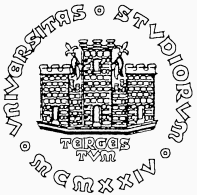
Computer Vision and Pattern Recognition

Course ID: 554SM – Fall 2018

Felice Andrea Pellegrino

University of Trieste
Department of Engineering and Architecture

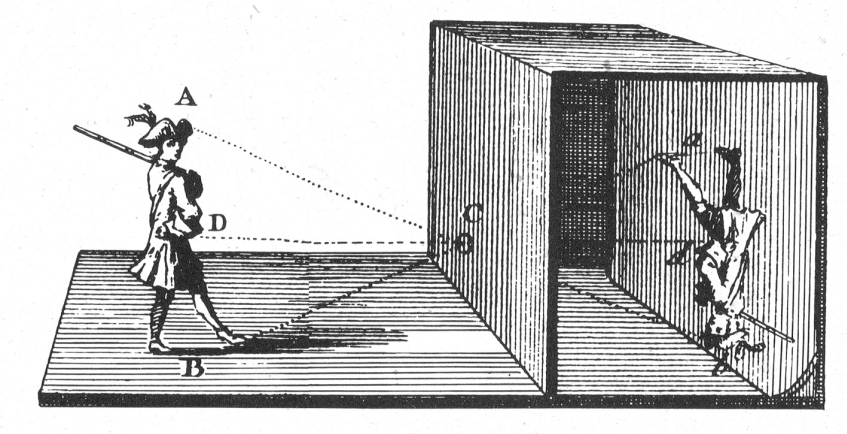


554SM –Fall 2018

Lecture 2: Image Formation

The pinhole camera

Camera obscura



The principle of Camera Obscura

Camera obscura (cont.)

The image of the New Royal Palace at Prague Castle created at the attic wall by a hole in the tile roofing. Credits: `https://commons.wikimedia.org/wiki/User:Gampe`

Milestones

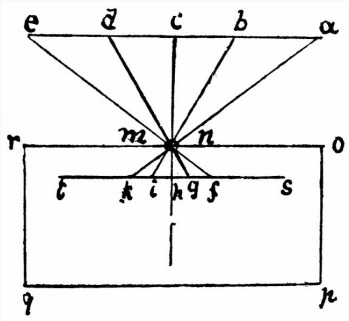


Diagram of camera obscura, Leonardo da Vinci, 1515

- Leonardo da Vinci (1452-1519): first record of camera obscura

Milestones

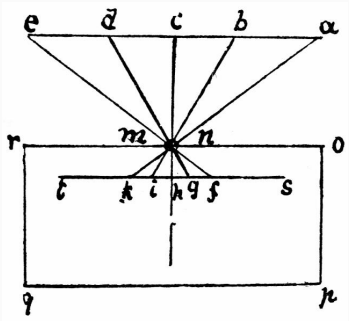
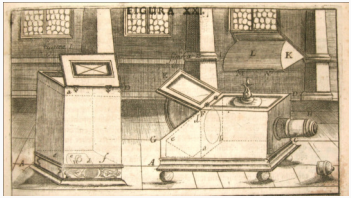


Diagram of camera obscura, Leonardo da Vinci, 1515



Drawing of a portable camera by Johann Zahn, 1685

- Leonardo da Vinci (1452-1519): first record of camera obscura
- Johann Zahn (1641-1707): first portable camera

Milestones

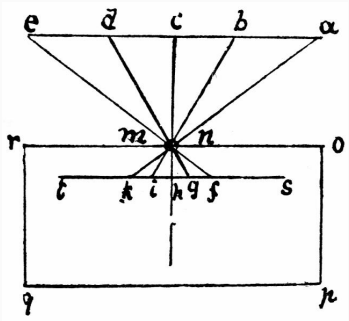
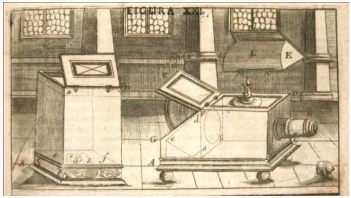
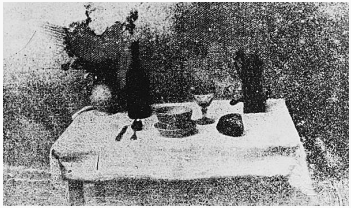


Diagram of camera obscura, Leonardo da Vinci, 1515



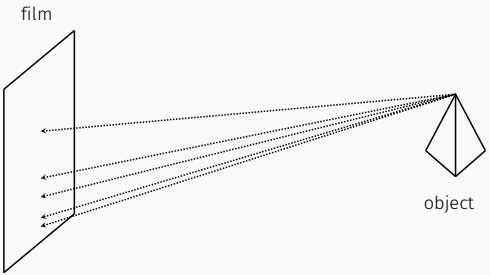
Drawing of a portable camera by Johann Zahn, 1685



Joseph Nicéphore Niépce, La table servie, 1822

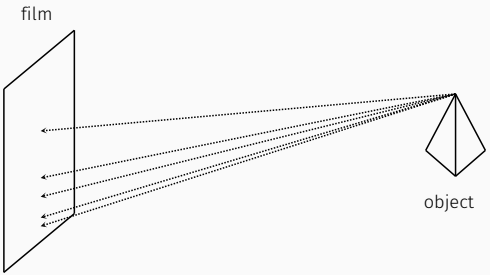
- Leonardo da Vinci (1452-1519): first record of camera obscura
- Johann Zahn (1641-1707): first portable camera
- Joseph Nicéphore Niépce (1765-1833): first photo

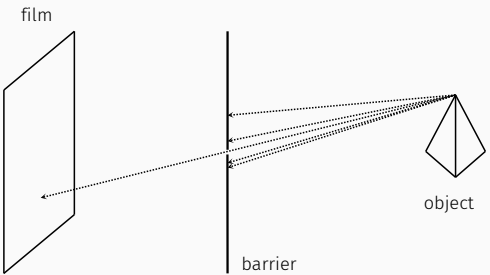
The pinhole camera



- Put a film in front of a scene
- Rays departing from the same point of the scene reach the film in different places
- Conversely, rays departing from different points of the scene, reach the film in the same place
- Ideally, a single light ray from the scene should reach the image plane in a given point

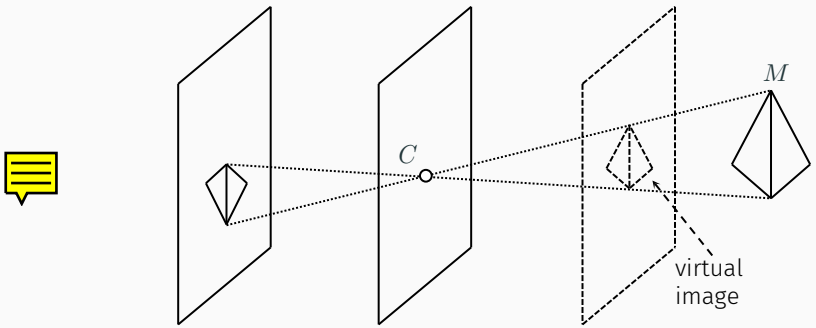
The pinhole camera





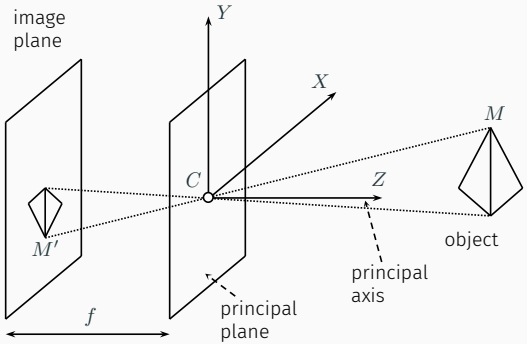
- Put a film in front of a scene
- Rays departing from the same point of the scene reach the film in different places
- Conversely, rays departing from different points of the scene, reach the film in the same place
- Ideally, a single light ray from the scene should reach the image plane in a given point
- Put in between a barrier with a hole

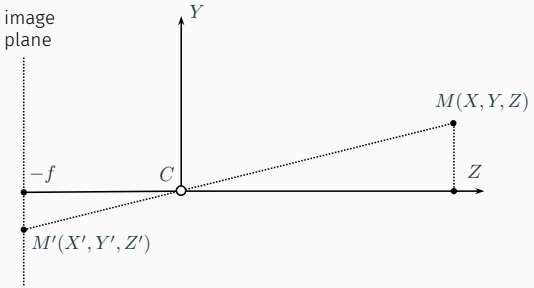
The pinhole camera (cont.)



- The image is reversed upside-down and left-right
- A *virtual image* is formed in front of the camera

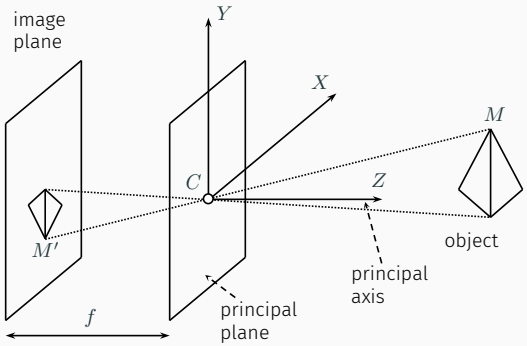
Perspective projection

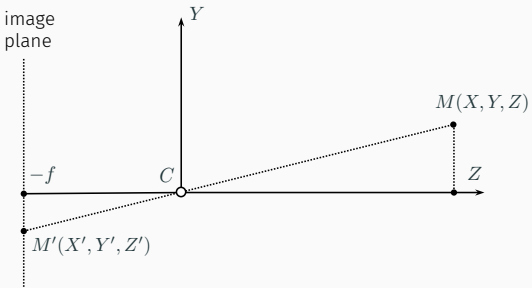




- The pinhole camera is defined by its *optical center* C and the image plane
- The distance of the image plane from C is the *focal length* f
- *Principal axis*: the perpendicular to the image plane through C
- *Principal plane*: the plane parallel to the image plane containing C
- Point M is projected to M' into the image plane

Perspective projection (cont.)



