Technical University of Cluj - Napoca
Computer Science Department

Sisteme de viziune in robotica

An2, Master Robotica - engleza

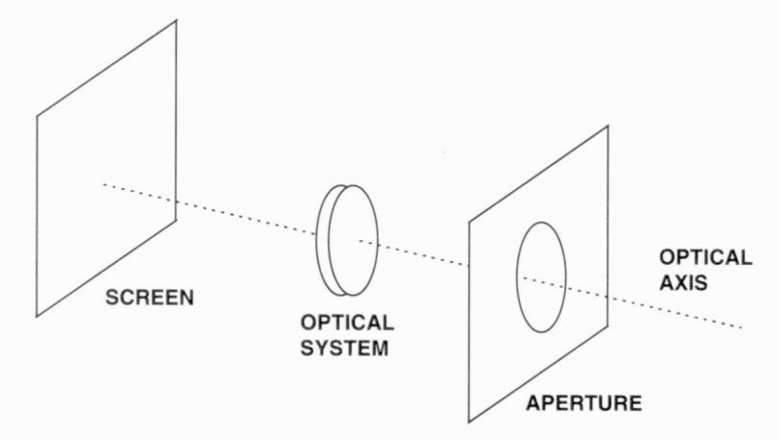
Image Representation

Purpose

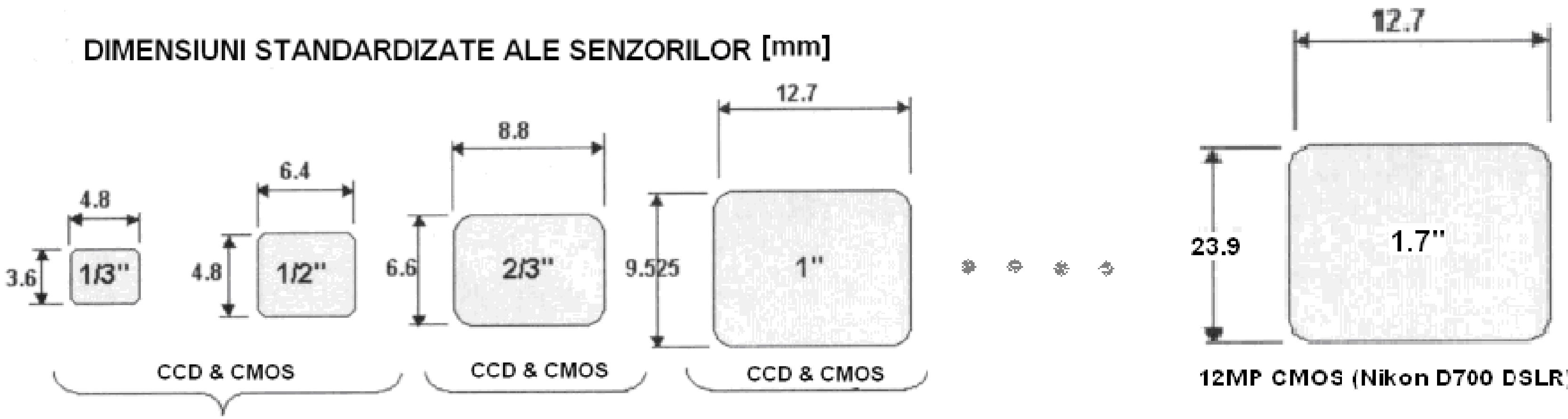
Presentation of the camera parameters

Principles of digital images formation

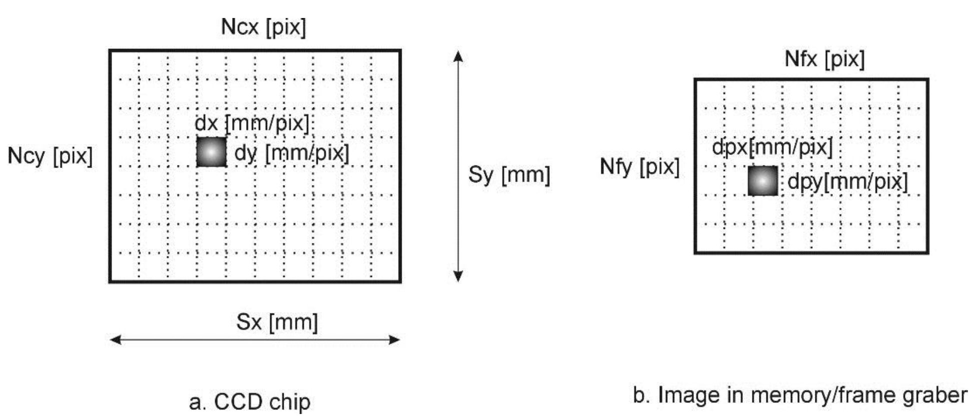
The basic elements of an imaging device



Sensor parameters



Standard camera sensor sizes



Parameters of the imager and image in memory

Sensor parameters

Sensor parameters:

Sx – width of the sensor chip [mm]
Sy – height of the sensor chip [mm]
Ncx – number of sensor elements in camera's x direction;
Ncy – number of sensor elements in camera's y direction;
dx – center to center distance between adjacent sensor elements in X (scan line) direction;

$$dx = Sx/Ncx;$$

dy - center to center distance between adjacent CCD sensor in the Y direction;

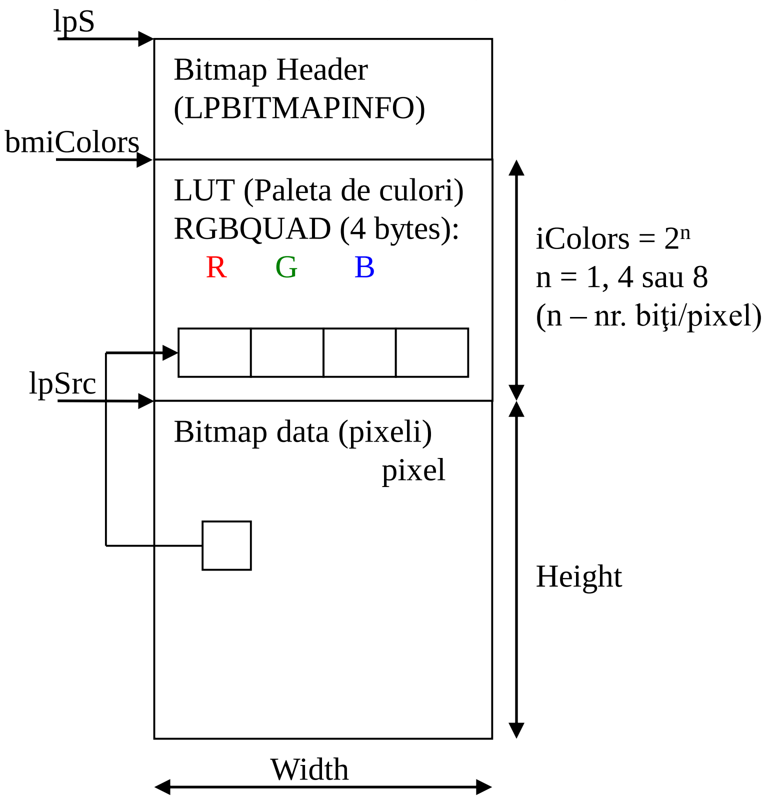
$$dy = Sy/Ncy;$$

Image parameters (related to the image in memory):

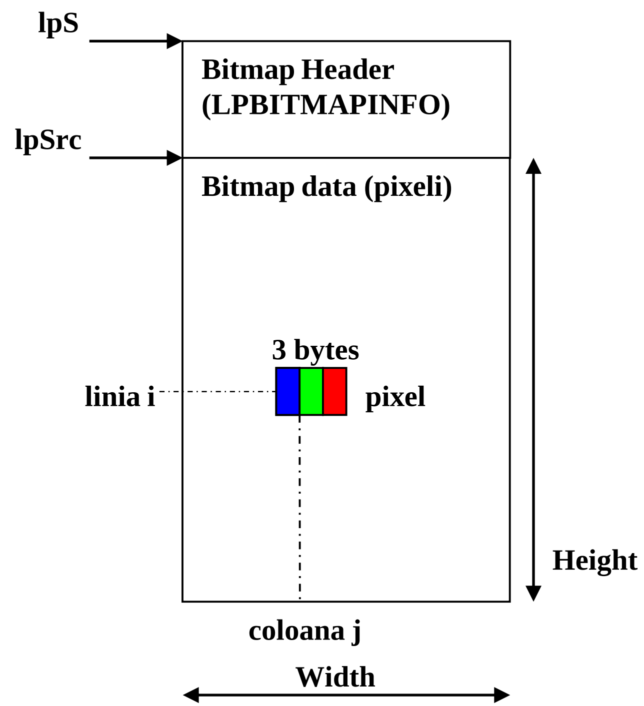
Nfx – number of pixels in x direction as sampled by the computer;
Nfy – number of pixels in frame grabber's y direction
dpx – effective X dimension of pixel in memory, dpx = dx* Ncx / Nfx;
dpy – effective Y dimension of pixel in memory, *dpy = dy* Ncy / Nfy*; ;
Ncx / Nfx – uncertainty factor for scaling horizontal scanlines;

Image formats

Spatial resolution : N_X x N_Y (Width x Height)
Color resolution/depth := number of colors encoded in a pixel
n = 1, 4, 8, 16, 24, 32 …. Bits / pixel ⇒ 2^n colors



Bitmap with LUT (1, 4, 8 bits/pixel)



RGB24 bitmap(24 biti/pixel)

Technical University of Cluj Napoca
Computer Science Department

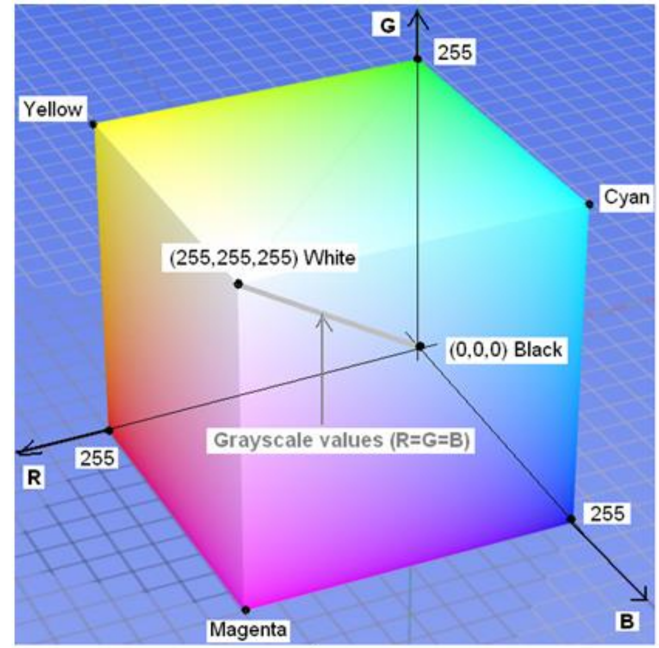
Color representation

Displays and cameras: RGB

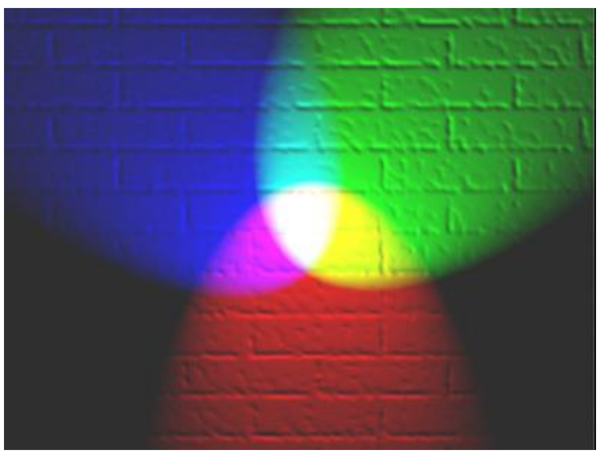
RGB ⇒ the color of each pixel is obtained by mixing the base components: (Red, Green şi Blue)

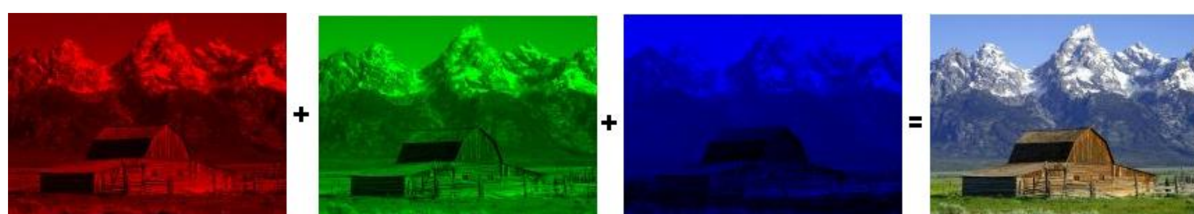
⇒ Aditive color model (R+G+B ⇒)

Grayscale / monochrome: R = G = B (diagonal of the cube)



RGB model maped in cube. Each color is encoded on 8 bitse (RGB24). Total number of colors is $2^8 x 2^8 x 2^8$ = 2^{24} = 16.777.216.

