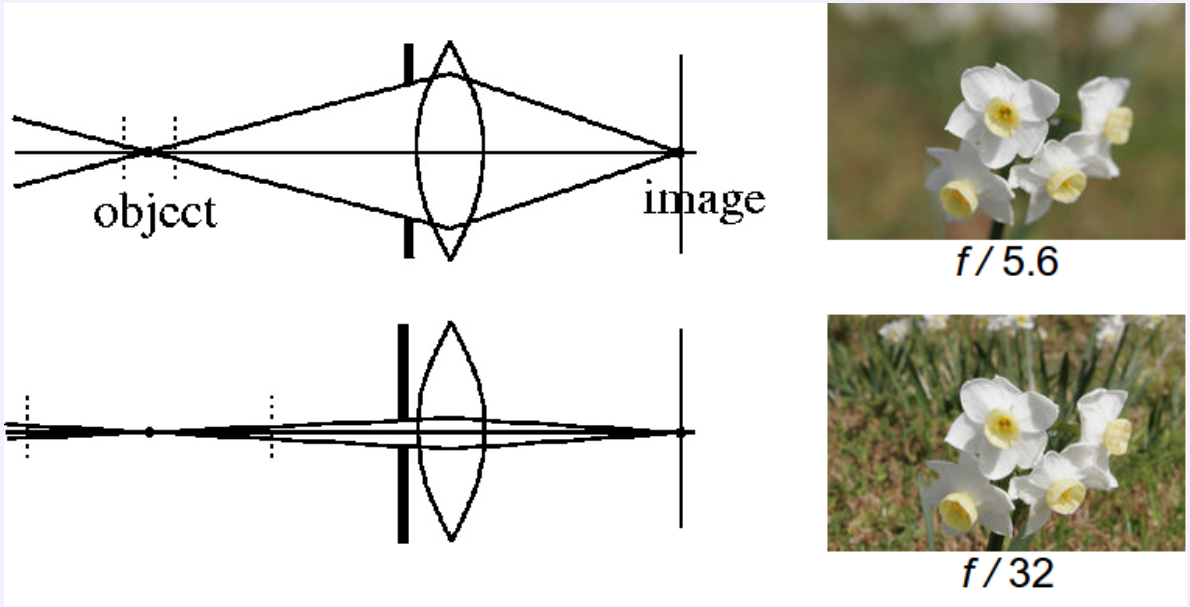




Normally, auto-focus cameras don’t allow to read the focus ring position *d*. We may use similar approach for computer controlled lenses.

Depth from Field

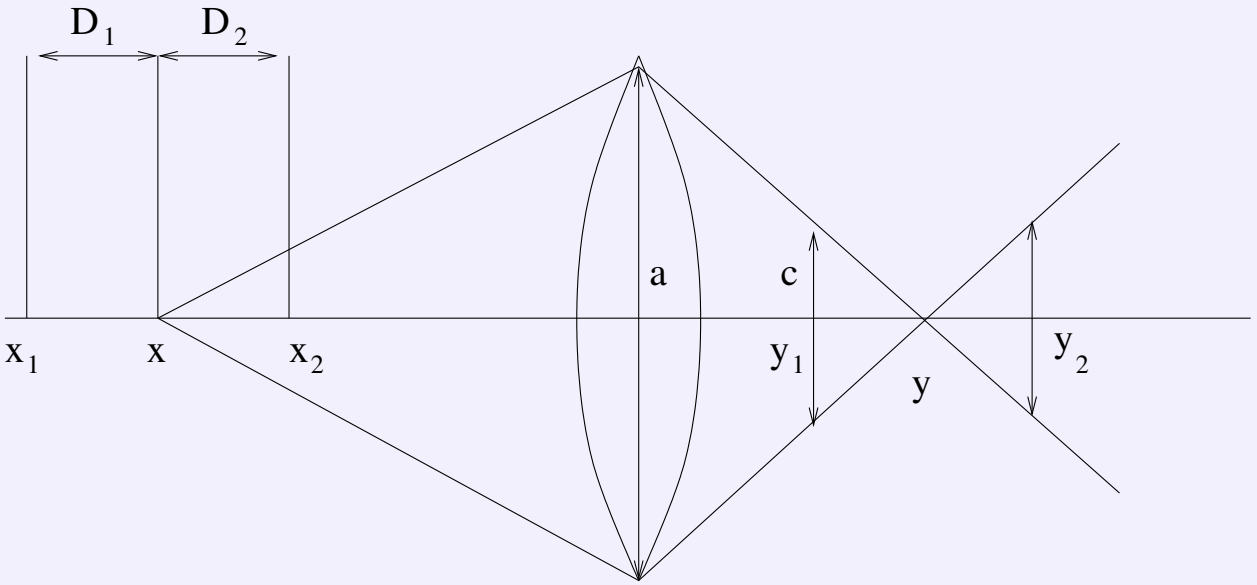
Informally, Depth of field (DoF) is the 3D range in which the scene is in focus.



When we tage photos we often want a large DoF to have the full scene in focus. For depth measurement we want a very short DoF.

DoF 1

Formally, DoF is is the distance between the nearest and the farthest object planes at which a 3D point will project into the same single pixel.



A light at distance x will be projected on the image plane as a blurred disc, called The circle of confusion, with diameter c. Noting the similar triangles we get:

$$\frac{c}{a} = \frac{y - y_1}{y} = \frac{y_2 - y}{y}$$

Simplification

The depth of field depend on the distance x, the lens aperture a, the focal length f, and the pixel diameter c.

$$D = \frac{2acfx(x-f)}{a^2f^2 - c^2x^2} \approx \frac{2cx^2}{af}$$

- The depth of field increases with squared distance and decreases with aperture and focal length.
- At 1m distance the accuracy may be computet to be around 1%. This is comparable or better to what usually can be obtained by methods such as stereo.

For good images we wish D large. For usage in depth from focusing we wish D small.

The Zoo of “Depth from X” - methods

- One image:
  - Shape from shading
  - Shape from texture
  - Depth from zooming
- Two images:
  - Stereo analysis
- Several images:
  - Photometric Stereo
  - Depth from Focusing
  - Depth from Defocusing
- Many images:
  - Optical flow based methods
  - Structure from Motion (a large class of methods)
- Active sensors: range sensors (Time Of Flight, Structured light, Laser).

But we have two eyes

Stereo vision is among the most important human sensing methods.
Next lecture we will talk a lot about stereo vision.