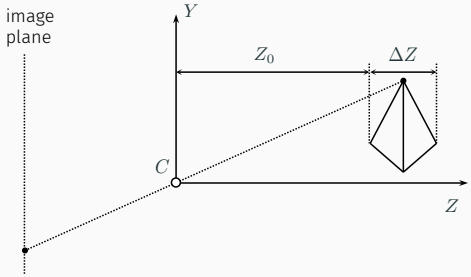


- Attach a reference frame to C, where the Z-axis is the principal axis
- By similarity of triangles, it follows that the 3D point $[X,\, Y,\, Z]^T$ is mapped to the point

$$[X',\, Y',\, Z']^T = [-\frac{f}{Z}X, -\frac{f}{Z}Y, -f]^T$$

- $\frac{f}{Z}$ is the *perspective scale factor*
- Farther away objects (larger Z) appear smaller

Weak perspective



- If the object is thin w.r.t. its distance from the camera, then the perspective scale factor is roughly constant:

$$\frac{f}{Z_0 + \Delta Z} \approx \frac{f}{Z_0}$$

- Then, perspective projection can be approximated by a *scaled orthographic projection*

$$X' = -\frac{f}{Z_0} X, \quad Y' = -\frac{f}{Z_0} Y$$

- Rule of thumb due to Leonardo da Vinci: $\frac{\Delta Z}{Z_0} < \frac{1}{10}$

Perspective vs orthographic

Perspective image

Orthographic image

Aperture

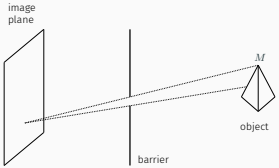
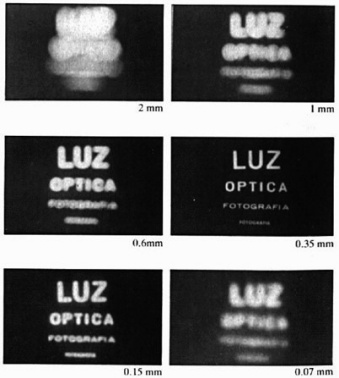
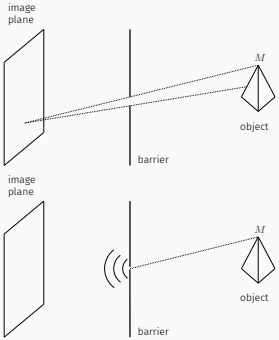
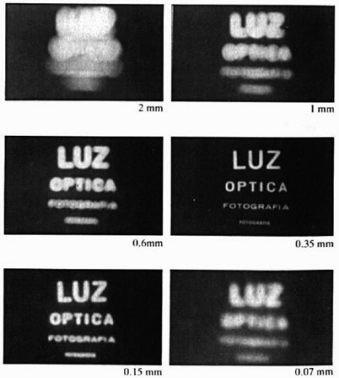


image
plane
M
object
barrier


2 mm
1 mm
LUZ
OPTICA
0.6mm
LUZ
OPTICA
FOTOGRAFIA
0.35 mm
LUZ
OPTICA
FOTOGRAFIA
0.15 mm
0.07 mm

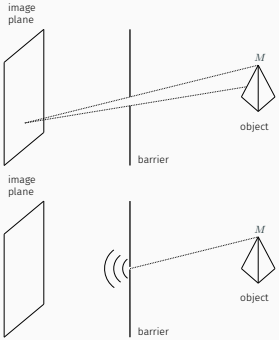
Aperture

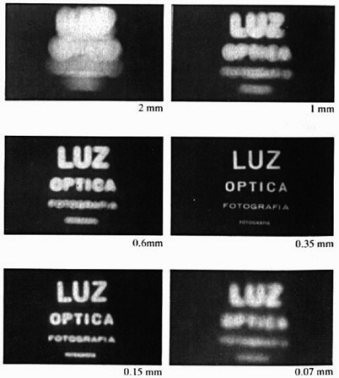




- Pinhole too big: many directions are averaged, blurring the image

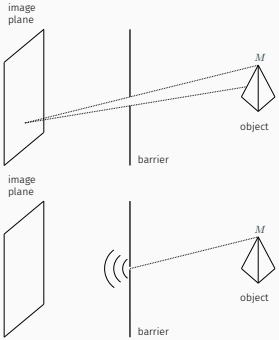
Aperture

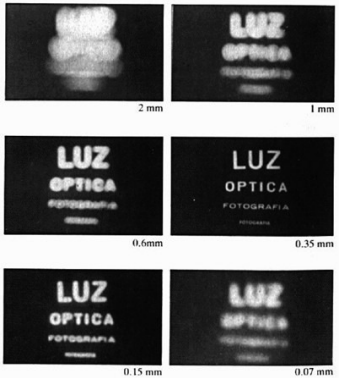




- Pinhole too big: many directions are averaged, blurring the image
- Pinhole too small: diffraction effects blur the image

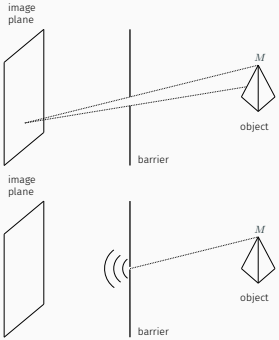
Aperture

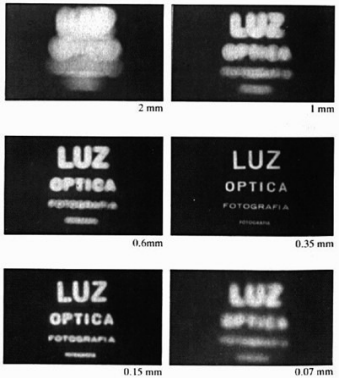




- Pinhole too big: many directions are averaged, blurring the image
- Pinhole too small: diffraction effects blur the image
- Generally, pinhole cameras are dark, because a small set of rays hits the screen.

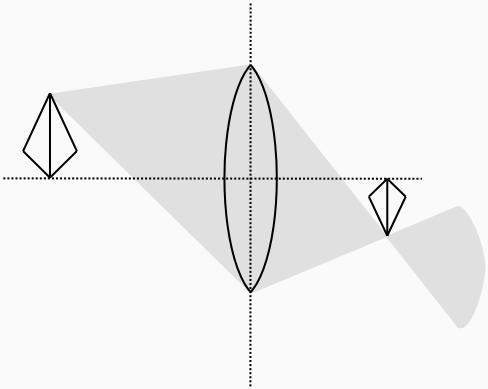
Aperture





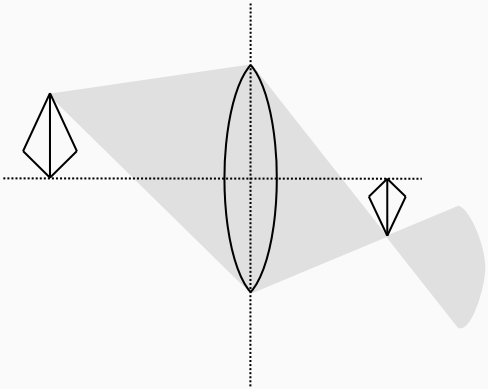
- Pinhole too big: many directions are averaged, blurring the image
- Pinhole too small: diffraction effects blur the image
- Generally, pinhole cameras are dark, because a small set of rays hits the screen.
- Solution: add lenses

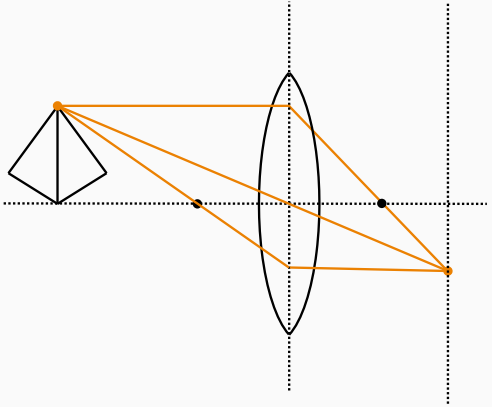
Lenses



- A lens collects rays departing from the same points and focuses them onto the screen

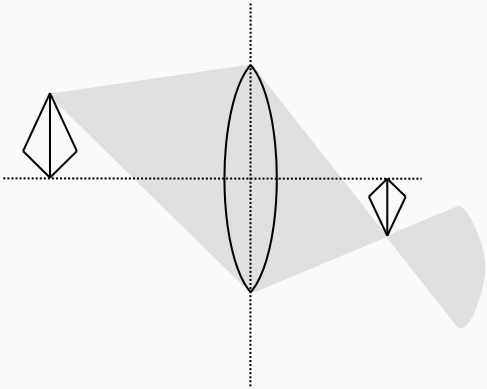
Lenses

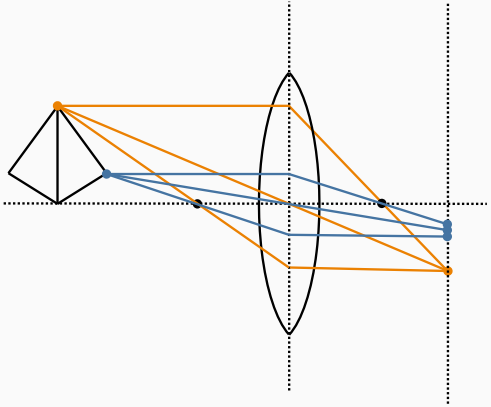




- A lens collects rays departing from the same points and focuses them onto the screen

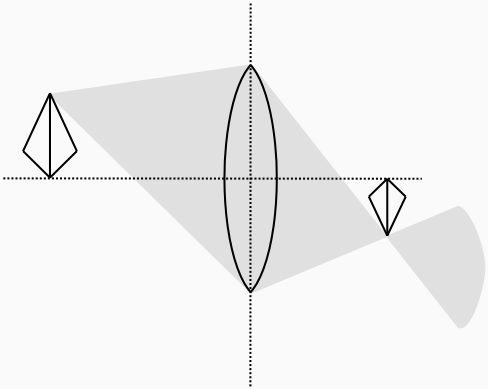
Lenses

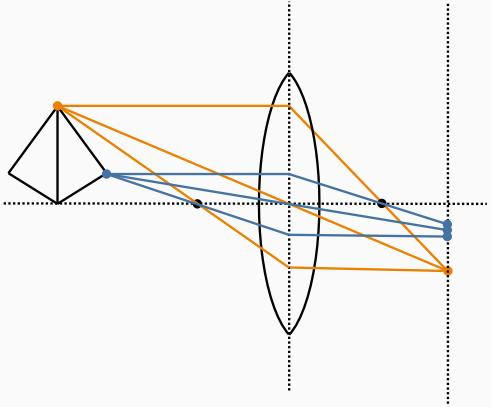




- A lens collects rays departing from the same points and focuses them onto the screen
- There is a specific distance at which objects are in focus

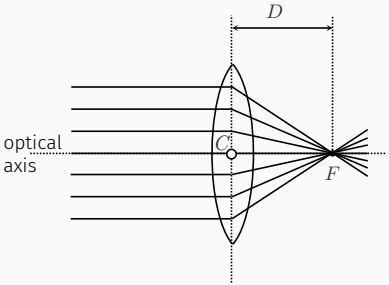
Lenses

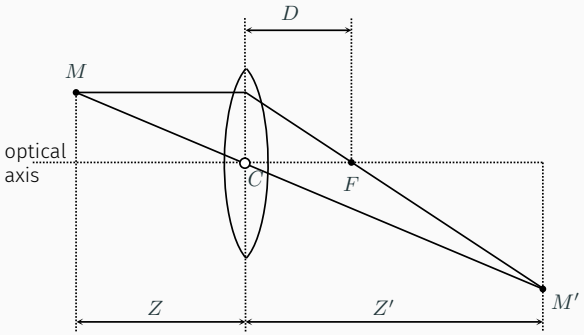




- A lens collects rays departing from the same points and focuses them onto the screen
- There is a specific distance at which objects are in focus
- Perspective projection is still valid within the *thin lens assumption*