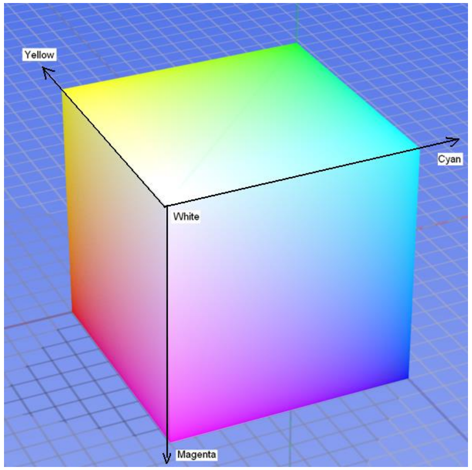




Technical University of Cluj Napoca
Computer Science Department

Color representation

Printers / plotters: CMY / CMYK



CMY: “substractive” color model

White: absence of all colors)

Black = C + M + Y

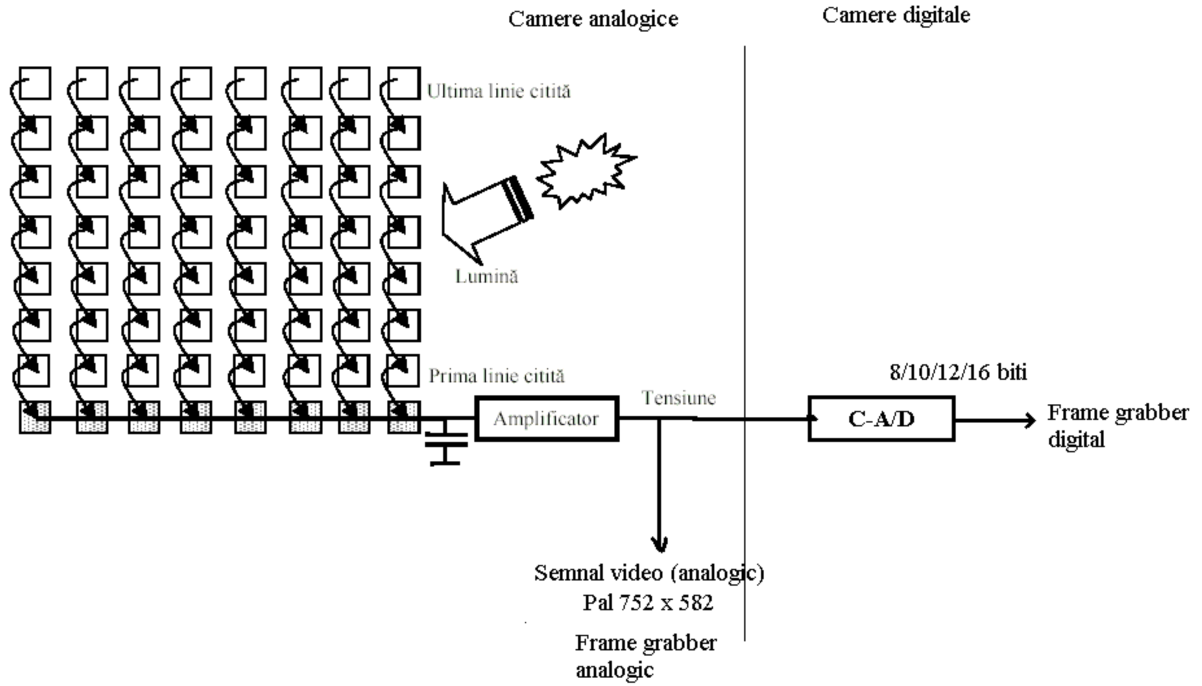
CMY

CMYK

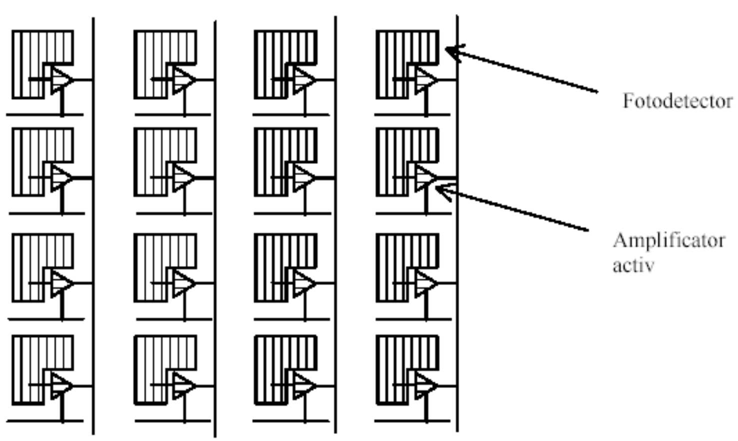
Image Acquisition and Formation

Sensor types

CCD (Charged Coupled Device)



CMOS



Sensor types

CMOS vs. CCD

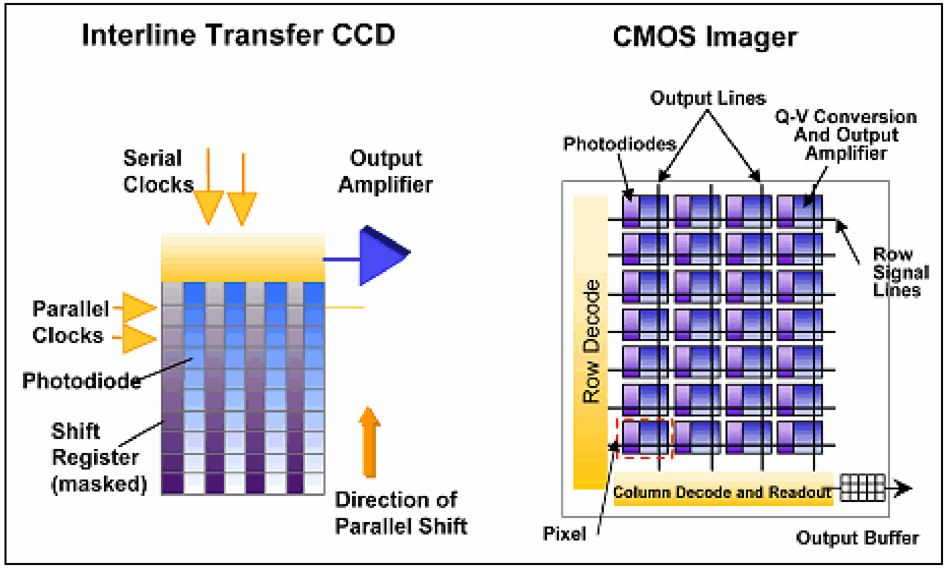


TABLE 1

Comparison of CCD and CMOS Image Sensor Features

CCD	CMOS
Smallest pixel size	Single power supply
Lowest noise	Single master clock
Lowest dark current	**Low power consumption**
~100% fill factor for full-frame CCD	**X, Y addressing and subsampling**
Established technology market base	Smallest system size
Highest sensitivity	Easy integration of circuitry
Electronic shutter without artifacts	

Image transfer

Camera ⇒ [Frame grabber] ⇒ Host computer (Memory)

Diagram 9: Equivalence between analog composite and digital video

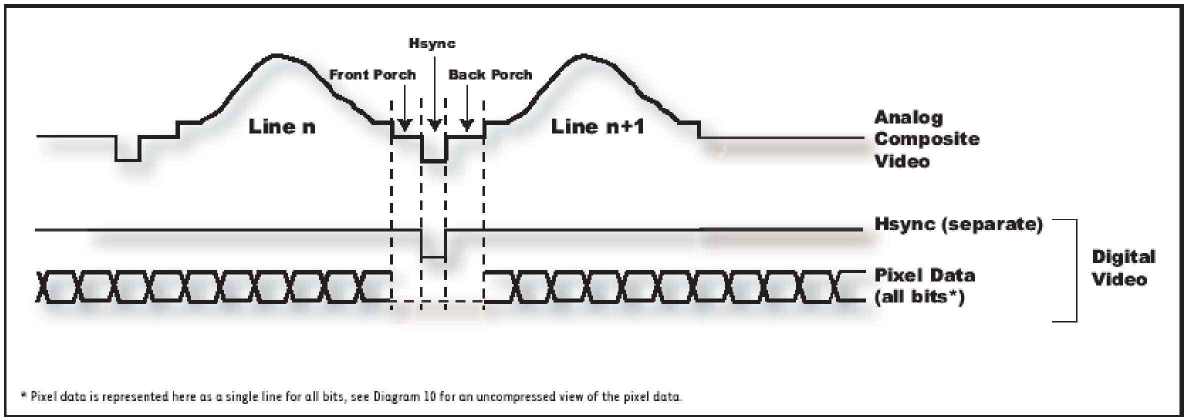
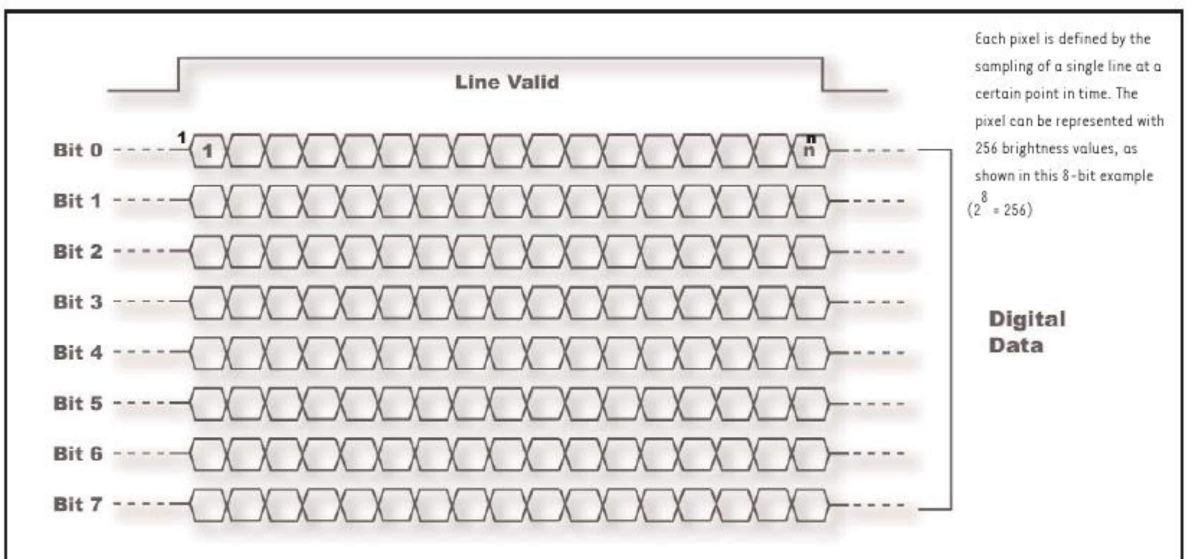


Diagram 10: 8-bit digital video.



Dedicated:

- Camera Link: 1.2Gbps (base) ... 3.6Gbps (full)
- RS 422 / EIA-644 (LVDS): 655Mbps
- IEEE 1394: 400 Mbps / 800 Mbps (Firewire)

Universal:

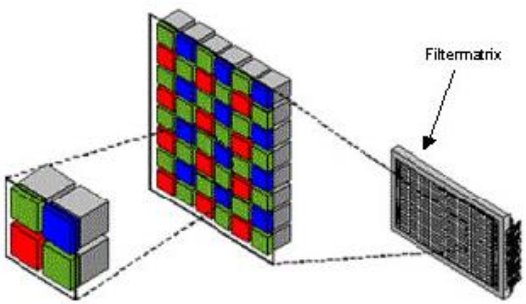
- GigaE Vision: 1Gbps, (Gigabit Ethernet protocol), low cost cables (CAT5e or CAT6), 100m distanta
- USB 3.1 "SuperSpeed+": 10Gbps
- USB 3.0 "SuperSpeed": 5Gbps
- USB 2.0: 480 Mbps
- USB 1.1 :12 mbps

Color sensors

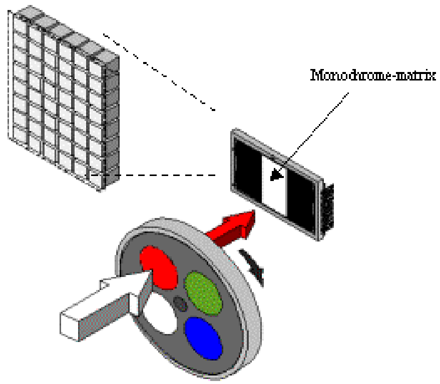
Color imagers

http://www.siliconimaging.com/RGB%20Bayer.htm
http://www.zeiss.de/c1256b5e0047ff3f/Contents-Frame/c89621c93e2600cac125706800463c66

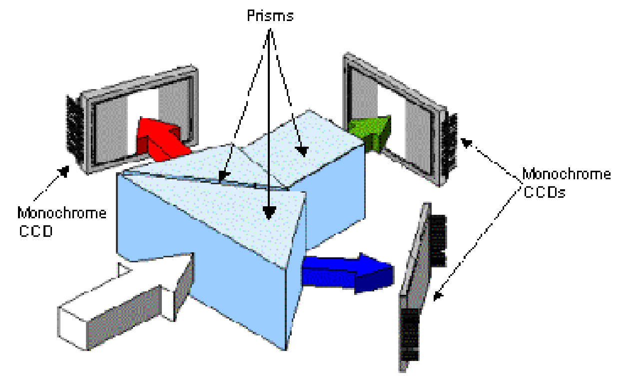
a) Bayer mask
For color photos, the majority of commercial digital color cameras use pixels covered with special color filters in the three primary colors red, green and blue.

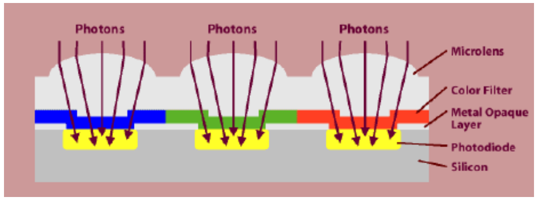


b) Filter wheel



c) 3-CCD camera





Bayer pattern decoding

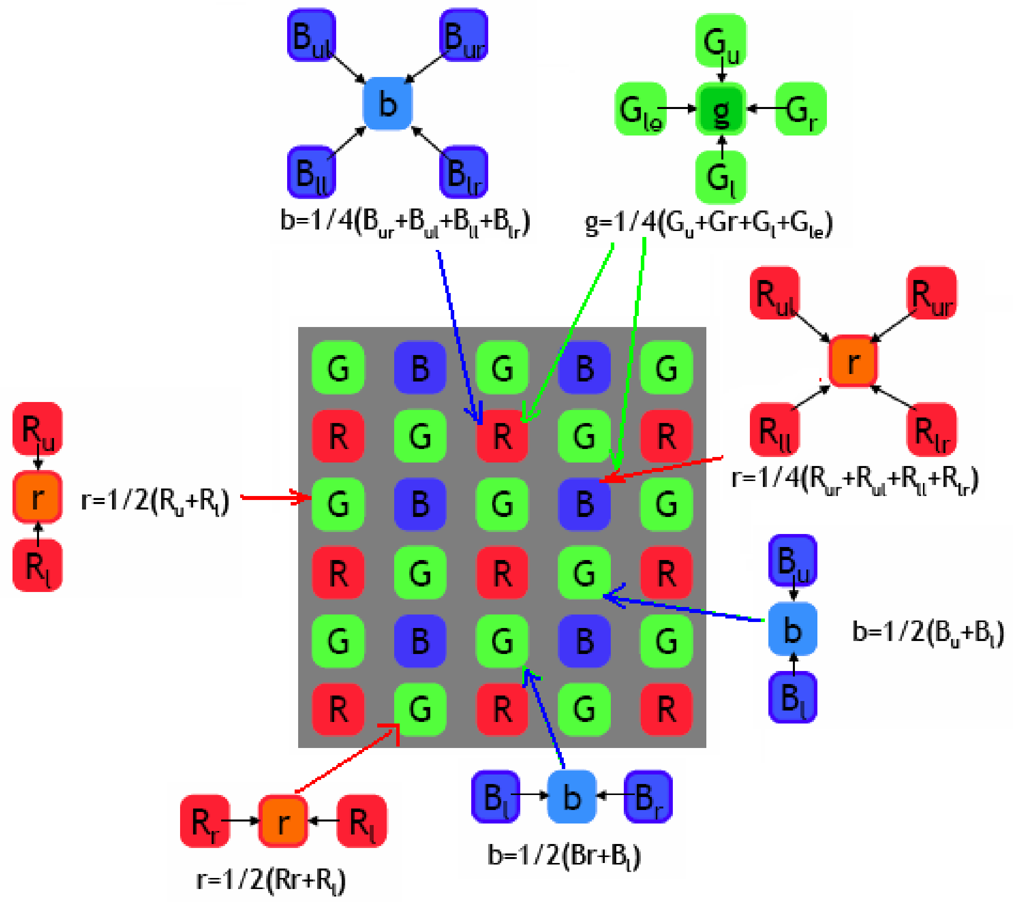


Image quality (Bayer pattern vs. 3CCD) ???

Application domains

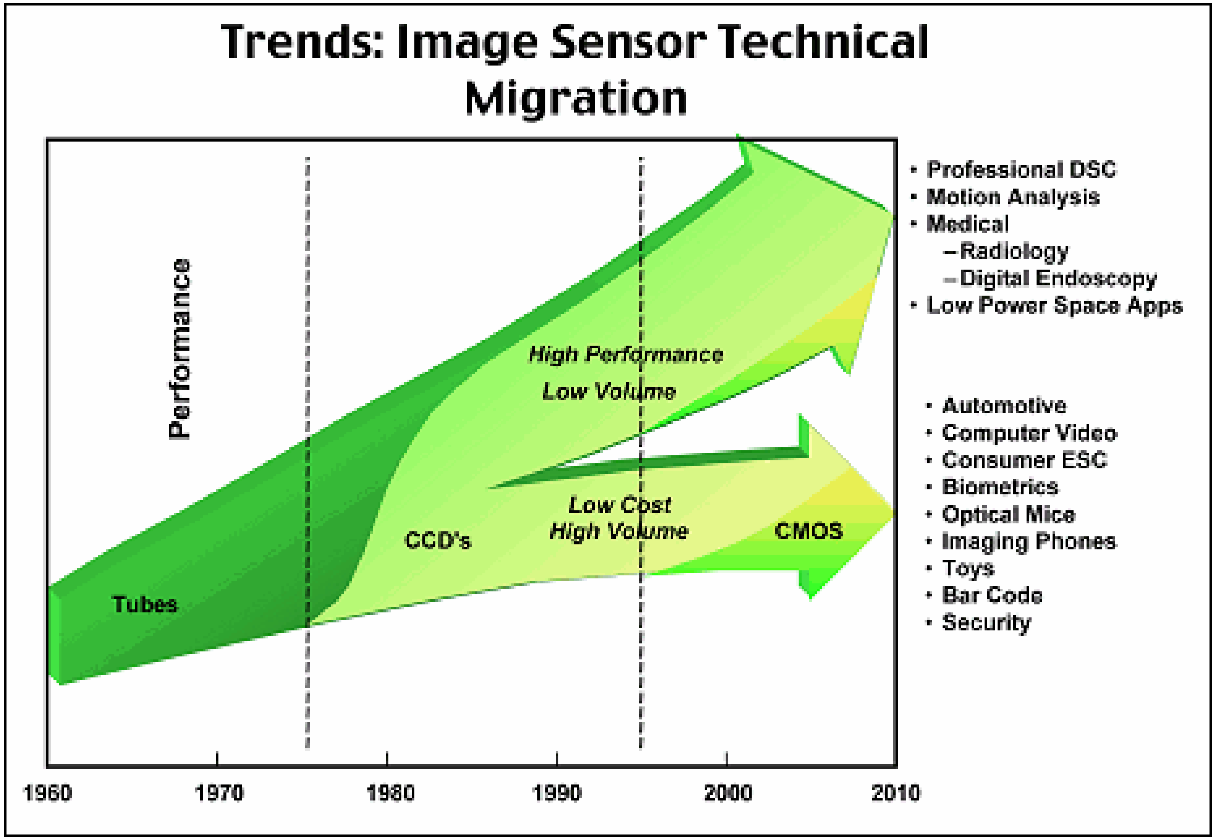


Image formation

The "thin lens" camera model [Trucco98]

The properties of the thin lens model:

1. Any ray entering the lens parallel to the optical axis on one side goes through the focus on the other side.
2. Any ray entering the lens from the focus on one side emerges parallel to the optical axis on the other size.
3. The ray going through the lens center, O, named *principal ray*, goes through point p un-deflected.

The fundamental equations of thin lenses: $Z \cdot z = f^2$

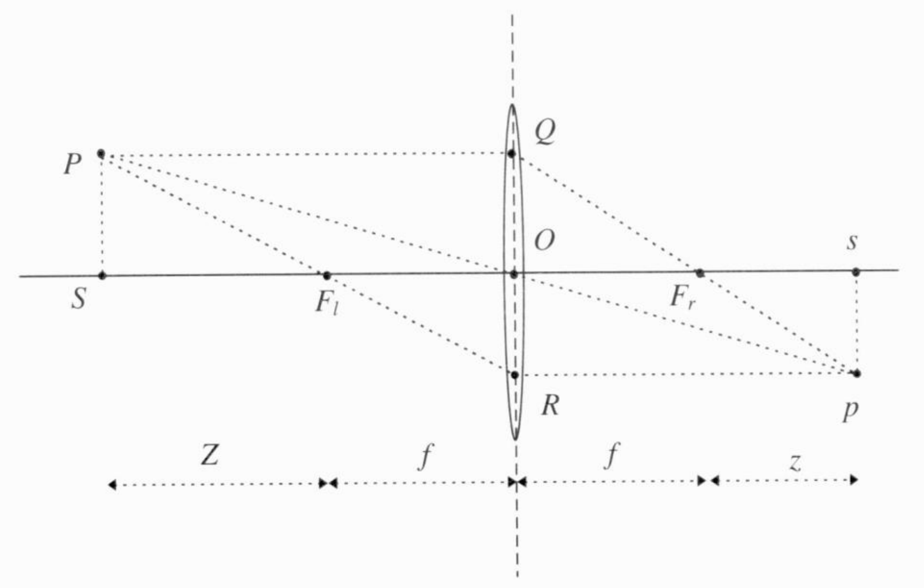
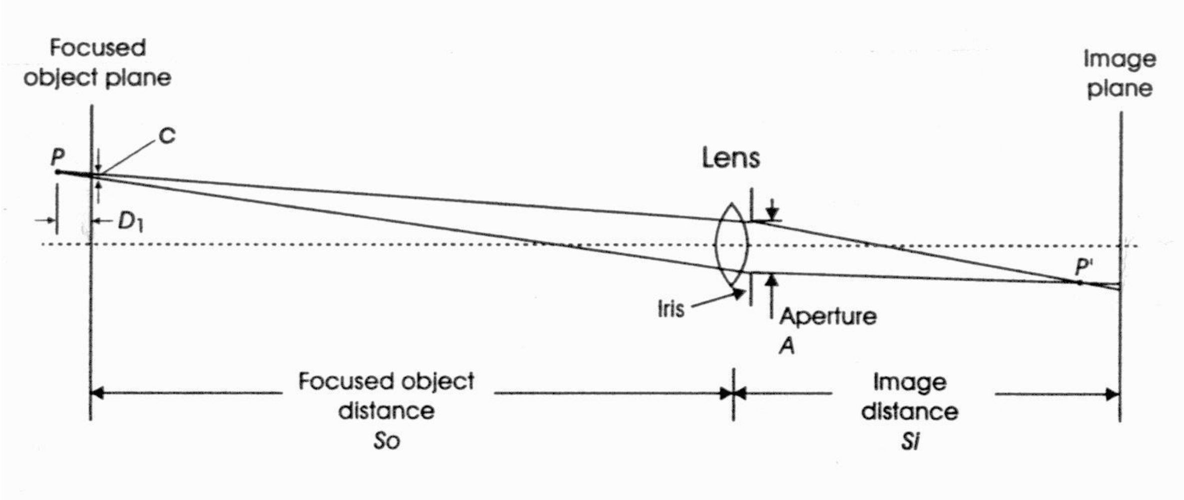