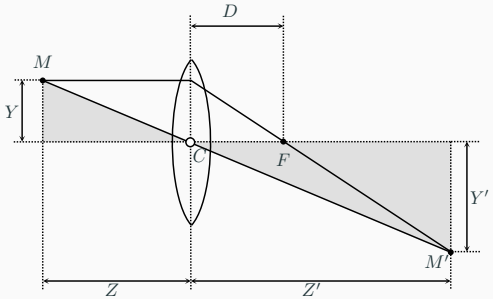
Thin lens

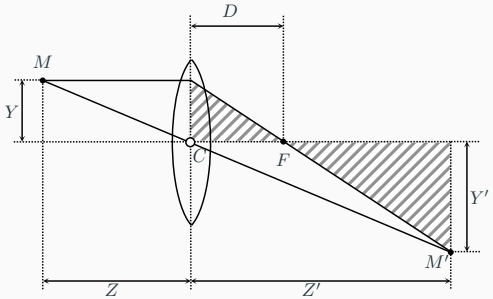




- A *thin lens* is composed of a single piece of glass with very low, equal curvature on both sides
- Any ray that enters parallel to the axis on one side of the lens proceeds towards the *focal point* F
- Any ray that passes through the center of the lens C does not change its direction
- The distance D from the center to the focal point is called *focal length*
- The image M' of M can be found by intersecting two rays

Thin lens equation





- Based on triangle similarity:

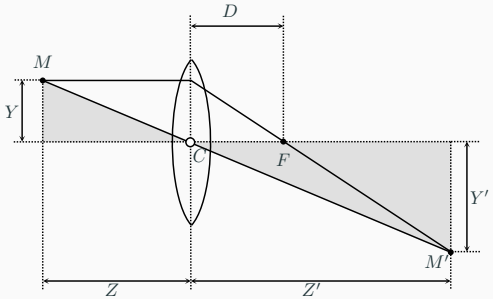
$$\frac{Y'}{Y} = \frac{Z'}{Z}, \quad \frac{Y'}{Y} = \frac{Z' - D}{D}$$

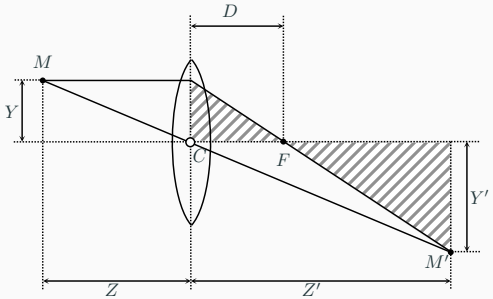
- Thus we get the *thin lens equation*

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D}$$

- Scene points at distance Z from the lens are in sharp focus at a distance Z'

Thin lens equation





- Based on triangle similarity:

$$\frac{Y'}{Y} = \frac{Z'}{Z}, \quad \frac{Y'}{Y} = \frac{Z' - D}{D}$$

- Thus we get the *thin lens equation*

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D}$$

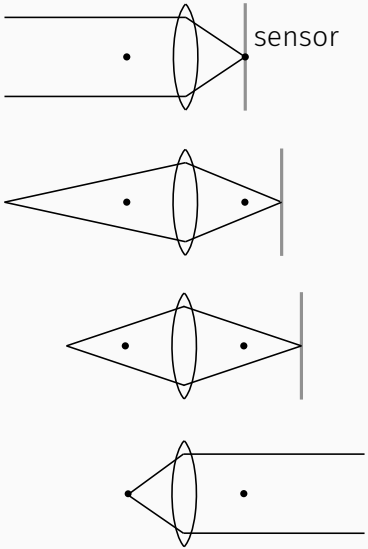
- Scene points at distance Z from the lens are in sharp focus at a distance Z'
- Point M is projected, when in focus, into the same position of a pinhole model having the optical center located in the lens center C

Changing the focusing distance

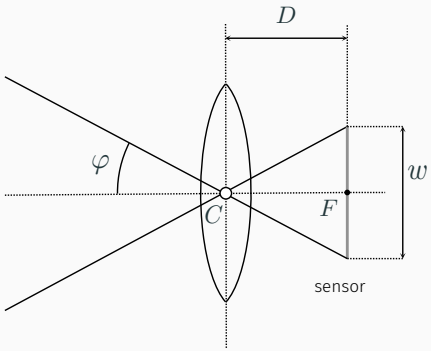
- to focus on objects at different distances, move sensor relative to lens
- at $Z = Z' = 2D$ we have $1:1$ imaging, because

$$\frac{1}{2D} + \frac{1}{2D} = \frac{1}{D}$$

- can't focus on objects closer to lens than its focal length D



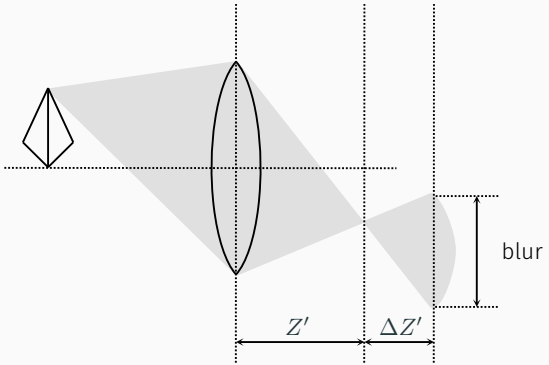
Field of view



- The *field of view* of a camera is the portion of the scene space that actually projects onto the sensor
- It depends on the focusing distance and on the sensor width w
- Conventionally it's defined when focusing at infinity:

$$\text{f.o.v.} \triangleq 2\varphi = 2 \arctan \frac{w}{2D}$$

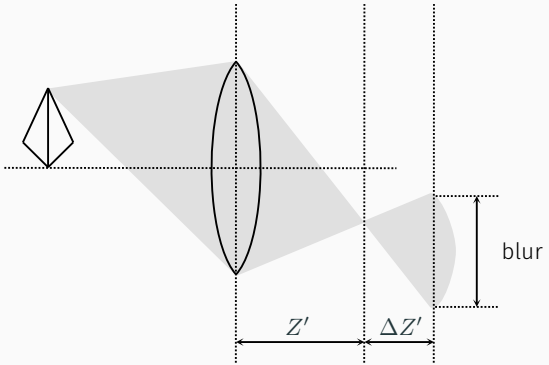
Blur circles

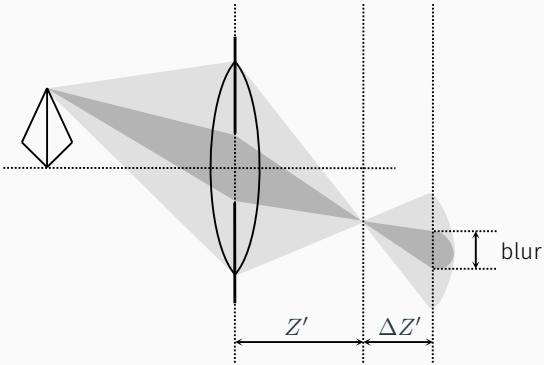


- In a plane lying $\Delta Z'$ from the focusing plane, the image is blurred
- Rays form a *blur circle*, whose size depends on $\Delta Z'$ and on the lens diameter d:

$$\text{blur circle diameter} = d\frac{|\Delta Z'|}{Z'}$$

Blur circles



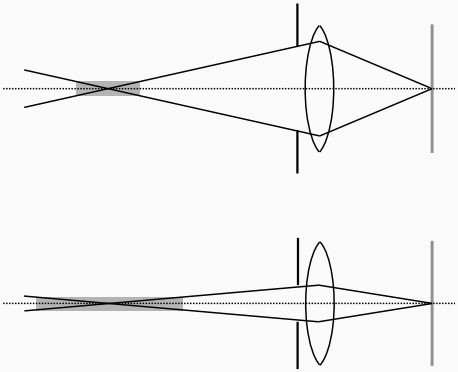


- In a plane lying $\Delta Z'$ from the focusing plane, the image is blurred
- Rays form a *blur circle*, whose size depends on $\Delta Z'$ and on the lens diameter d:

$$\text{blur circle diameter} = d\frac{|\Delta Z'|}{Z'}$$

- The smaller the diameter, the smaller the blur circle

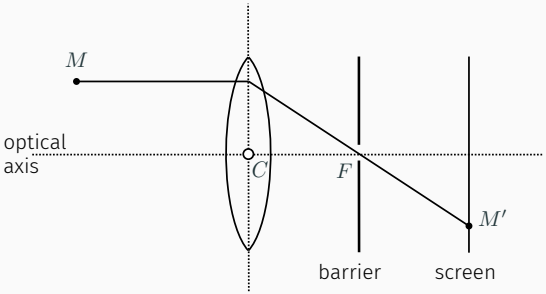
Depth of field



https://en.wikipedia.org/wiki/Depth_of_field

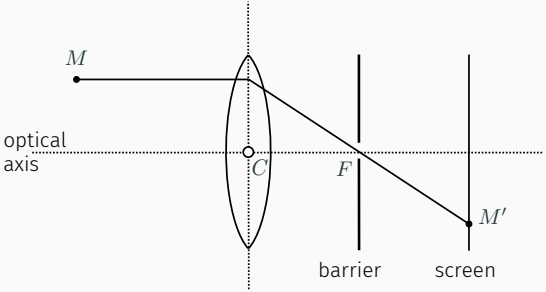
- For the same reason, the distance between the near and far planes that will keep the diameter of the blur circles below some threshold ε *increases as the lens diameter decreases*
- *Depth of field* increases as the lens diameter decreases

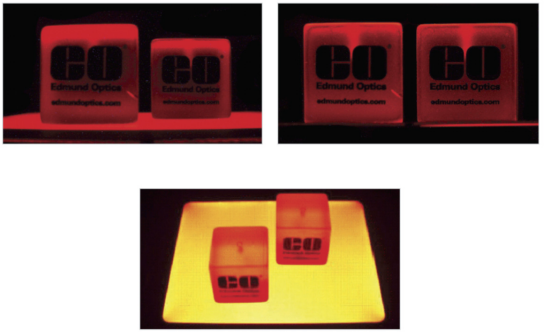
Telecentric optics



- Put a barrier with a hole in correspondence of the focus F
- The light beams from M are all blocked, except those parallel to the optical axis
- The image of M does not depend on Z, thus the device performs an orthographic projection
- *Telecentric* camera

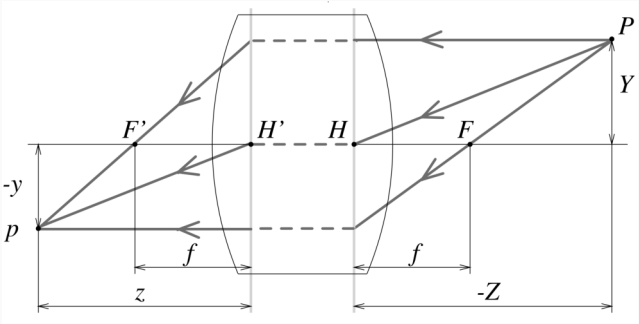
Telecentric optics





- Put a barrier with a hole in correspondence of the focus F
- The light beams from M are all blocked, except those parallel to the optical axis
- The image of M does not depend on Z, thus the device performs an orthographic projection
- *Telecentric* camera