Image formation

Image focusing



Obtaining a focused image

- pinhole camera (aperture is a pont)
- optical system (lens)

Measures

- Circle-of-confusion (c) – its projection on the image plane < 1 pixel (focused image)
- Depth of field – distance (D_1) around the FOP within the (c) projection on the image < 1 pixel

Technical University of Cluj Napoca
Computer Science Department

CAMERA MODEL

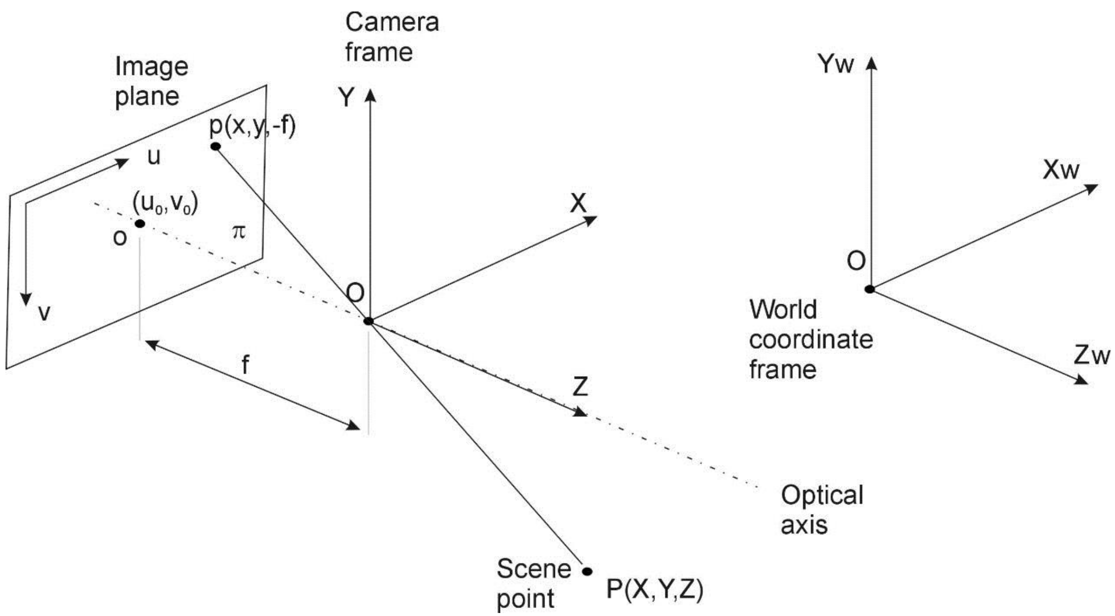
The perspective camera model (pinhole)

- The most common geometric model of an imaging camera.
- Each point in the object space is projected by a straight line through the projection center (pinhole/lens center) into the image plane.

The fundamental equations of the *perspective camera model* are [Trucco98]:

$$\begin{cases} x = f \cdot \dfrac{X_C}{Z_C} \\ y = f \cdot \dfrac{Y_C}{Z_C} \end{cases}$$

[XC, YC, ZC] are the coordinates of point P in the camera coordinate system



CAMERA MODEL

Physical camera parameters

Intrinsic parameters := internal geometrical and optical characteristics of the camera (those that specify the camera itself).

- Focal length := the distance between the optical center of the lens and the image plane: *f* [mm] or [pixels].
- Effective pixel size (dpx, dpy) [mm];
- Principal point := location of the image center in pixel coordinates: (u0,v0)
- Distortion coefficients of the lens: radial (k1, k2) and tangential (p1, p2).

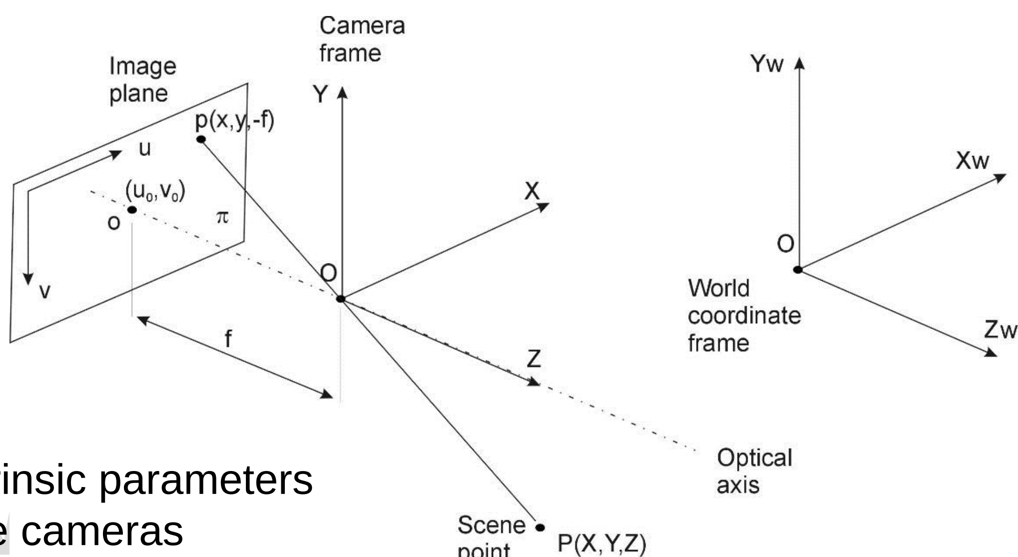
Extrinsic parameters := the 3-D position and orientation of the camera frame relative to a certain world coordinate system:

- Rotation vector $\mathbf{r} = [\, Rx, Ry, Rz \,]^T$ or

  its equivalent rotation matrix R

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

- Translation vector $\mathbf{T} = [\, Tx, Ty, Tz \,]^T$;



In multi-camera (stereo) systems, the extrinsic parameters also describe the relationship between the cameras

Camera frame ↔ image plane transformation

Camera frame ⇒ image plane transformation

(*projection / normalization*) : P = $[X_C, Y_C, Z_C]^T$ [metric units] ⇒ p = $[u, v]^T$ [pixels]

1. Transform P = $[X_C, Y_C, Z_C]^T$ ⇒ p = $[x, y, -f]^T$

Fundamental equations of the *perspective camera model* normalized with cu *1/Z*:

$$\begin{bmatrix} x \\ y \end{bmatrix} = f \cdot \begin{bmatrix} X_C / Z_C \\ Y_C / Z_C \end{bmatrix} = f \cdot \begin{bmatrix} x_N \\ y_N \end{bmatrix}$$

f – focal distance [metric units]

2. Transform **p** $[x, y]^T$ [metric units] ⇒ image coordinates $[u, v]^T$ [pixels]

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} D_u \cdot x \\ D_v \cdot y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

Du, Dv - coefficients needed to transform metric units to pixels: Du = 1 / dpx; Dv = 1 / dpy

1 + 2 ⇒ projection equation: $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} x_N \\ y_N \\ 1 \end{bmatrix}$

A – is the camera matrix:

$$A = \begin{bmatrix} f_X & 0 & u_0 \\ 0 & f_Y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

f_X – is the focal distance expressed in units of horizontal pixels: $f_X = f \cdot D_u = \frac{f}{dpx}$

f_Y – is the focal distance expressed in units of vertical pixels: $f_Y = f \cdot D_v = \frac{f}{dpx}$

Camera frame ↔ image plane transformation

Image plane transformation ⇒ camera frame

(*reconstruction*) : p = [u, v]^T [pixels] ⇒ P = [X_C, Y_C, Z_C]^T [metric units]

$$\begin{bmatrix} x_N \\ y_N \\ 1 \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Notes:

With one camera we cannot measure depth (Z). We can determine only the projection equation / normalized coordinates:

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} X_C / Z_C \\ Yc / Z_C \end{bmatrix}$$

To measure the depth (Z) a stereo system (2 cameras) is needed

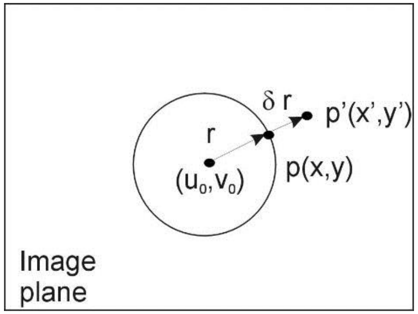
Technical University of Cluj Napoca
Computer Science Department

Modeling the lens distortions

Radial lens distortion

Causes the actual image point to be displaced radially in the image plane

$$\begin{bmatrix} \partial x^r \\ \partial y^r \end{bmatrix} = \begin{bmatrix} x \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + ...) \\ y \cdot (k_1 \cdot r^2 + k_2 \cdot r^4 + ...) \end{bmatrix}$$



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

$k_1, k_2, \ldots$ - radial distortion coefficients

Tangential distortion

Appears if the centers of curvature of the lenses' surfaces are not strictly collinear

$$\begin{bmatrix} \partial x^t \\ \partial y^t \end{bmatrix} = \begin{bmatrix} 2 p_1 \cdot xy + p_2 (r^2 + 2x^2) \\ p_1 (r^2 + 2y^2) + 2 p_2 \cdot xy \end{bmatrix}$$

p_1, p_2 – tangential distortion coefficients

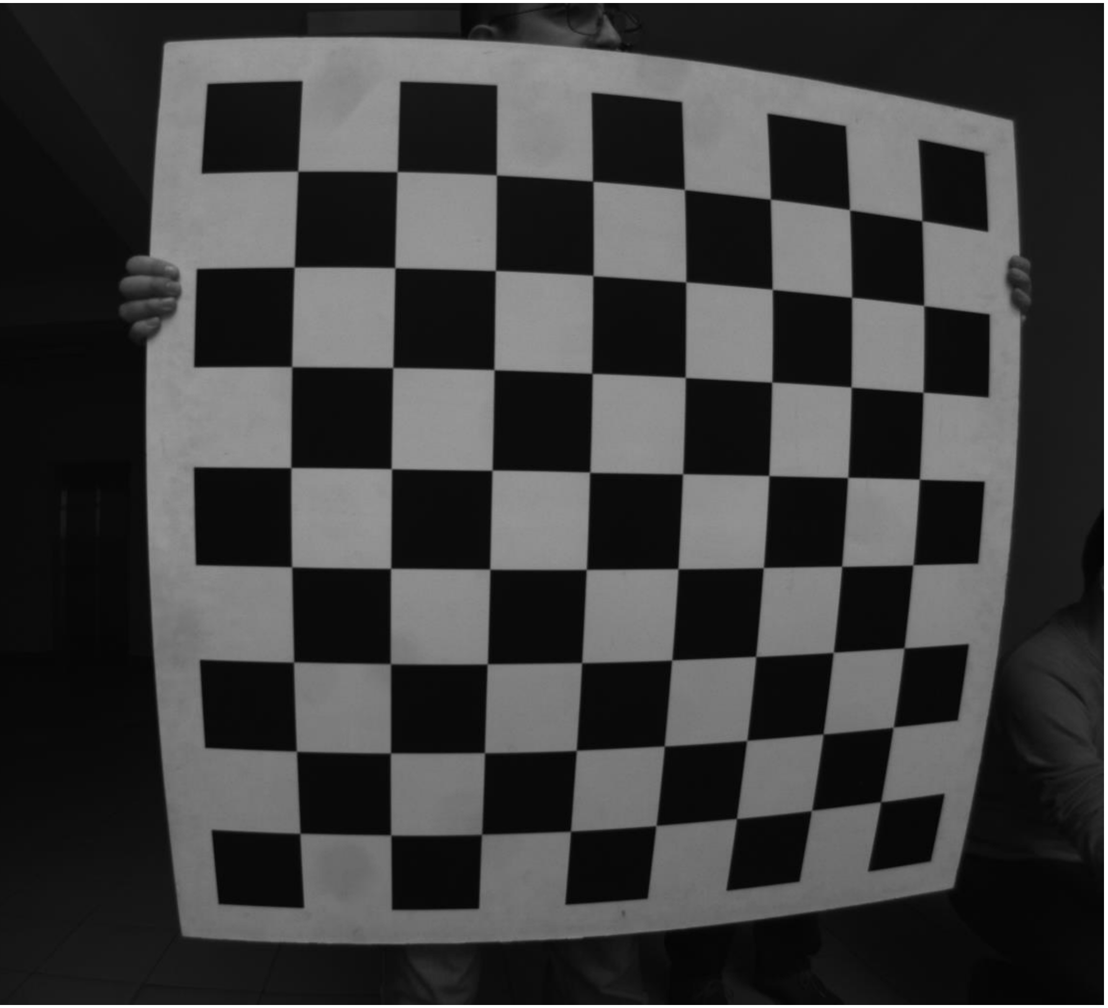
Transform **p** [x, y]T [metric units] ⇒ image coordinates [u, v]T [pixels]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} D_u \cdot (x + \partial x^r + \partial x^t) \\ D_v \cdot (y + \partial y^r + \partial y^t) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

⇒The projection equations become non-linear

Solution: perform distortion correction on image and afterwards linear projection

Modeling the lens distortions



Radial distortion for a camera with f = 4.5mm lens:

k_1= -0.22267

k_2= 0.05694

k_3= -0.00009

k_4= 0.00036

k_5= 0.00000

Distortion correction

The idea:

Between the distorted (*x', y'*) = (x+∂*x*,*y*+∂*y*) and corrected image (*x, y*) is a correspondence:

$$\begin{bmatrix} \partial x \\ \partial y \end{bmatrix} = \begin{bmatrix} \partial x^r + \partial x^t \\ \partial y^r + \partial y^t \end{bmatrix} = \begin{bmatrix} x \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) + 2 p_1 \cdot xy + p_2 (r^2 + 2x^2) \\ y \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) + p_1 (r^2 + 2y^2) + 2 p_2 \cdot xy \end{bmatrix}$$

Correction algorithm:

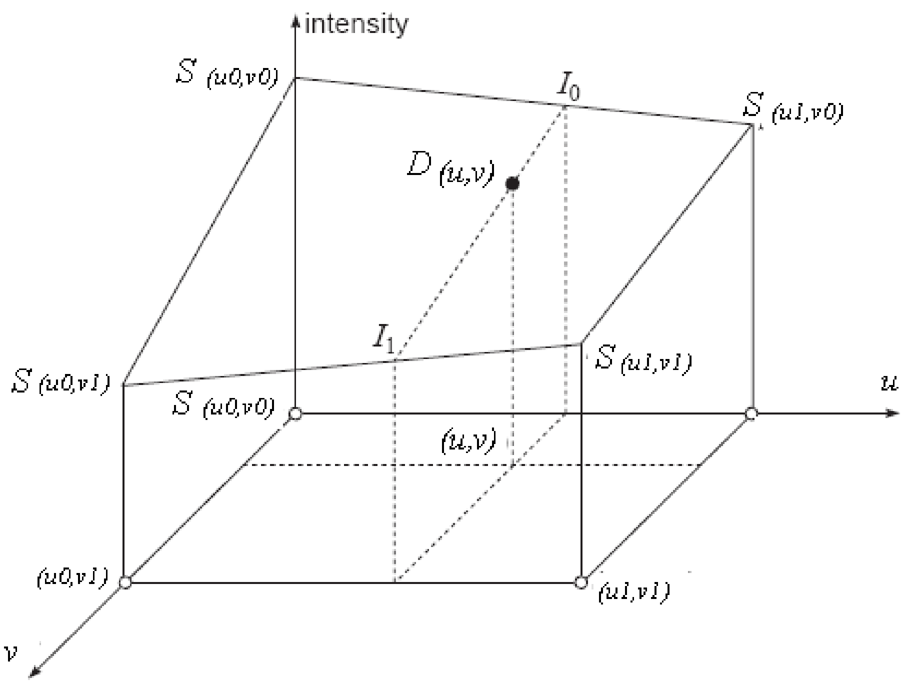
For each pixel at the integer location (u,v) in the destination (corrected) image **D**:

1. Compute the (x, y) coordinates în the camera reference system:
$$\begin{cases} x = \dfrac{u - u_0}{f_X} \\ y = \dfrac{v - v_0}{f_y} \end{cases}$$
2. Compute the distorted coordinates in the camera reference system: (*x', y'*) = (*x*+∂*x*,*y*+∂*y*)
3. For the distorted coordinate (*x', y'*) compute their image coordinates *(u', v'):*
$$\begin{cases} u' = u_0 + x' \cdot f_X \\ v' = v_0 + y' \cdot f_y \end{cases}$$
4. Assign to pixel location (u, v) from the destination image **D** the interpolated value from the source image **S** at location *(u', v'):*

$$D(u,v) = S(u',v')$$

Distortion correction

Bilinear interpolation:



$u_0 = int(u');$

$v_0 = int(v');$

$u_1 = u_0 + 1;$

$v_1 = v_0 + 1;$

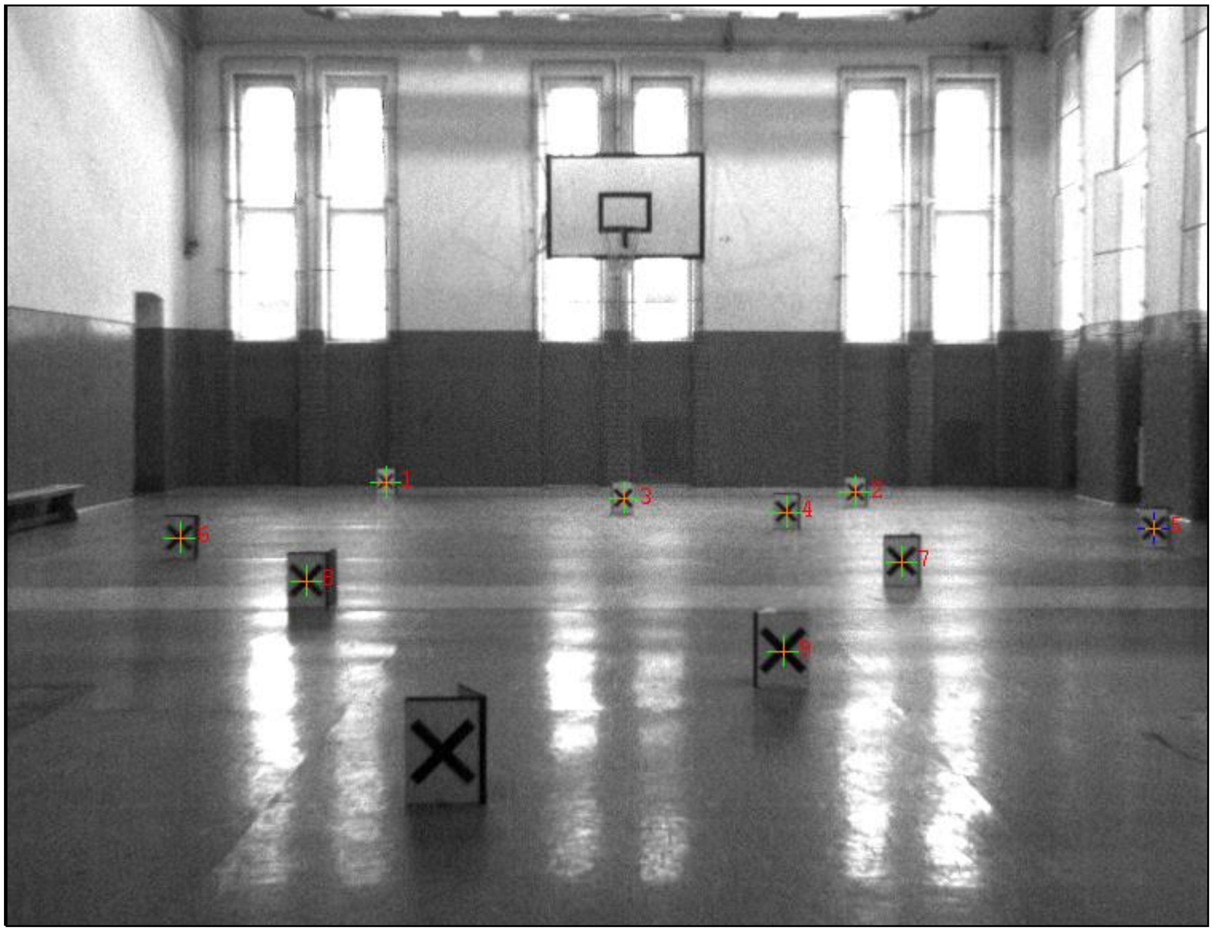
$I_0 = S(u_0,v_0))*(u_1 - u')$
$\quad + S(u_0,v_1))*(u' - u_0);$

$I_1 = S(u0,v_1))*(u_1 - u')$
$\quad + S(u_1,v_1))*(u' - u_0);$

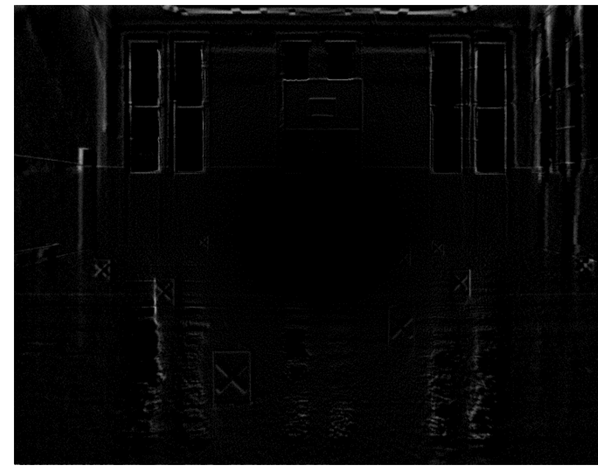
$D(u,v) = I_0 *(v_1 - v') + I_1*(v' - v_0);$

Lenses distortion correction

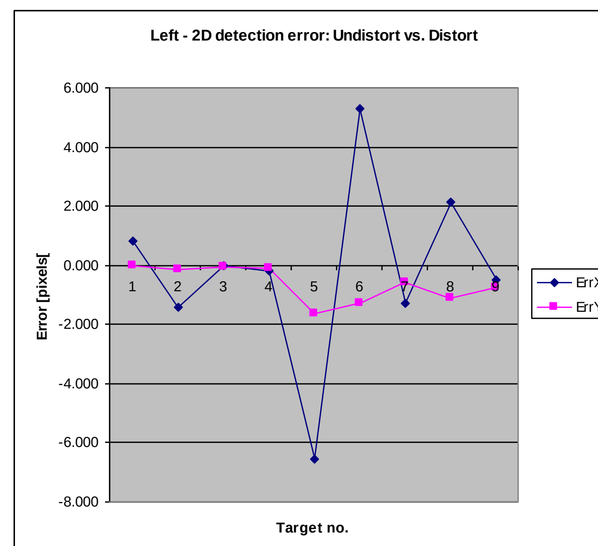
8.5 mm lens, CCD camera



Undistorted image



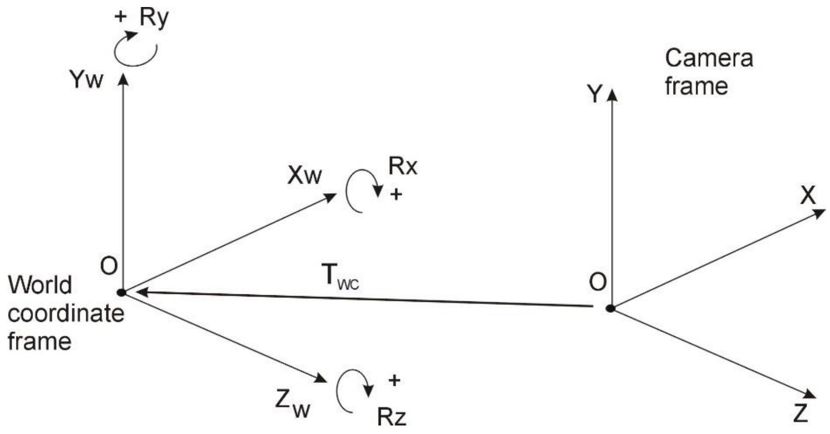
Difference image



Camera frame ↔ world reference frame transformation

Direct mapping (world ⇒ camera)