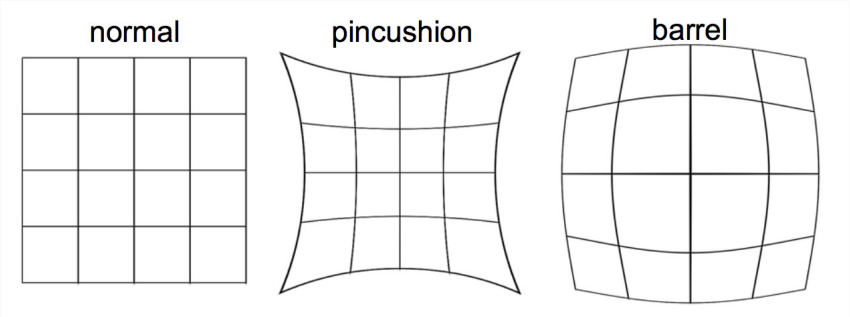
Real (thick) lenses



Real lenses introduce *aberrations* such as

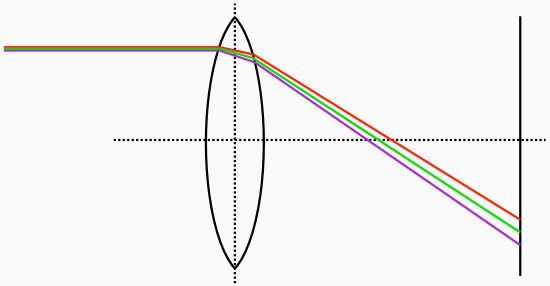
- Radial distortion
- Chromatic aberration

Radial distortion



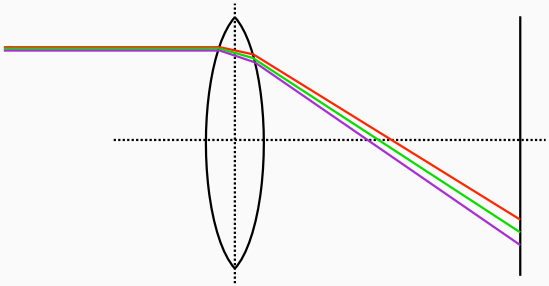
- Radial distortion: the image magnification decreases or increases as a function of the distance from the optical axis
- Cause: different portions of the lens have in practice different focal lengths
- *Pincushion distortion*: when the magnification increases
- *Barrel distortion*: when the magnification decreases

Chromatic aberration



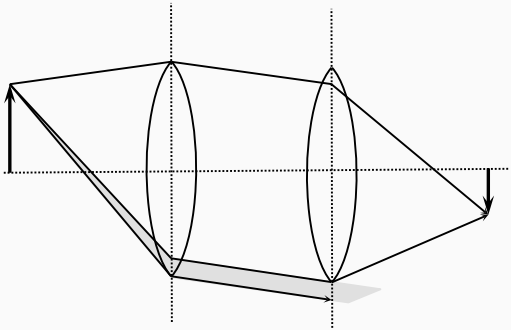
- Chromatic aberration: different components of the same light beam hit the image plane in different places
- Cause: the index of refraction of a transparent medium depends on the wavelength (or color) of the incident light rays

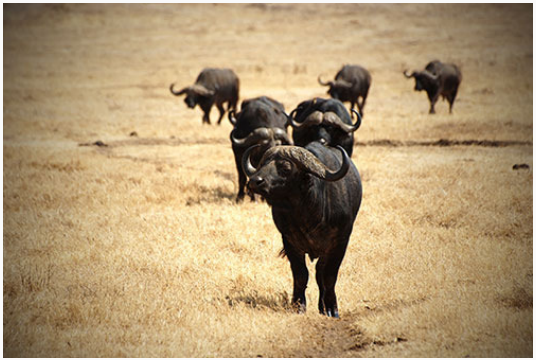
Chromatic aberration



- Chromatic aberration: different components of the same light beam hit the image plane in different places
- Cause: the index of refraction of a transparent medium depends on the wavelength (or color) of the incident light rays

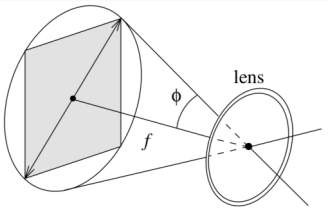
Vignetting

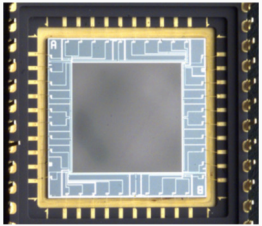




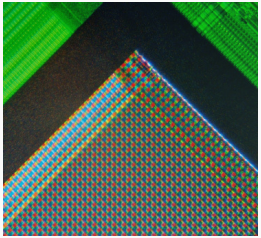
- *Compound lenses* may reduce aberrations, but they introduce a defect: vignetting
- Vignetting: brightness drops in the image periphery
- Cause: in compound lenses, light beams emanating from object points located off-axis are partially blocked by individual lens components

Sensing





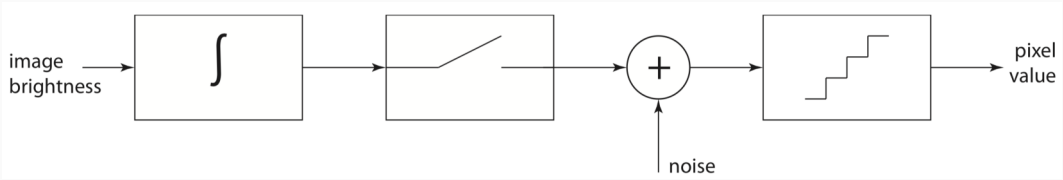
A sensor.



A micrograph of the corner of the photosensor array of a webcam digital camera. Credits: `https://commons.wikimedia.org/wiki/User:Natural_Philo`

- Sensor: a rectangular array of sensing elements (*pixels*);
- light → photosensitive element → current → charge of a capacitor;
- the more intense the light ⇒ the more the charge;
- the longer the exposure time ⇒ the more the charge;
- the capacitor is discharged periodically (*shutter time*).

Sensing (cont.)

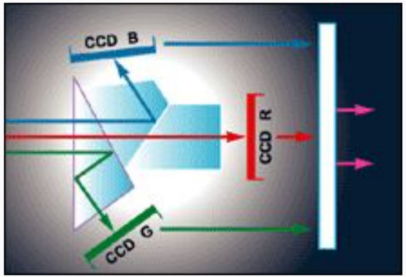


A simple sensor model is composed by

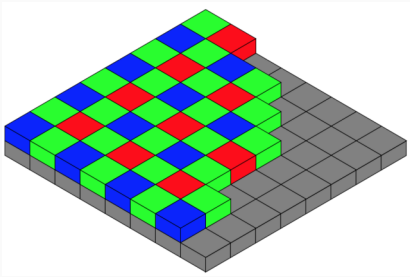
- an integrator;
- a sampler;
- a quantizer.

Noise statistics depend on input brightness and on camera settings (for instance, the gain).

Sensing (cont.)



Schematic diagram of a 3-sensor beam splitter. From `http://www.usa.canon.com/`.



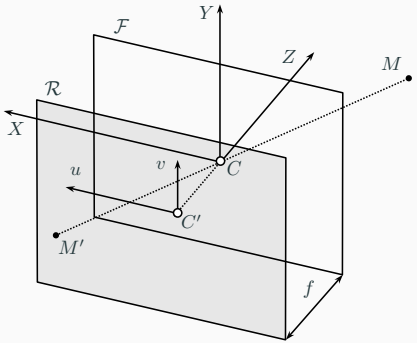
The Bayer color pattern. From `http://en.wikipedia.org/wiki/Image:Bayer_pattern_on_sensor.svg`.

Two approaches for sensing color:

- the 3-sensor method: a set of glass prisms uses a combination of internal reflection and refraction to split the incoming image into three, each reaching a different sensor;
- the Bayer mosaic: half of the pixels are sensitive to the green band of the light spectrum, and one quarter each to blue and red. This is consistent with the distribution of color-sensitive cones in the human retina. Color components that are not sensed (two per pixel), are inferred by interpolation.

Camera model

Simplified model



- Consider a simplified model of the camera
- (X, Y, Z) is the world reference frame, centered in $C \in \mathcal{F}$ (the focal plane)
- The Z axis corresponds to the optical axis
- (u, v) is the image frame, centered in the intersection between the optical axis and the image plane
- u and v are oriented as X and Y

- The perspective projection is a map $\mathbb{R}^3 \longrightarrow \mathbb{R}^2$

$$(x, y, z) \longrightarrow \left(-\frac{fx}{z}, -\frac{fy}{z}\right) \tag{1}$$

- The map is nonlinear (due to division by z)
- A linear map can be recovered by using *homogeneous coordinates*

Homogeneous coordinates

Homogeneous coordinates allow to add *improper* points (points at infinite) to the usual Euclidean plane (or space).

The basic idea is to represent a point (x, y) of the plane as a *triplet* (u, v, w) of real numbers, in such a way that

$$x = \frac{u}{w} \qquad \text{and} \qquad y = \frac{v}{w}.$$

Clearly, the triplets (u, v, w) and $\lambda(u, v, w)$, for $\lambda \neq 0$ represent the same point. The null triplet $(0, 0, 0)$ does not represent any point.

Definition

The *projective plane* $\mathbb{P}^2$ is the set of equivalence classes of triplets of real numbers $(u, v, w) \neq (0, 0, 0)$ where the following equivalence holds:

$$(u, v, w) \sim \lambda(u, v, w) \qquad \forall \lambda \neq 0.$$

The triplet entries are the *homogeneous coordinates* of the point of $\mathbb{P}^2$.

A similar definition holds for $\mathbb{P}^3$, where the points are represented by *four* real numbers.

Homogeneous coordinates (cont.)

- Points whose last coordinate is zero, are said to be *improper*. They represent a point at infinite *along a direction specified by the non-zero coordinates*.
- Points whose last coordinate is not zero, are said to be *proper*.

- For representing *proper* points we can simply "add one coordinate". With no loss of generality it may be taken as 1, obtaining the so called *augmented vector*.

$$(x, y) \longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (x, y, z) \longrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Converting from homogeneous coordinates amounts to dividing by the last coordinate:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \longrightarrow (x/w, y/w) \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \longrightarrow (x/w, y/w, z/w)$$

Homogeneous coordinates (cont.)

- As already said, in homogeneous coordinates,

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \sim \lambda \begin{bmatrix} x \\ y \\ w \end{bmatrix} \qquad \forall \lambda \neq 0$$

  meaning that they represent the same point in the Euclidean space.

Let the augmented vector corresponding to x be denoted by $\bar{x}$.

- Translation: a translation of t may be written as

$$x' = x + t$$

or

$$x' = [I \mid t]\,\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} I & t \\ 0^\top & 1 \end{bmatrix} \bar{x}.$$

Transformations (cont.)

- Rotation and translation (rigid body motion):

$$x' = Rx + t$$

or

$$x' = [R \mid t]\,\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix} \bar{x}.$$

where R is an orthonormal matrix, thus $RR^\top = R^\top R = I$ and $\det(R) = 1$.

In the 2D case, given the angle θ of rotation, R takes the form:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Transformations (cont.)

- Similarity transformation (where s is a scale factor):

$$x' = sRx + t$$

or

$$x' = [sR \mid t]\,\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} sR & t \\ 0^\top & 1 \end{bmatrix} \bar{x}.$$

Transformations (cont.)

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	3	lengths	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	4	angles	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times 3}$	6	parallelism	

Hierarchy of 2D transformations.

Transformations (cont.)

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{3\times4}$	3	orientation	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{3\times4}$	6	lengths	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{3\times4}$	7	angles	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{3\times4}$	12	parallelism	

Hierarchy of 3D transformations.

Perspective projection matrix

Let m and m' be the vectors of homogeneous coordinates of M (w.r.t. the camera frame) and M' (w.r.t. the image frame):

$$m = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad m' = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix},$$

Then

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = z \begin{bmatrix} -fx/z \\ -fy/z \\ 1 \end{bmatrix} = \begin{bmatrix} -fx \\ -fy \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Thus, in matrix notation

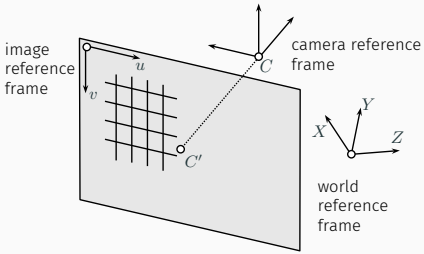
$$m' = \frac{1}{z} P m$$

or equivalently

$$m' \simeq P m$$

where $\simeq$ means "equal up to a scale factor". The matrix P is the *perspective projection matrix*.

General model



For a more general model we need to consider:

- the rigid transformation from the world reference frame to the camera frame
- the **pixelization** (from meters to pixel, in the image reference frame)

- The pixelization amounts to a rescaling followed by a translation:

$$V = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- (u_0, v_0) are the coordinates (in pixel) of C'
- k_u and k_v the reciprocal of the pixel size along u and v:

$$k_u = \frac{1}{s_u} \qquad k_v = \frac{1}{s_v} \qquad \left[\text{pixel} \times \text{m}^{-1}\right]$$

Thus the new P taking into account the sensor is

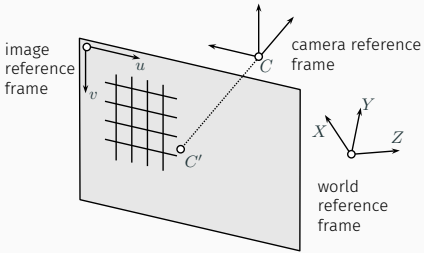
$$\begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K\,[I \mid 0]$$

where, by posing $fk_u = \alpha_u$ and $fk_v = \alpha_v$

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad [I \mid 0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- K is the *calibration matrix*
- K describes the camera *intrinsics* (those parameters that depend on the camera itself, as opposite to the camera's pose in space, which are called the *extrinsics*)
- Notice that K is upper triangular

General model (cont.)



Let's take into account:

- the **rigid transformation** from the world reference frame to the camera frame
- the pixelization (from meters to pixel, in the image reference frame)

Let m_c be the vector of homogeneous coordinates of M with respect to the camera reference frame and let

$$G = \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix}$$

be the rigid body motion from the world frame to the camera frame (a rotation R followed by a translation t). It follows that

$$m_c = Gm$$

The matrix G represents the camera extrinsics (the pose of the camera in space). The perspective projection is thus

$$m' \simeq K \, [I \mid 0] \, Gm$$

Perspective projection matrix, general case

The perspective projection matrix in the general case is

$$P = K\,[I \mid 0]\,G$$

It can be written alternatively as

$$P = K\,[R \mid t]$$

By partitioning R and t row-wise:

$$R = \begin{bmatrix} r_1^\top \\ r_2^\top \\ r_3^\top \end{bmatrix}, \qquad t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

we obtain

$$P = \begin{bmatrix} \alpha_u r_1^\top + u_0 r_3^\top & \alpha_u t_1 + u_0 t_3 \\ \alpha_v r_2^\top + v_0 r_3^\top & \alpha_v t_2 + v_0 t_3 \\ r_3^\top & t_3 \end{bmatrix} \tag{2}$$

1. A more general form of K takes into account a further intrinsic parameter (the *skew angle* ϑ between the u and v axes, usually very close to $\pi/2$), thus

$$K = \begin{bmatrix} \alpha_u & \alpha_u \cot\vartheta & u_0 \\ 0 & \alpha_v / \sin\vartheta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

   In that case, (2) becomes

$$P = \begin{bmatrix} \alpha_u r_1^\top - \frac{\alpha_u}{\tan\vartheta} r_2^\top + u_0 r_3^\top & \alpha_u t_1 - \frac{\alpha_u}{\tan\vartheta} t2 + u_0 t_3 \\ \frac{\alpha_v}{\sin\vartheta} r_2^\top + v_0 r_3^\top & \frac{\alpha_v}{\sin\vartheta} t_2 + v_0 t_3 \\ r_3^\top & t_3 \end{bmatrix}$$

Notes and observations (cont.)

2. P is defined up to a scale factor (for $\lambda \neq 0$, λP results in the same projection as P). A perspective projection matrix is said to be *normalized* if has the form

$$\left[\begin{array}{ccc|c} * & * & * & * \\ * & * & * & * \\ \hline & a^\top & & * \end{array}\right]$$

   and $\|a\|^2 = a^\top a = 1$, i.e. a is a unit vector. Notice that (2) is normalized.

3. P has 12 entries, but (due to the scale factor) 11 degrees of freedom. Indeed, it depends on $6 + 5 = 11$ independent parameters (6 extrinsic and 5 intrinsic: $\alpha_u, \alpha_v, u_0, v_0, \vartheta$)

4. Partitioning P row-wise

$$P = \begin{bmatrix} p_1^\top \\ p_2^\top \\ p_3^\top \end{bmatrix}$$

we get

$$m' = Pm = \begin{bmatrix} p_1^\top m \\ p_2^\top m \\ p_3^\top m \end{bmatrix} \xrightarrow{\text{back to inhomogeneous}} \begin{cases} u = \dfrac{p_1^\top m}{p_3^\top m} \\ v = \dfrac{p_2^\top m}{p_3^\top m} \end{cases}$$

where u and v are the Cartesian coordinates of point M' in the image plane. Thus the last equation is the generalization of (1).

Characterization of perspective projection matrices

Theorem

(Faugeras, 1993) Let $P = [Q \mid q]$ be 3×4 matrix and let $q_i^\top$, $i = 1, 2, 3$ denote the rows of Q.

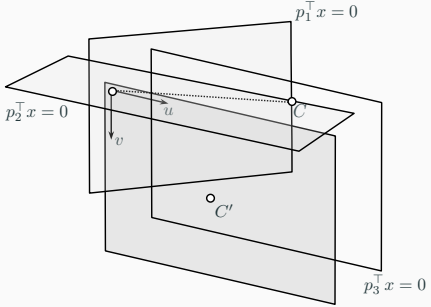
- *A necessary and sufficient condition for P to be a perspective projection matrix is that*

$$\det(Q) \neq 0$$

- *A necessary and sufficient condition for P to be a zero-skew perspective projection matrix is that*

$$\det(Q) \neq 0 \qquad \text{and} \qquad (q_1 \times q_3)^\top (q_2 \times q_3) = 0$$

Center of projection



The focal plane $\mathcal{F}$ contains the points that are projected to the infinite (because the straight line through C does not intersects the image plane). Thus its equation is

$$p_3^\top x = 0.$$

The planes $p_1^\top x = 0$ and $p_2^\top x = 0$ project to the axes $u = 0$ and $v = 0$, respectively.

The center of projection C belongs to all the three planes, thus it is defined by:

$$\begin{cases} p_1^\top x = 0 \\ p_2^\top x = 0 \\ p_3^\top x = 0 \end{cases}$$

In other words, C is represented (in homogeneous coordinates) by the kernel of P.

Center of projection (cont.)

P has rank 3 by construction (both K and R are invertible) hence, by the Rank-Nullity Theorem

$$\dim(\ker P) = 4 - 3 = 1.$$

Let

$$\tilde{c} = [x_C \;\; y_C \;\; z_C]^\top$$

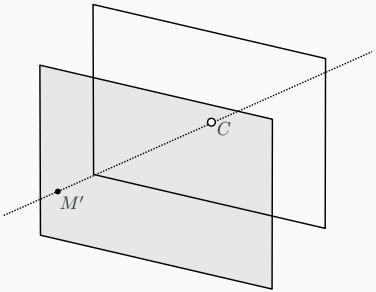
be the vector of the Cartesian coordinates of C w.r.t. the world frame. We can write

$$Pc = \left[\begin{array}{c|c} Q & q \end{array}\right] \begin{bmatrix} \tilde{c} \\ 1 \end{bmatrix} = 0$$

then, solving for $\tilde{c}$

$$\tilde{c} = -Q^{-1}q.$$

Optical ray



The optical ray of an image point M' is the locus of points in space that projects onto M'.

May be expressed as the line through C and a point α located at infinity that projects onto M'

$$\alpha = \begin{bmatrix} Q^{-1}m' \\ 0 \end{bmatrix}$$

thus a parametric representation (in homogeneous coordinates) of the optical ray is

$$m = c + \lambda \begin{bmatrix} Q^{-1}m' \\ 0 \end{bmatrix}, \qquad \lambda \in \mathbb{R}$$

Depth of a point

Consider a point M whose Cartesian coordinates are $\tilde{m} = [x_M \ y_M \ z_M]^\top$.

The projection equation with explicit scale factor ζ is

$$\zeta m' = Pm$$

Representing M in homogeneous coordinates as $m = [\tilde{m}^\top 1]^\top$ we get

$$\begin{bmatrix} \zeta u \\ \zeta v \\ \zeta \end{bmatrix} = P \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}$$

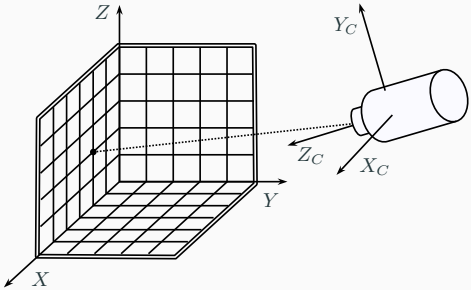
If P is normalized, it takes the form (2) thus we obtain, from the third scalar equation

$$\zeta = r_3^\top \tilde{m} + t_3$$

Recalling that $m_c = Gm$, it follows that ζ is the third coordinate of M w.r.t. the camera frame or the distance from M to the focal plane (*depth of the point* M).