$\mathbf{XX}_W = [\ X_W\ ,\ Y_W\ ,\ Z_W\ ]^T$ (world coordinate system - WRF) ⇒ $\mathbf{XX}_C = [X_C\ ,\ Y_C\ ,\ Z_C]^T$ (camera coordinate system – CRF)



$$\mathbf{XX}_C = \mathbf{R}_{WC} \cdot \mathbf{XX}_W + \mathbf{T}_{WC}$$

where:

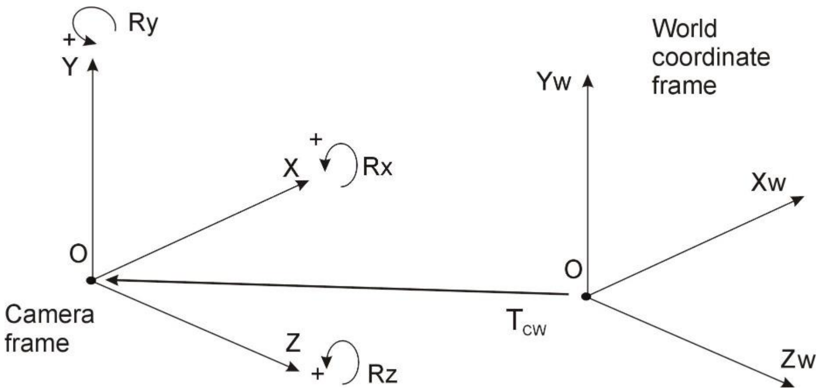
$\mathbf{T}_{WC} = [\ Tx, Ty, Tz\ ]^T$ – world to camera translation vector;
$\mathbf{R}_{WC}$ – world to camera rotation matrix:

Camera frame ↔ world reference frame transformation

Inverse mapping (camera ⇒ world)

$\mathbf{XX}_C = [X_C, Y_C, Z_C]^T$ (camera coordinate system – CRF) ⇒ $\mathbf{XX}_W = [X_W, Y_W, Z_W]^T$ (world coordinate system - WRF)

$$\mathbf{XX}_W = \mathbf{R}_{WC}^{-1} \cdot (\mathbf{XX}_C - \mathbf{T}_{WC})$$



Rotation matrix is orthogonal [Trucco1998]:

$$\mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = 1 \Rightarrow \mathbf{R}^T = \mathbf{R}^{-1}$$

$$\mathbf{XX}_W = \mathbf{R}_{WC}^T \cdot (\mathbf{XX}_C - \mathbf{T}_{WC}) = \mathbf{R}_{CW} \cdot (\mathbf{XX}_C + \mathbf{T}_{CW})$$

where:

$\mathbf{T}_{CW} = [T_X\ T_Y\ T_Z]^T$ – camera to world translation vector $\quad T_{CW} = -T_{WC}$

$\mathbf{R}_{CW}$ – camera to world translation vector $\quad R_{CW} = R_{WC}^T$

Technical University of Cluj Napoca
Computer Science Department

Rotation Matrix

World-to-camera

$$\mathbf{R}_{WC} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{n}^{XW} & \mathbf{n}^{YW} & \mathbf{n}^{ZW} \end{bmatrix} = \begin{bmatrix} n_X^{XW} & n_X^{YW} & n_X^{ZW} \\ n_Y^{XW} & n_Y^{YW} & n_Y^{YW} \\ n_Z^{XW} & n_Z^{YW} & n_Z^{YW} \end{bmatrix}$$

$\mathbf{n}^{XW} = \begin{bmatrix} n_X^{XW} & n_Y^{XW} & n_Z^{XW} \end{bmatrix}^T$ – normal vector of **OX**$_W$ axis in the CRF

$\mathbf{n}^{YW} = \begin{bmatrix} n_X^{YW} & n_Y^{YW} & n_Z^{YW} \end{bmatrix}^T$ – normal vector of **OY**$_W$ axis in the CRF

$\mathbf{n}^{ZW} = \begin{bmatrix} n_X^{ZW} & n_Y^{ZW} & n_Z^{ZW} \end{bmatrix}^T$ – normal vector of **OZ**$_W$ axis in the CRF

Camera-to-world

$$\mathbf{R}_{CW} = \mathbf{R}_{WC}^{\ T} = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{n}^{XC} & \mathbf{n}^{YC} & \mathbf{n}^{ZC} \end{bmatrix} = \begin{bmatrix} n_X^{XC} & n_X^{YC} & n_X^{ZC} \\ n_Y^{XC} & n_Y^{YC} & n_Y^{YC} \\ n_Z^{XC} & n_Z^{YC} & n_Z^{YC} \end{bmatrix}$$

$\mathbf{n}^{XC} = \begin{bmatrix} n_X^{XC} & n_Y^{XC} & n_Z^{XC} \end{bmatrix}^T$ – normal vector of **OX**$_C$ axis in the WRF

$\mathbf{n}^{YC} = \begin{bmatrix} n_X^{YC} & n_Y^{YC} & n_Z^{YC} \end{bmatrix}^{YT}$ – normal vector of **OY**$_C$ axis in the WRF

$\mathbf{n}^{ZC} = \begin{bmatrix} n_X^{ZC} & n_Y^{ZC} & n_Z^{ZC} \end{bmatrix}^T$ – normal vector of **OZ**$_C$ axis in the WRF

Rotation Matrix ↔ Rotation Vector

Rotation vector

$\mathbf{r}_{WC} = [R_X\ R_Y\ R_Z]^T$ (R_X - pitch, R_Y - yaw, R_Z - tilt / roll)

$r_{WC} \Rightarrow R_{WC}$ transform:

$$r_{11} = \cos(R_Y)\cos(R_Z)$$
$$r_{12} = \sin(R_X)\sin(R_Y)\cos(R_Z) - \cos(R_X)\sin(R_Z)$$
$$r_{13} = \cos(R_X)\sin(R_Y)\cos(R_Z) + \sin(R_X)\sin(R_Z)$$
$$r_{21} = \cos(R_Y)\sin(R_Z)$$
$$r_{22} = \sin(R_X)\sin(R_Y)\sin(R_Z) + \cos(R_X)\cos(R_Z)$$
$$r_{23} = \cos(R_X)\sin(R_Y)\sin(R_Z) - \sin(R_X)\cos(R_Z)$$
$$r_{31} = -\sin(R_Y)$$
$$r_{32} = \sin(R_X)\cos(R_Y)$$
$$r_{33} = \cos(R_X)\cos(R_Y)$$

$R_{WC} \Rightarrow r_{WC}$ transform:

$$R_Y = \arcsin(r_{31})$$

If $\cos(R_Y) \neq 0$:

$$R_X = \text{atan2}\left(-\frac{r_{32}}{\cos(R_Y)}, \frac{r_{33}}{\cos(R_Y)}\right)$$
$$R_Z = -\text{atan2}\left(-\frac{r_{21}}{\cos(R_Y)}, \frac{r_{11}}{\cos(R_Y)}\right)$$

If $\cos(R_Y) = 0$:

$$R_X = \text{atan2}\left(r_{12}, r_{22}\right)$$

$$R_Z = 0$$

3D (world) ⇒ 2D (image) mapping using the Projection Matrix

Projection matrix

$$\mathbf{P} = \mathbf{A} \cdot \left[\mathbf{R}_{WC} \mid \mathbf{T}_{WC}\right]$$

The projection equation of a 3D world point [X_W , Y_W , Z_W]:

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \mathbf{P} \cdot \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

$s=z_S$ – scaling factor

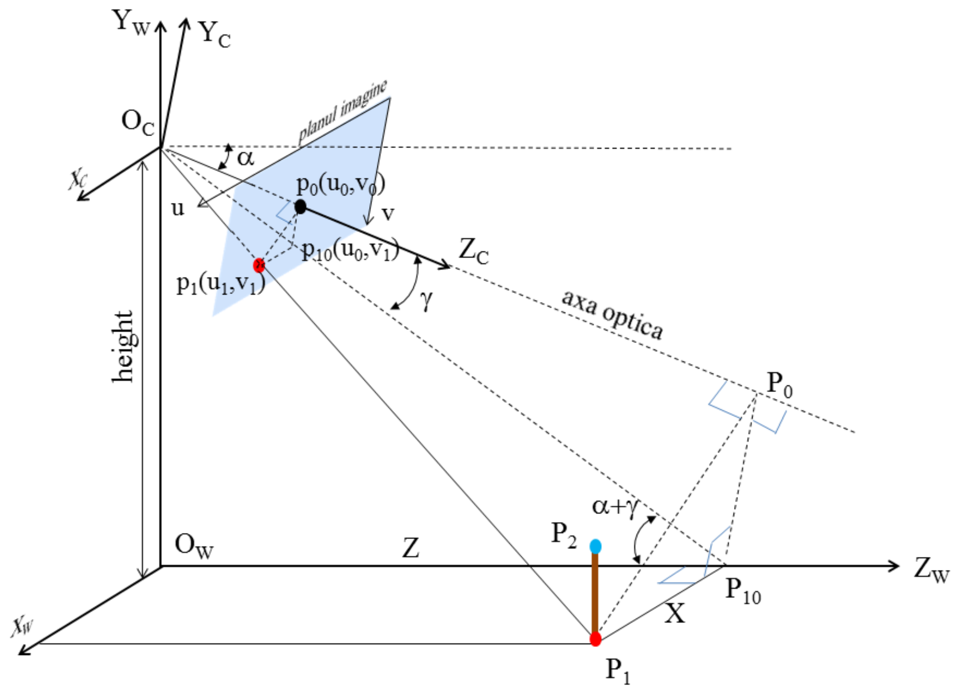
Obtaining the 2D image coordinates

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_S / z_S \\ y_S / z_S \end{bmatrix}$$

2D (image) ⇒ 3D (world) mapping (monocular vision)

Can be done only in a simplified scenario:

- O_W (WRF origin) is the ground projection of the O_C (CRF origin)
- $O_W Z_W$ and $O_C Z_C$ coplanar
- *height* si pitch (α) are fixed and known (ex: fixed surveillance camera fith fixed folcal length f)



Input data

Intrinsic camera parameters:

f_X , f_Y = focal distances [pixels]

$p_0(u_0, v_0)$ = principal point [pixels]

Extrinsic camera parameters:

α = camera pitch

height = relative to the ground

$p_1(u_1,v_1)$ = image projection of $P_1(X,0,Z)$

$p_2(u_2,v_2)$ = image projection of $P_2(X,Y,Z)$

Output data

3D coordinates of P_1 and P_2 in WRF

2D (image) ⇒ 3D (world) mapping (monocular vision)

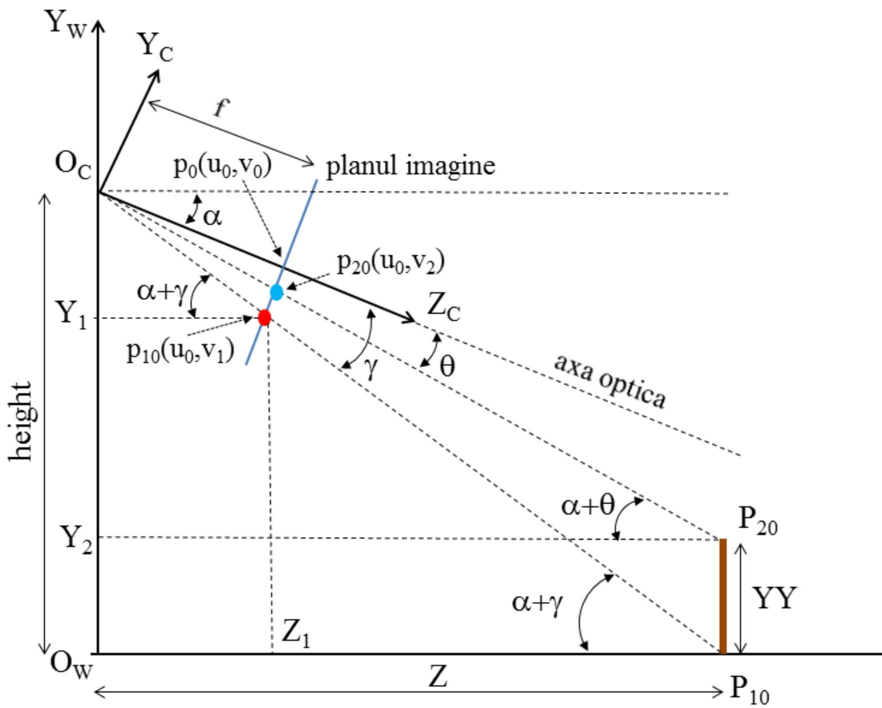
1. Side-view projection on $Y_W O_W Z_W$ plane ($X_W = 0$) (fig. de mai jos). P_1 is projected in P_{10}, p_1 in p_{10}, P_2 in P_{20} etc.

The angle between the $[O_C P_{10}]$ segment with the optical axis of the camera is obtained from triangle $\Delta O_C p_{10} p_0$:

$$\gamma = \tan^{-1}\left(\frac{v_1 - v_0}{f_Y}\right)$$

The depth Z of P_1 (respective P_{10}) in WRF is deduced from triangle $\Delta O_C O_W P_{10}$:

$$Z = [O_W P_{10}] = \frac{height}{\tan(\alpha + \gamma)}$$



Proiectia laterala (pe planul $Y_W O_W Z_W$) a scenei

From triangle $\Delta O_C p_{20} p_0$ is deduced deduced:

$$\theta = \tan^{-1}\left(\frac{v_2 - v_0}{f_Y}\right)$$

From triangle $\Delta O_C Y_2 P_{20}$ the is deduced:

$$YY = [O_C O_W] - [O_C Y_2] = height - Z \cdot \tan(\alpha + \theta)$$

Technical University of Cluj Napoca
Computer Science Department

2D (image) ⇒ 3D (world) mapping (monocular vision)

2. The lateral offset X of P_1 relative to O_WZ_W axis is deduced from the *top view / bird-eye view projections* of the scene on the horizontal $X_WO_WZ_W$ plane:

From the similarity of triangles $\Delta O_W p_{10} p_1$ and $\Delta O_W P_{10} P_1$:

$$X = [P_1 P_{10}] = [p_1 p_{10}] \cdot \frac{[O_W P_{10}]}{[O_W Z_1]} = \frac{(u_1 - u_0)}{[O_W Z_1]} \cdot Z$$

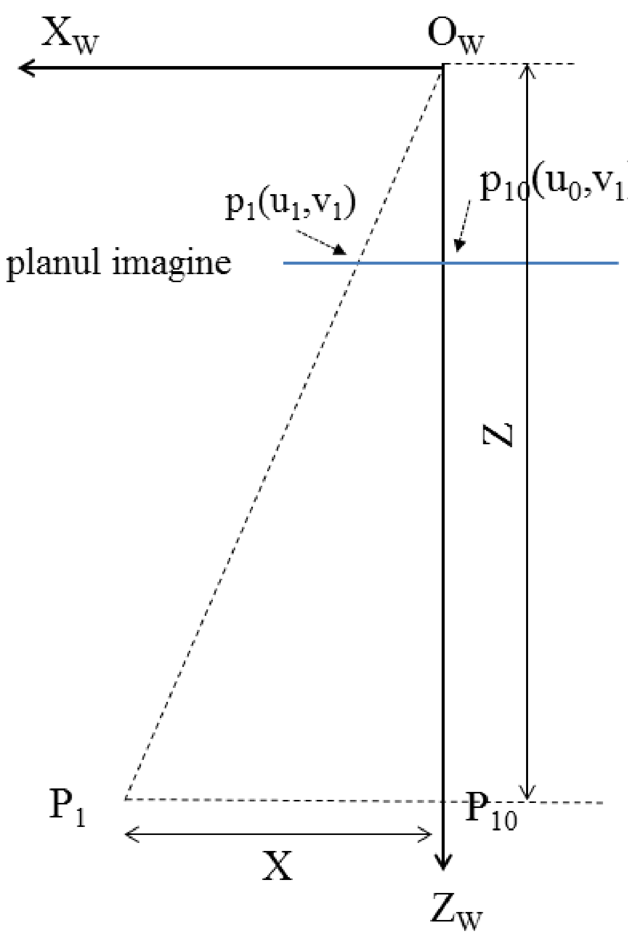
$[O_W Z_1]$ (where Z_1 is the projection of p_{10}) is deduced from the side-view projection:

$$[O_W Z_1] = [Y_1 p_{10}] = [O_C p_{10}] \cdot \cos(\alpha + \gamma) = \frac{f_Y}{\cos(\gamma)} \cdot \cos(\alpha + \gamma)$$

From the above 2 eq.

$$X = \frac{(u_1 - u_0)}{f_Y} \cdot \frac{Z}{\cos(\alpha + \gamma)} \cdot \cos(\gamma)$$

Note: fx=fy=f [pixeli] !



Proiectia de sus / top-view (pe planul $X_WO_WZ_W$)

Goal

The fundamental equations of the *pinhole camera model* are [Trucco98]:

$$\begin{cases} x = f \cdot \dfrac{X_C}{Z_C} \\ y = f \cdot \dfrac{Y_C}{Z_C} \end{cases}$$

$P(X_C,Y_C,Z_C)$ 3D point in the camera coordinate system
p(x,y,-f) its projection on the image plane

Knowing the image coordinates (x,y) we cannot infer the depth (Z), only the projection equations

Measure depth (Z) $\Rightarrow$ at least two cameras (stereo-system)

Stereo camera configurations

- Canonic (parallel axes) – theoretical model (impossible to obtain in practice) $\Rightarrow$ image rectification
- Coplanar axes (but unparallel)
- General configuration

Technical University of Cluj Napoca
Computer Science Department

STEREOVISION

Basics of epipolar geometry

- Epipole (baseline intersection with the image plane) $\Rightarrow$ one epipole / each image (is the intersection of all epipolar lines of the image)
- Epipolar plane $\Rightarrow$ one plane for each 3D point
- Epipolar line $\Rightarrow$ one line for each 3D point

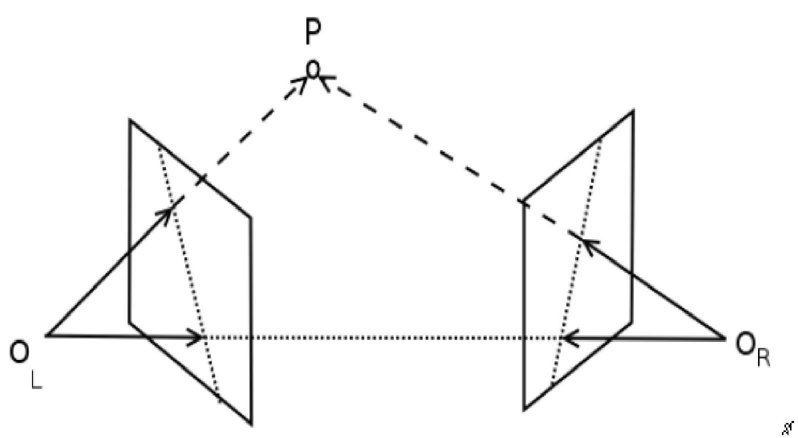
Epipolar geometry constraint

- For every image point (x_L, y_L) there is a unique epipolar line (e_R) in the right image which will contaun the corresponding right projection point (y_R,y_R) and vice-versa

Computing (e) [Trucco] :

$$a * x + b * y + c = 0$$

$$\begin{vmatrix} a_R \\ b_R \\ c_R \end{vmatrix} = \mathbf{F} * \mathbf{P}_L \quad ; \quad \begin{vmatrix} a_L \\ b_L \\ c_L \end{vmatrix} = \mathbf{F}^T * \mathbf{P}_R$$



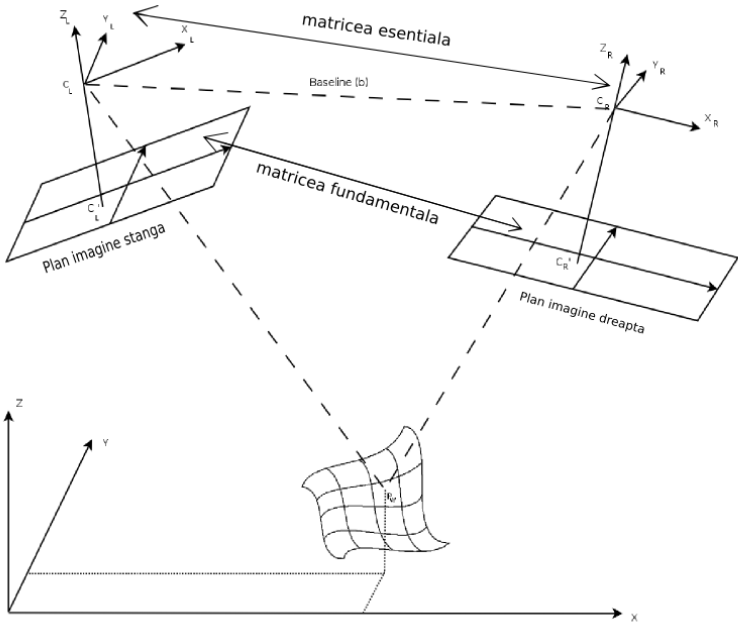
where:

F – fundamental matrix $\quad \mathbf{P}_L = \begin{vmatrix} x_L \\ y_L \\ 1 \end{vmatrix} ; \quad \mathbf{P}_R = \begin{vmatrix} x_R \\ y_R \\ 1 \end{vmatrix}$

STEREOVISION

Computing the fundamental (F) and essential (E) matrices

$$\mathbf{F} = \left(\mathbf{A}_R^{-1}\right)^T * \mathbf{E} * \mathbf{A}_L^{-1}$$



where:

$$\mathbf{E} = \mathbf{R}_{LR} * \mathbf{S}$$

$$\mathbf{R}_{LR} = \mathbf{R}_{CW-R}^T * \mathbf{R}_{CW-L}$$

$$\mathbf{T}_{LR} = \mathbf{T}_{CW-R} - \mathbf{T}_{CW-L} = \begin{bmatrix} T_X & T_Y & T_Z \end{bmatrix}^T$$

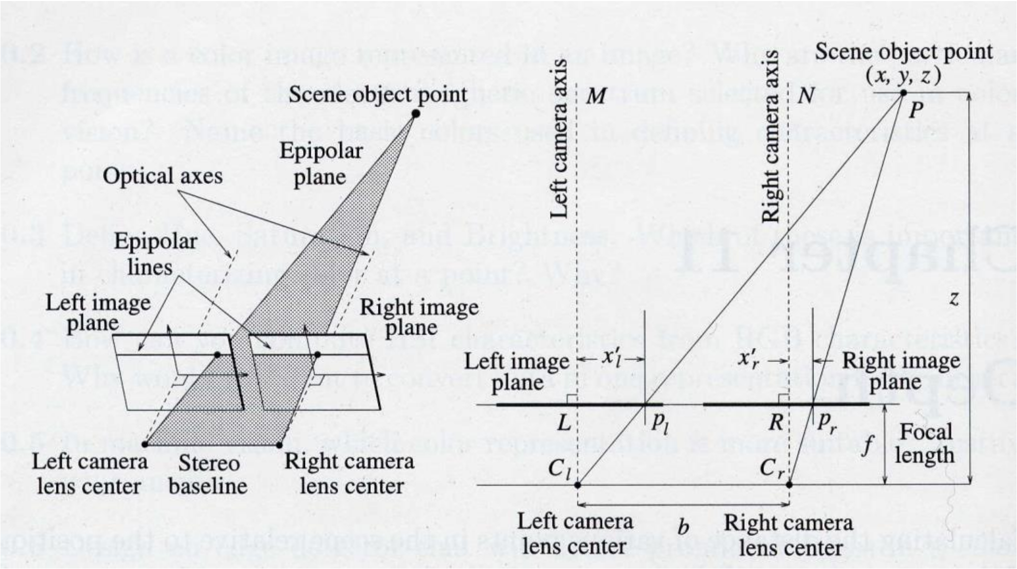
$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