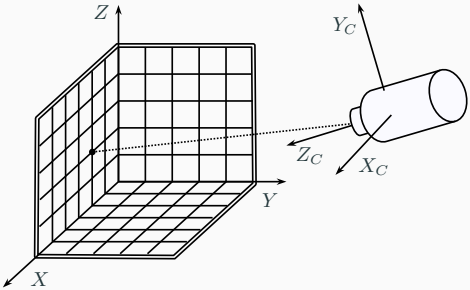
Camera calibration

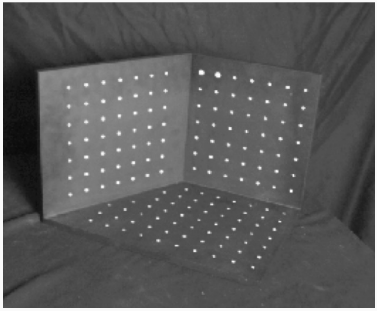
Camera calibration



- *Camera calibration problem*: estimate intrinsic and extrinsic camera parameters, based on some measurements

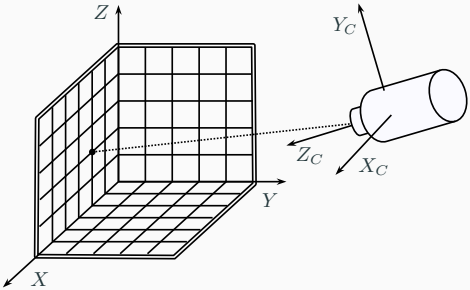
Camera calibration

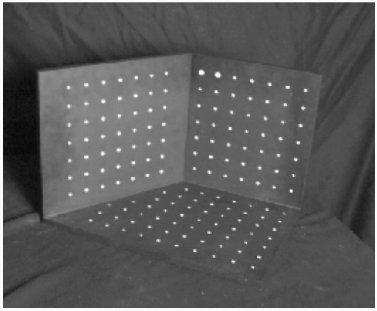




- *Camera calibration problem*: estimate intrinsic and extrinsic camera parameters, based on some measurements
- *Calibration pattern*: a 3D object of known geometry and generating image features which can be located accurately

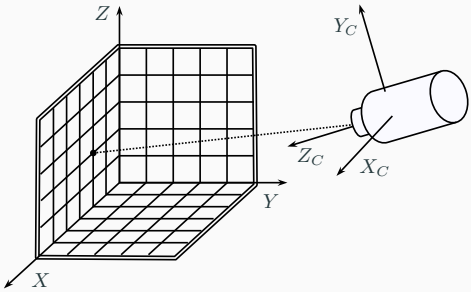
Camera calibration

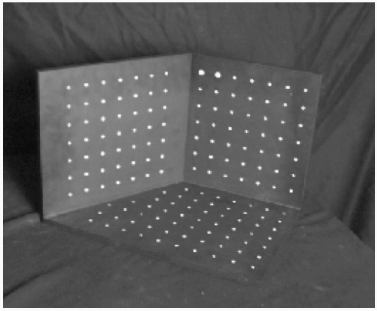




- *Camera calibration problem*: estimate intrinsic and extrinsic camera parameters, based on some measurements
- *Calibration pattern*: a 3D object of known geometry and generating image features which can be located accurately
- Parameters may be estimated based on a set of known 3D-2D correspondences

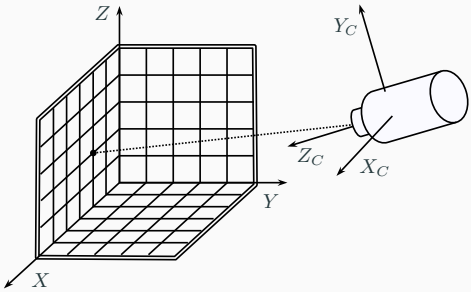
Camera calibration

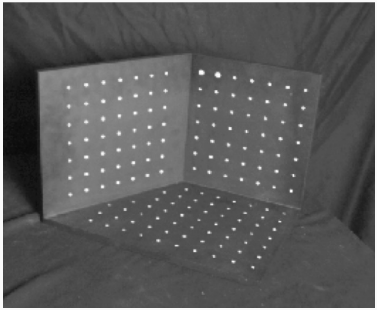




- *Camera calibration problem*: estimate intrinsic and extrinsic camera parameters, based on some measurements
- *Calibration pattern*: a 3D object of known geometry and generating image features which can be located accurately
- Parameters may be estimated based on a set of known 3D-2D correspondences
- *Direct methods*: formulate a problem in which the camera parameters are the unknowns or

Camera calibration





- *Camera calibration problem*: estimate intrinsic and extrinsic camera parameters, based on some measurements
- *Calibration pattern*: a 3D object of known geometry and generating image features which can be located accurately
- Parameters may be estimated based on a set of known 3D-2D correspondences
- *Direct methods*: formulate a problem in which the camera parameters are the unknowns or
- *Indirect methods*: first estimate P and then compute the camera parameters

Direct linear transform

A simple indirect method is called *Direct Linear Transform (DLT)*. Suppose to have n correspondences

$$m_{(i)} = \begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \\ 1 \end{bmatrix} \quad \longleftrightarrow \quad m'_{(i)} = \begin{bmatrix} u_{(i)} \\ v_{(i)} \\ 1 \end{bmatrix}, \qquad i = 1, \dots, n$$

For each correspondence, two scalar equations hold

$$\begin{cases} u_{(i)} = \dfrac{p_1^\top m_{(i)}}{p_3^\top m_{(i)}} \\ v_{(i)} = \dfrac{p_2^\top m_{(i)}}{p_3^\top m_{(i)}} \end{cases} \quad \Longrightarrow \quad \begin{cases} m_{(i)}^\top p_1 - u_{(i)} m_{(i)}^\top p_3 = 0 \\ m_{(i)}^\top p_2 - v_{(i)} m_{(i)}^\top p_3 = 0 \end{cases}$$

In more compact form:

$$\begin{bmatrix} m_{(i)}^\top & 0^\top & -u_{(i)} m_{(i)}^\top \\ 0^\top & m_{(i)}^\top & -v_{(i)} m_{(i)}^\top \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = 0$$

Direct linear transform (cont.)

By stacking the n pairs of equations we get

$$\underbrace{\begin{bmatrix} m_{(1)}^\top & 0^\top & -u_{(1)}m_{(1)}^\top \\ 0^\top & m_{(1)}^\top & -v_{(1)}m_{(1)}^\top \\ \vdots & \vdots & \vdots \\ m_{(n)}^\top & 0^\top & -u_{(n)}m_{(n)}^\top \\ 0^\top & m_{(n)}^\top & -v_{(n)}m_{(n)}^\top \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}}_{p} = 0$$

i.e. a homogeneous linear system

$$Ap = 0$$

of $2n$ equations in 12 unknowns (the entries of P):

$$p = [P_{11}\ P_{12}\ P_{13}\ P_{14}\ P_{21}\ P_{22}\ P_{23}\ P_{24}\ P_{31}\ P_{32}\ P_{33}\ P_{34}]^\top$$

Direct linear transform (cont.)

The system of equations may be written alternatively as

$$\begin{bmatrix} x_{(1)} & y_{(1)} & z_{(1)} & 1 & 0 & 0 & 0 & 0 & -u_{(1)}x_{(1)} & -u_{(1)}y_{(1)} & -u_{(1)}z_{(1)} & -u_{(1)} \\ 0 & 0 & 0 & 0 & x_{(1)} & y_{(1)} & z_{(1)} & 1 & -v_{(1)}x_{(1)} & -v_{(1)}y_{(1)} & -v_{(1)}z_{(1)} & -v_{(1)} \\ x_{(2)} & y_{(2)} & z_{(2)} & 1 & 0 & 0 & 0 & 0 & -u_{(2)}x_{(2)} & -u_{(2)}y_{(2)} & -u_{(2)}z_{(2)} & -u_{(2)} \\ 0 & 0 & 0 & 0 & x_{(2)} & y_{(2)} & z_{(2)} & 1 & -v_{(2)}x_{(2)} & -v_{(2)}y_{(2)} & -v_{(2)}z_{(2)} & -v_{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{(n)} & y_{(n)} & z_{(n)} & 1 & 0 & 0 & 0 & 0 & -u_{(n)}x_{(n)} & -u_{(n)}y_{(n)} & -u_{(n)}z_{(n)} & -u_{(n)} \\ 0 & 0 & 0 & 0 & x_{(n)} & y_{(n)} & z_{(n)} & 1 & -v_{(n)}x_{(n)} & -v_{(n)}y_{(n)} & -v_{(n)}z_{(n)} & -v_{(n)} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix} = 0$$

Direct linear transform (cont.)

- From a set of n point correspondences, we obtain a $2n \times 12$ coefficient matrix A
- At least $n = 6$ points are needed for computing P
- In general, and in absence of measuring errors, A has rank 11, and the solution is the null-space of A
- In practice, A will be full-rank and the system can be solved by linear least squares:

$$\underset{||p||=1}{\operatorname{argmin}} ||Ap||^2 \tag{3}$$

  where the constraint forces to discard the trivial solution $p = 0$ (which is not acceptable because $P \neq 0$). Notice that since P is defined up to a scale factor, any constraint of the form $||p|| = \lambda \neq 0$ will result in the same actual estimated projection.
- The solution of (3) is well-known to be the right singular vector of A corresponding to the smallest singular value. If

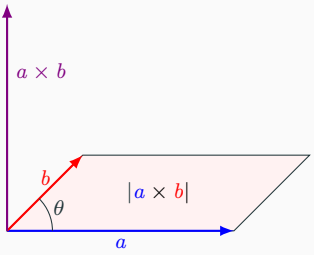
$$A = U \Sigma V^\top$$

  then $p = v_{12}$ i.e. the rightmost column of V.

Cross product

The *cross product* $a \times b$ is a vector defined as

$$a \times b = \begin{bmatrix} \det \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \\ -\det \begin{bmatrix} a_1 & a_3 \\ b_1 & b_3 \end{bmatrix} \\ \det \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \end{bmatrix}.$$



The cross product is null if and only $b = \lambda a$ for any scalar λ. It may be obtained as a matrix multiplication with a skew-symmetric matrix. Let

$$[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Then $[a]_\times \, b = a \times b$. Notice that $[a]_\times$ must be rank two because it must satisfy

$$[a]_\times \, b = 0 \qquad \Rightarrow \qquad b = \lambda a.$$

Definition

Given the matrices $A^{m\times n}$ and $B^{p\times q}$, the Kronecker product is the $mp \times nq$ matrix:

$$A \otimes B = \begin{bmatrix} A_{11}B & \dots & A_{1n}B \\ \vdots & & \vdots \\ A_{m1}B & \dots & A_{mn}B \end{bmatrix}$$

The product is defined for any pair of matrices, is non commutative, and enjoys the following properties:

1. $(A \otimes B) \otimes C = A \otimes (B \otimes C)$;
2. $(A \otimes B)^\top = A^\top \otimes B^\top$;
3. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$;
4. For square A and B: if λ_A is an eigenvalue of A and λ_B is an eigenvalue of b, then $\lambda_A \lambda_B$ is an eigenvalue of $A \otimes B$;
5. $\mathsf{rank}(A \otimes B) = \mathsf{rank}(A)\,\mathsf{rank}(B)$.

Kronecker product and vector operator (cont.)

To state another property we must introduce the *vector operator* (Henderson and Searle, 1979).