E – essential matrix

$\mathbf{R}_{LR}$ – relative left-to-right rotation matrix

$\mathbf{T}_{LR}$ – relative left-to-right rotation matrix

The canonical model



Assumptions

- Image planes are coplanar ⇒ optical axes are parallel
- Horizontal image axes are collinear
- Epipolar lines – horizontal
- $v_{0L} = v_{0R} \Rightarrow y_L = y_R$

Depth estimation (canonical)

$$x_l' = f_X \cdot \frac{X_1}{Z_1}$$

$$x_r' = f_X \cdot \frac{X_2}{Z_2}$$

$$d = x_l' - x_r' = f_X \cdot \left(\frac{X_1}{Z_1} - \frac{X_2}{Z_2}\right) = f_X \cdot \frac{X_1 - X_2}{Z} = f_X \frac{b}{Z}$$

$$Z = \frac{f_X \cdot b}{d}$$

Depth estimation (coplanar)

Coplanar but non-parallel optical axes: θ angle

$$Z = \frac{f_X \cdot b}{d + f_X \cdot \tan(\theta)}$$

The stereo-correlation problem

For an given left image point p_L a right image point p_R must be find so that the pair (p_L, p_R) represents the projections of the same 3D point P on the two image planes.

a. *Features selection*

- low level features: pixels.
- high level features: edge segments / corners

b. *Features matching*

- use of epipolar geometry constraints (epipolar lines) for search space reduction
- for a left image point, a right correspondent point must be chosen out of a set of candidates.
- the correlation function is the measure used for discrimination.

c. *Increasing the resolution of the correlation*

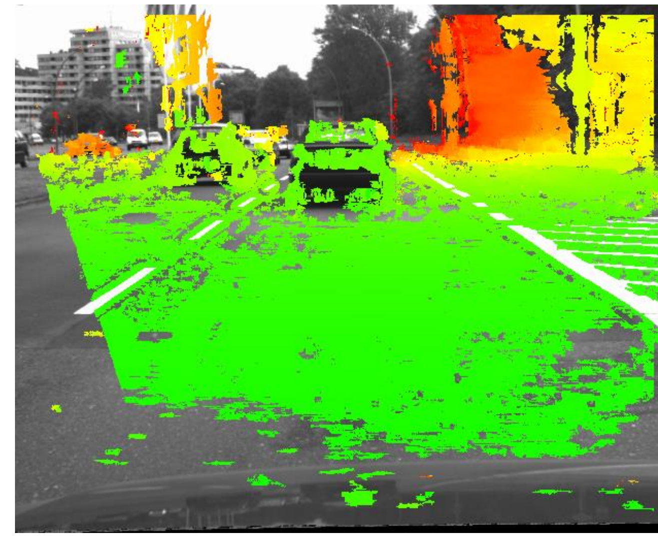
- compute the disparity with sub-pixel accuracy $\Rightarrow$ far range & high accuracy stereo-vision

Technical University of Cluj Napoca
Computer Science Department

STEREOVISION

Features selection

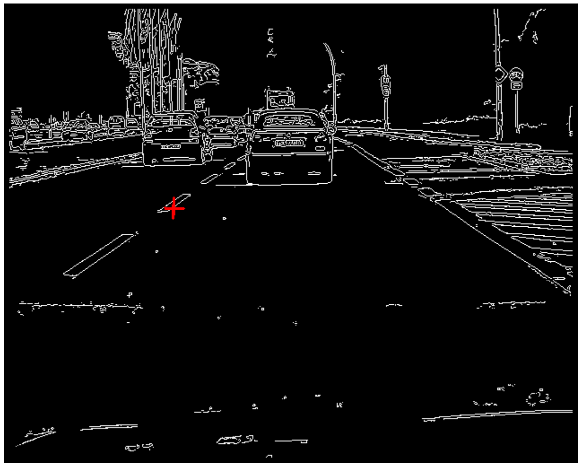
Low level features - each image pixel is reconstructed (dense stereo)



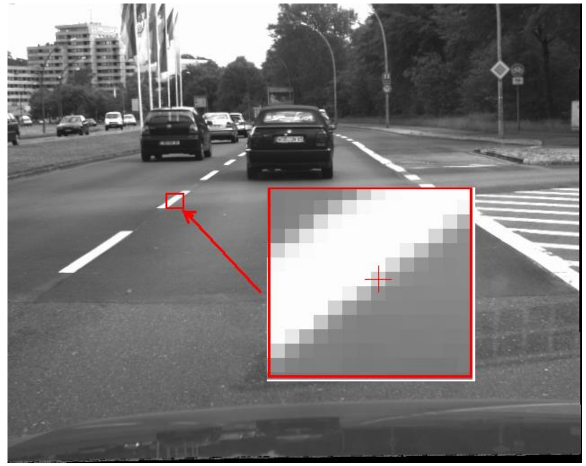
High level features – only edge pixels are reconstructed (edge based stereo)

STEREOVISION

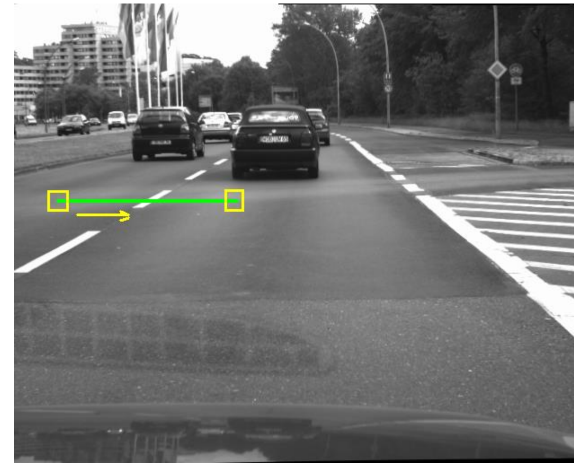
Features matching



Edge feature in left image

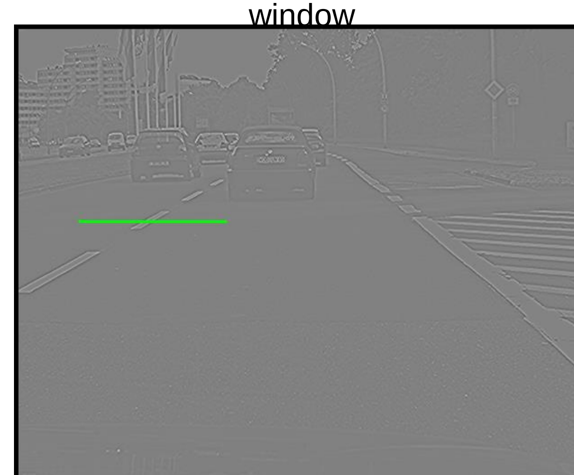


Grayscale left image correlation window



The correlation window slides along the epipolar line in the right image window

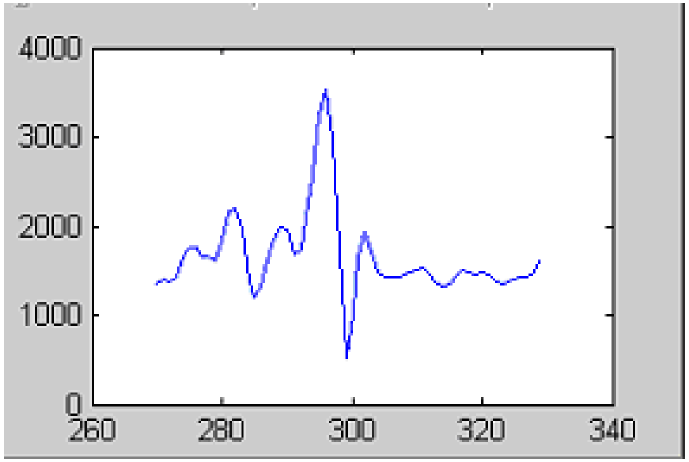
LoG left image correlation window



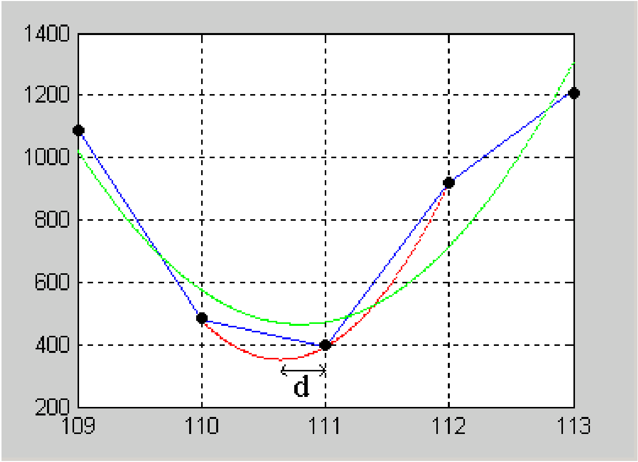
The correlation function

Any distance measure function SAD, SSD, normalized correlation

$$SAD(x_R, y_R) = \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \left| I_L(x_L + i, y_L + j) - I_R(x_R + i, y_R + j) \right|$$



Global minima of the correlation function

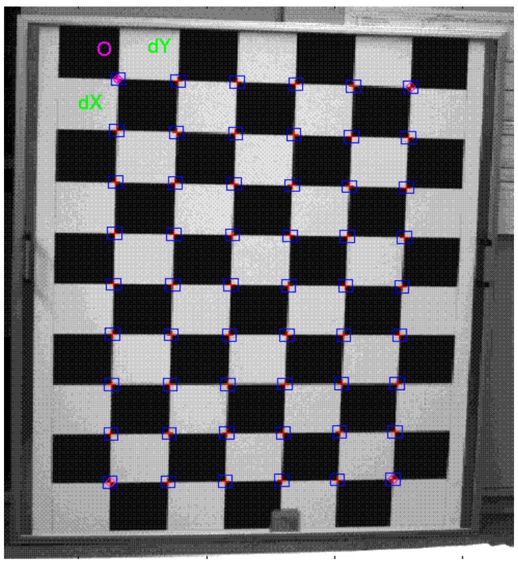


Detection of the sub-pixel position of global minima of the correlation function using a parabolic interpolator (2 points)

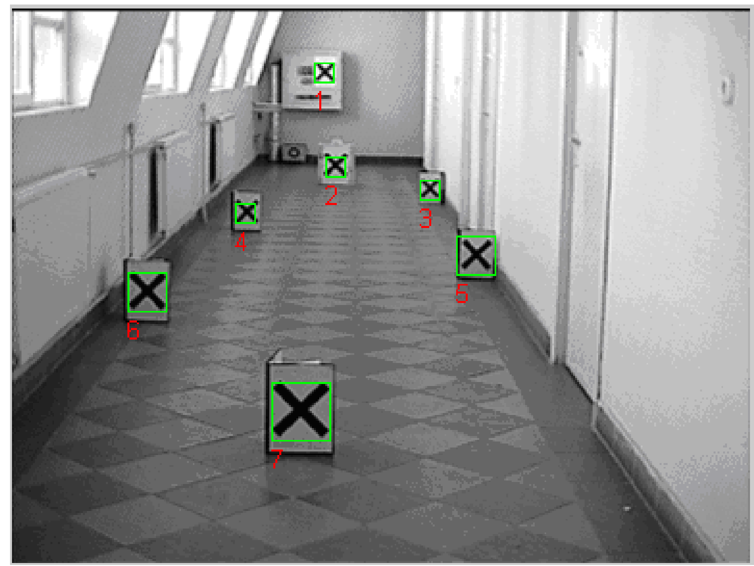
Technical University of Cluj Napoca
Computer Science Department

Camera calibration

Most of the methods are using a planar calibration object or a 3D setup in which control points can be detected in the image.



Planar pattern used for intrinsic parameters calibration



3D scenario with X-shaped targets used for extrinsic parameters calibration

Technical University of Cluj Napoca
Computer Science Department

Camera calibration

Parameters estimation: minimization process of the total error between the 2D image coordinates of the control points (detected from the images: m_i) and the image projections $\bar{m}_i$ of the 3D coordinates of the control points estimated using the camera model (intrinsic + extrinsic parameters):

$$\| m_i - \bar{m}_i \| \qquad \sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \breve{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2$$

Camera Calibration Toolbox for Matlab (J.Y. Bouguet)

- Best calibration toolbox for intrinsic parameters

http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#links

Omni-directional camera calibration

http://www-sop.inria.fr/icare/personnel/Christopher.Mei/ChristopherMeiPhDStudentToolbox.html

Technical University of Cluj Napoca
Computer Science Department