Definition

Given a matrix $X = [x_1 \; x_2 \; \dots x_n]$, the vector operator vec is the operator that stacks the columns one underneath the other to a single vector:

$$\text{vec}\,(X) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

The following useful identity holds:

$$\text{vec}\,(AXB) = \left(B^\top \otimes A\right) \text{vec}\,(X)\,. \tag{4}$$

The above identity is useful for "pulling out" the unknown X from a matrix equation.

Kronecker product and vector operator (cont.)

Example: consider the following linear system of equation in the unknown X

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

By taking the vec of both sides and applying the previous identity we get:

$$[B_1 \;\; B_2] \otimes \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ X_{12} \\ X_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

thus we obtain an easy to manage equation:

$$\begin{bmatrix} B_1 A_{11} & B_1 A_{12} & B_2 A_{11} & B_2 A_{12} \\ B_1 A_{21} & B_1 A_{22} & B_2 A_{21} & B_2 A_{22} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ X_{12} \\ X_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Kronecker product and vector operator (cont.)

For square matrices, the vech ("vector-half") turns out to be useful:

Definition

Given a *square* $n \times n$ matrix $X = [x_1 \; x_2 \; \dots x_n]$, the vector-half operator vech is the operator that stacks in a single column the elements of X, *which are on or below the diagonal*, taken in column-wise order. Thus, $\mathsf{vech}(X)$ has $n(n+1)/2$ elements.

For example:

$$\mathsf{vech}\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) = \begin{bmatrix} a \\ d \\ g \\ e \\ h \\ i \end{bmatrix}$$

There exists a $n^2 \times n(n+1)/2$ matrix D_n, called *duplication matrix*, such that:

$$D_n \, \mathsf{vech}(X) = \mathsf{vec}(X).$$

Alternative derivation for Direct Linear Method

The projection equation states, for each correspondence $i = 1, \ldots, n$:

$$m'_{(i)} \simeq P m_{(i)}.$$

Take the cross product of both sides by the vector $P m_{(i)}$:

$$m'_{(i)} \times P m_{(i)} \simeq P m_{(i)} \times P m_{(i)} = 0.$$

Thus we can get rid of the arbitrary scale factor and write:

$$m'_{(i)} \times P m_{(i)} = 0,$$

which is equivalent to

$$\left[m'_{(i)}\right]_\times P m_{(i)} = 0,$$

and, by transposing, to

$$m_{(i)}^\top P^\top \left[m'_{(i)}\right]_\times^\top = 0^\top.$$

Alternative derivation for Direct Linear Method (cont.)

By applying the vector operator to both sides, and recalling that

$$\mathsf{vec}\,(AXB) = \left(B^\top \otimes A\right) \mathsf{vec}\,(X)\,,$$

we obtain a triplet of linear equations in the unknown $\mathsf{vec}(P^\top)$:

$$\left(\left[m'_{(i)}\right]_\times \otimes m_{(i)}^\top\right) \mathsf{vec}\,(P^\top) = 0,$$

or, by expanding the coefficients:

$$\begin{bmatrix} 0^\top & -m_{(i)}^\top & v_{(i)} m_{(i)}^\top \\ m_{(i)}^\top & 0^\top & -u_{(i)} m_{(i)}^\top \\ -v_{(i)} m_{(i)}^\top & u_{(i)} m_{(i)}^\top & 0^\top \end{bmatrix} \mathsf{vec}\,(P^\top) = 0.$$

Notice that only two equations are independent because the Kronecker product $\left(\left[m'_{(i)}\right]_\times \otimes m_{(i)}^\top\right)$ is rank two (the two factors are rank two and rank one, respectively).

Degenerate point configurations

- For random configurations of $n \geq 6$ points, A will have rank 11 (in absence of measuring errors) or 12
- In either case, a unique (up to a scale factor) solution can be found.
- Are there any particular point configurations (*degenerate* configurations) such that P cannot be uniquely determined?
- The degenerate configurations are those for which

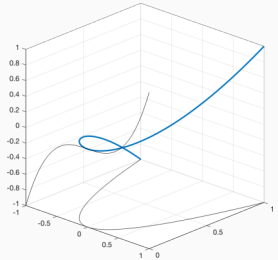
$$\dim(\ker A) \geq 2 \qquad \Longrightarrow \qquad \operatorname{rank}(A) \leq 10$$

  meaning that the space of solutions has dimension greater than one.
- Coplanar points: there exist a, b, c and d such that

$$ax_{(i)} + by_{(i)} + cz_{(i)} + d = 0, \qquad i = 1, \ldots, n$$

  thus it is easy to prove that $\operatorname{rank}(A) \leq 8$. Therefore, coplanar points are a degenerate configuration.

Degenerate point configurations (cont.)



A twisted cubic.

A complete characterization of degenerate configurations is rather complicated.

The most important degenerate configurations are:

- the points all lie on the union of a plane and a single straight line containing the camera center;
- the camera and the points all lie on a twisted cubic passing through the origin.

In practice: “take many more than six non-coplanar points”.

Non-linear refinement

- The DLT minimizes the *algebraic* residual $||Ap||^2$
- The algebraic residual has no geometrical meaning and is not invariant w.r.t. changes of the reference frame
- Instead, the *geometric* residual:

$$\varepsilon(P) = \sum_{i=1}^{n} \left(\frac{p_1^\top m_{(i)}}{p_3^\top m_{(i)}} - u_{(i)} \right)^2 + \left(\frac{p_2^\top m_{(i)}}{p_3^\top m_{(i)}} - v_{(i)} \right)^2$$

  best expresses the actual need (minimizing the *reprojection error*, i.e. the distance between the estimated and the measured projections)
- The resulting *non-linear optimization problem* can be solved iteratively, starting from an initial guess obtained via the DLT.

Computing the camera parameters

Given the estimated perspective projection matrix P, the intrinsic and extrinsic camera parameters can be computed as follows.

The matrix P is given in the form

$$P = \left[\begin{array}{c|c} Q & q \end{array}\right]$$

where Q is invertible (otherwise P does not represent a perspective projection).

We want to express P as

$$P = K \left[\begin{array}{c|c} R & t \end{array}\right] = \left[\begin{array}{c|c} KR & Kt \end{array}\right]$$

where K is upper triangular and R is orthogonal (a rotation matrix).

Thus $Q = KR$ and $q = Kt$.

Computing the camera parameters (cont.)

Recall that any real square matrix can be written as the product of an orthogonal matrix and an upper triangular matrix (QR factorization).

Let $Q^{-1} = US$ be the QR factorization of Q^{-1}. We can write:

$$Q = S^{-1}\,U^{-1}$$

where S^{-1} is upper triangular being the inverse of an upper triangular matrix, and U^{-1} is orthogonal as U is orthogonal. Hence a solution is

$$R = U^{-1}, \qquad K = S^{-1}, \qquad t = K^{-1}q.$$

Notice that P may contain an arbitrary scale factor that will affect K. The proper scale for K is obtained by imposing $K_{33} = 1$ (dividing K by K_{33}).

Computing the camera parameters (cont.)

A further observation is that it is not possible to recover the "true" focal length f from the 3D-2D correspondences only (i.e. without knowing the actual sensor size or the pixel size).

Indeed, recall that the calibration matrix K takes the form:

$$K = \begin{bmatrix} \alpha_u & \alpha_u \cot \vartheta & u_0 \\ 0 & \alpha_v / \sin \vartheta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fk_u & fk_u \cot \vartheta & u_0 \\ 0 & fk_v / \sin \vartheta & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where k_u and k_v are the reciprocal of the pixel sizes, along u and v respectively.

If we, for instance, double the focal length as well as the pixel sizes we get the same calibration matrix.

Thus, we can only recover α_u and α_v.

Dealing with radial distortion

- For more accurate camera modeling, *radial distortion* has to be taken into account
- Radial distortion appears at the periphery of the image, especially for short focal length

A simple model for radial distortion has two parameters: k_1 and k_2:

$$\begin{cases} \hat{u} = (u - u_0)(1 + k_1 r_d^2 + k_2 r_d^4) + u_0 \\ \hat{v} = (v - v_0)(1 + k_1 r_d^2 + k_2 r_d^4) + v_0 \end{cases} \tag{5}$$

where u, v are the ideal projections (in absence of radial distortion), $\hat{u}, \hat{v}$ are the actual projections and

$$r_d^2 = \left(\frac{u - u_0}{\alpha_u}\right)^2 + \left(\frac{v - v_0}{\alpha_v}\right)^2.$$

Estimating k_1 and k_2

Equations (5) may be rewritten as

$$\begin{cases} \hat{u} - u = (u - u_0) r_d^2 k_1 + (u - u_0) r_d^4 k_2 \\ \hat{v} - v = (v - v_0) r_d^2 k_1 + (v - v_0) r_d^4 k_2 \end{cases} \tag{6}$$

Assuming that P is known, the ideal (undistorted) coordinates (u, v) can be computed from the 3D coordinates (x, y, z) by applying the projection.

Thus we obtain a linear system (6) of two equations in the unknowns k_1 and k_2. Notice that r_d is also known because from P we get $u_0, v_0, \alpha_u, \alpha_v$.
In practice it is advisable to use many correspondences and solve the resulting overdetermined system

$$A \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = b$$

using the least squares:

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = (A^\top A)^{-1} A^\top b.$$

Compensating for radial distortion

Once k_1 and k_2 have been estimated, the radial distortion can be compensated by inverting the transformation. If we switch to normalized coordinates

$$x = \frac{u - u_0}{\alpha_u}, \qquad y = \frac{v - v_0}{\alpha_v}$$

equations (5) may be rewritten as

$$\begin{cases} \hat{x} = x\left(1 + k_1(x^2 + y^2) + k_2(x^4 + 2x^2y^2 + y^4)\right) \\ \hat{y} = y\left(1 + k_1(x^2 + y^2) + k_2(x^4 + 2x^2y^2 + y^4)\right) \end{cases} \tag{7}$$

Thus, the ideal (undistorted) coordinates x, y may be obtained from the actual (distorted) by solving a non-linear system of equations in the unknowns x, y, for instance by using Newton's method.

Newton's method

Iterative method for solving

$$f(x) = 0.$$

When $f : \mathbb{R} \longrightarrow \mathbb{R}$, using Taylor expansion:

$$f(x + \Delta x) \approx f(x) + \Delta x f'(x).$$

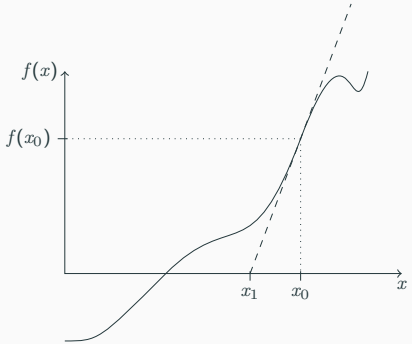
Suppose that the current estimate of the solution is x and the solution is $x + \Delta x$. Then $f(x + \Delta x) = 0$ and from the previous equation we get

$$\Delta x = -\frac{f(x)}{f'(x)}.$$

Due to the approximation, we need to iterate until convergence (guaranteed if the initial guess is sufficiently close to the solution.)
When $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ we get

$$\Delta x = -J(x)^{-1} f(x),$$

where J is the Jacobian matrix $[J_{ij}] = \dfrac{\partial f_i(x)}{\partial x_j}$.



Camera calibration in the presence of radial distortion

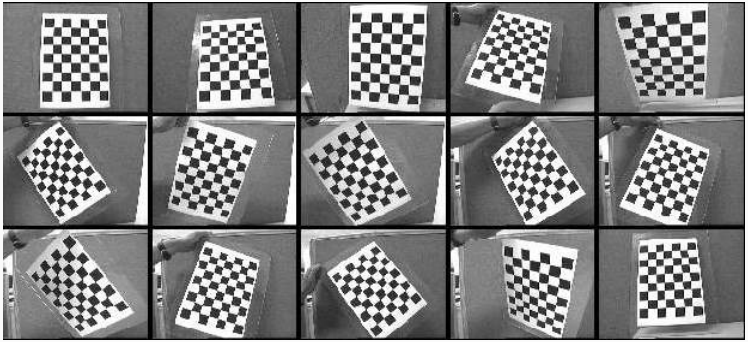
Given the correspondences

$$m_{(i)} = \begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \\ 1 \end{bmatrix} \quad \longleftrightarrow \quad m'_{(i)} = \begin{bmatrix} u_{(i)} \\ v_{(i)} \\ 1 \end{bmatrix}, \qquad i = 1, \ldots, n$$

do the following steps:

1. estimate P
2. get intrinsic parameters from P
3. estimate k_1 and k_2
4. compensate for radial distortion and get new $m'_{(i)}, \quad i = 1, \ldots, n$
5. go to step 1 until convergence of P and k_1, k_2

Zhang's method



- DLT calibration requires an image of many (at least six) non co-planar points. Building an accurate 3D calibration object may be difficult.
- Zhang's calibration method (Zhang, 2000) requires many (at least three) images of a plane (easier to get).
- Zhang's method is based on *homography*.

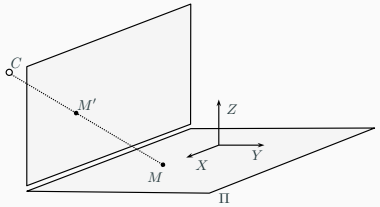
Definition

A homography is an isomorphism of $\mathbb{P}^n$, i.e. a map

$$x \longrightarrow Hx, \qquad H^{(n+1)\times(n+1)}, \text{ non-singular}$$

H and λH, $\lambda \neq 0$ represent the same homography.

Zhang's method (cont.)



The map from a 3D plane Π to its perspective image is easily seen to be a homography. Assume for simplicity that the plane has equation $z = 0$. Then

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [p_1 \ p_2 \ p_3 \ p_4] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \underbrace{[p_1 \ p_2 \ p_4]}_{\doteq H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where the matrix H is non-singular unless for degenerate cases.

Given at least 4 pairs of original and projected points, a matrix H corresponding to a homography can be computed via DLT (proof at the end of the lecture).

Zhang's method (cont.)

Suppose a homography H of a 3D plane Π to the image plane is known (e.g. estimated from measurements):

$$H = \lambda \left[p_1 \; p_2 \; p_4 \right].$$

Since $P = K \left[R | t \right] = K \left[r_1 \; r_2 \; r_3 \; t \right]$ we get $p_1 = K r_1$, $p_2 = K r_2$, $p_4 = K t$ and by substitution:

$$H = \lambda K \left[r_1 \; r_2 \; t \right]. \tag{8}$$

Thus:

$$\begin{cases} h_1 = \lambda K r_1 \\ h_2 = \lambda K r_2 \end{cases} \Longrightarrow \begin{cases} r_1 = \dfrac{1}{\lambda} K^{-1} h_1 \\ r_2 = \dfrac{1}{\lambda} K^{-1} h_2 \end{cases} \tag{9}$$

Now, being R a rotation matrix, we have

$$\begin{cases} r_1^\top r_2 = 0 \\ r_1^\top r_1 = r_2^\top r_2 \end{cases} \xrightarrow{(9)} \begin{cases} h_1^\top (K K^\top)^{-1} h_2 = 0 \\ h_1^\top (K K^\top)^{-1} h_1 = h_2^\top (K K^\top)^{-1} h_2 \end{cases}$$

Zhang's method (cont.)

Thus, from a plane Π and the corresponding estimated homography H we get two scalar constraints:

$$\begin{cases} h_1^\top B h_2 = 0 \\ h_1^\top B h_1 = h_2^\top B h_2 \end{cases} \tag{10}$$

where $B = (KK^\top)^{-1}$. The constraints are linear in the unknown B.

By taking n planes and the corresponding estimated homographies

$$\Pi^{(k)}, H^{(k)} \quad k = 1, \ldots, n$$

we get $2n$ constraints on the same B.

How can we solve for B? First, note that $B = B^\top$ by construction, so B is symmetric; hence there are 6 unknowns. Let

$$b = [B_{11}\ B_{12}\ B_{22}\ B_{13}\ B_{23}\ B_{33}]^\top.$$

be the vector of the unknowns.

Zhang's method (cont.)

For $i, j \in \{1, 2\}$, it's easily verified that

$$h_i^\top B h_j = v_{ij}^T b$$

where

$$v_{ij} = \begin{bmatrix} H_{1i}H_{1j} \\ H_{1i}H_{2j} + H_{2i}H_{1j} \\ H_{2i}H_{2j} \\ H_{3i}H_{1j} + H_{1i}H_{3j} \\ H_{3i}H_{2j} + H_{2i}H_{3j} \\ H_{3i}H_{3j} \end{bmatrix}.$$

Thus, the two equations become

$$\begin{bmatrix} v_{12}^\top \\ (v_{11} - v_{22})^\top \end{bmatrix} b = 0.$$

For n planes, by stacking equations of the above form, we get

$$Vb = 0 \tag{11}$$

where V is $2n \times 6$.

Zhang's method (cont.)

For b being uniquely determined (up to a scale factor), V must have rank 5 (the solution being the null-space of V)

- At least $n = 3$ planes in general configuration are needed
- In practice, the more the planes, the better the accuracy
- In practice, due to noise, V will be full-rank and the system can be solved by linear least squares:

$$\underset{||b||=1}{\text{argmin}} \, ||Vb||^2 \tag{12}$$

  where the constraint forces to discard the trivial solution $b = 0$.

- The solution of (12) is well-known to be the right singular vector of V corresponding to the smallest singular value. If the singular value decomposition of V is

$$V = U\mathbf{\Sigma}S^\top$$

  then $b = s_6$ i.e. the rightmost column of S.

Zhang's method (cont.)

Given $B = (KK^\top)^{-1}$, the matrix K can be found by the Cholesky factorization. Recall that any real symmetric and positive definite matrix can be decomposed *uniquely* as the product of lower triangular matrix having strictly positive eigenvalues and its transpose (Cholesky factorization).

Let

$$B = LL^\top$$

be the Cholesky factorization of B, where L is lower triangular. On the other hand we have

$$B = (KK^\top)^{-1} = (K^\top)^{-1}K^{-1} = (K^{-1})^\top K^{-1}$$

where $(K^{-1})^\top$ is lower triangular, being the transpose of an upper triangular matrix. By comparison, we get

$$K^{-1} = L^\top \qquad \Longrightarrow \qquad K = (L^\top)^{-1}.$$

As usual, the proper scale for K is obtained by imposing $K_{33} = 1$.

Zhang's method (cont.)

Finally, the extrinsic parameters for each image are readily computed from (8):

$$r_1 = \lambda K^{-1} h_1, \qquad r_2 = \lambda K^{-1} h_2, \qquad t = \lambda K^{-1} h_3, \qquad r_3 = r_1 \times r_2$$

where $\lambda = 1/||K^{-1} h_1|| = 1/||K^{-1} h_2||$ and $\times$ denotes cross product.

Due to noise, the resulting matrix R may not be orthogonal.

The closest (in the Frobenius norm) orthogonal matrix R' to a given matrix R is known to be (see for instance (Golub and Van Loan, 1996), pg. 601)

$$R' = UV^\top$$

where $R = U\Sigma V^\top$ is the singular value decomposition of R.

Alternative derivation

Consider again the pair of equations (10):

$$h_1^\top B h_2 = 0$$
$$h_1^\top B h_1 = h_2^\top B h_2$$

and apply the vec operator and identity (4):

$$\left(h_2^\top \otimes h_1^\top\right) \text{vec}\,(B) = 0$$
$$\left(\left(h_2^\top \otimes h_1^\top\right) - \left(h_2^\top \otimes h_2^\top\right)\right) \text{vec}\,(B) = 0$$

Since B is symmetric, the actual unknowns are six. Thus we can use the vech operator and the duplication matrix D_3 obtaining

$$\left(h_2^\top \otimes h_1^\top\right) D_3 \,\text{vech}\,(B) = 0$$
$$\left(\left(h_2^\top \otimes h_1^\top\right) - \left(h_2^\top \otimes h_2^\top\right)\right) D_3 \,\text{vech}\,(B) = 0$$

By stacking the pairs of equations provided by each image, we finally get a homogeneous system of equations:

$$V \,\text{vech}\,(B) = 0,$$

which is the same as (11).

Estimating homographies

Consider a plane having $z = 0$. The perspective projection is then

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [p_1 \; p_2 \; p_3 \; p_4] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \underbrace{[p_1 \; p_2 \; p_4]}_{\doteq H} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\doteq m}.$$

Let H be partitioned row-wise as

$$H = \begin{bmatrix} h_1^\top \\ h_2^\top \\ h_3^\top \end{bmatrix}.$$

Then we get

$$\begin{cases} wu = h_1^\top m \\ wv = h_2^\top m \\ w = h_3^\top m \end{cases} \implies \begin{cases} u = \dfrac{h_1^\top m}{h_3^\top m} \\ v = \dfrac{h_2^\top m}{h_3^\top m} \end{cases} \implies \begin{cases} h_1^\top m - u h_3^\top m = 0 \\ h_2^\top m - v h_3^\top m = 0 \end{cases}$$

Estimating homographies (cont.)

By transposing we get

$$\begin{cases} m^\top h_1 - u m^\top h_3 = 0 \\ m^\top h_2 - v m^\top h_3 = 0 \end{cases},$$

and, in matrix form:

$$\begin{bmatrix} m^\top & 0^\top & -u m^\top \\ 0^\top & m^\top & -v m^\top \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0.$$

Hence, each correspondence provides two equations.

Since H has nine entries but is defined up to a scale factor, the actual degrees of freedom are eight. Therefore, at least four correspondences are necessary.

In practice, it is advisable to use many more than four points, obtaining a "tall" matrix of size $2N \times 9$. Due to measurement and modeling errors, the matrix will be rank nine, thus a least squares solution can be found by means of Singular Value Decomposition.

References

Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. MIT press.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix computations, third edition*. The John Hopkins University Press.

Henderson, H. V. and Searle, S. (1979). Vec and vech operators for matrices, with some uses in jacobians and multivariate statistics. *Canadian Journal of Statistics*, 7(1):65–81.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334.

554SM –Fall 2018

Lecture 2
Image Formation

END