MAI4CAREU | Master programmes in Artificial Intelligence 4 Careers in Europe

Visual Computing Group

University of Cyprus – MSc Artificial Intelligence

MAI644 – COMPUTER VISION

Lecture 15: Camera Calibration

Melinos Averkiou

CYENS Centre of Excellence

University of Cyprus - Department of Computer Science

m.averkiou@cyens.org.cy

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Last time

- Perspective projection
- Vanishing points
- Full camera model

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Today’s Agenda

- Full camera model in matrix form
- Camera calibration
- Calibration – Projective camera model
- Calibration – Affine camera model

[material based on Paris Kaimakis]

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Today's Agenda

- Full camera model in matrix form
- Camera calibration
- Calibration - Projective camera model
- Calibration – Affine camera model

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Remember - Full camera model

It consists of three transformations:

1. The **Euclidean** (**Rigid**) **transformation** between the camera and the world, i.e., the translation and rotation of the camera with respect the world origin – takes points from 3D world coordinates to 3D camera coordinates

2. The **perspective projection** onto the camera plane – takes points from 3D camera coordinates to 2D image coordinates

3. **CCD imaging**, i.e., the geometry of the CCD array (the size and shape of the pixels) and its position with respect to the optical axis – takes points from 2D image coordinates to 2D pixel coordinates

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Remember - Full camera model – Putting it all together

So here is the full camera model combining all three transformations

$$\left.\begin{aligned} u &= u_0 + k_u x \\ v &= v_0 + k_v y \end{aligned}\right\} \Rightarrow \left.\begin{aligned} u &= u_0 + k_u \frac{f}{Z_c} X_c \\ v &= v_0 + k_v \frac{f}{Z_c} Y_c \end{aligned}\right\}$$

$$\Rightarrow \left.\begin{aligned} u &= u_0 + k_u f \frac{r_{11}X + r_{12}Y + r_{13}Z + T_x}{r_{31}X + r_{32}Y + r_{33}Z + T_z} \\ v &= v_0 + k_v f \frac{r_{21}X + r_{22}Y + r_{23}Z + T_y}{r_{31}X + r_{32}Y + r_{33}Z + T_z} \end{aligned}\right\}$$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Full camera model in matrix form - Euclidean

- How can we express all three transformations in the form of matrices ?
  - Homogeneous coordinates
- Let's start from the Euclidean (rigid) transformation connecting the world and camera coordinate systems.
- This is composed of a rotation (3DOF) expressing the camera pose with respect to the world coordinate frame, and a translation (3DOF) expressing the camera location with respect to the world origin.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

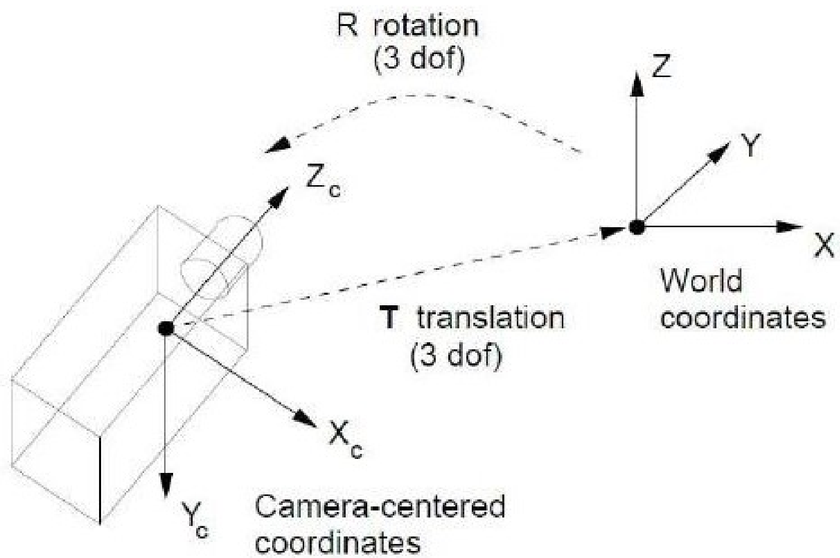
Full camera model in matrix form - Euclidean

- We want to express a ‘world’ 3D point **X**(X, Y, Z) as a 3D point $\mathbf{X_C}$ (X_C, Y_C, Z_C) in camera coordinates
- In homogeneous coordinates we have $\tilde{X}$(λX,λY,λZ,λ) and $\widetilde{X_C}$(λX_C,λY_C,λZ_C,λ) – λ can be set to 1 as it has no effect on $\mathbf{X_C}$, the cartesian equivalent of $\widetilde{X_C}$
- The Euclidean transformation can now be expressed as:

$$\begin{bmatrix} \lambda X_C \\ \lambda Y_C \\ \lambda Z_C \\ \lambda \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & T_x \\ r21 & r22 & r23 & T_y \\ r31 & r32 & r33 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda X \\ \lambda Y \\ \lambda Z \\ \lambda \end{bmatrix}$$

- Or equivalently:

$$\tilde{\mathbf{X}}_c = \mathbf{P}_e \tilde{\mathbf{X}} \quad \text{where} \quad \mathbf{P}_e = \left[\begin{array}{ccc|c} & \mathbf{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$



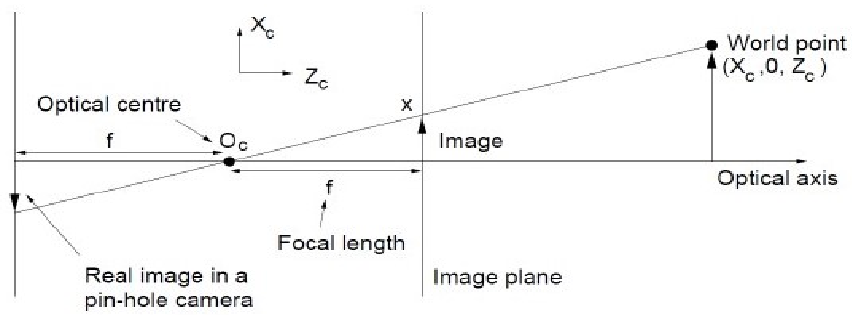
Co-financed by the European Union
Connecting Europe Facility
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Full camera model in matrix form - Perspective

- We want to compute the projection of a 3D point $\mathbf{X_C}$ (X_C, Y_C, Z_C) expressed in camera coordinates onto a 2D point $\mathbf{x}(x,y)$ lying on the image plane
- In homogeneous coordinates we have $\widetilde{X_C}(\lambda X_C, \lambda Y_C, \lambda Z_C, \lambda)$ and $\tilde{x}(sx,sy,s)$ – again λ and s can be set to 1
- Perspective projection can now be expressed as:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda X_c \\ \lambda Y_c \\ \lambda Z_c \\ \lambda \end{bmatrix}$$

$$\left. \begin{aligned} x &= \frac{f}{Z_c} X_c \\ y &= \frac{f}{Z_c} Y_c \end{aligned} \right\}$$



- Or equivalently:

$$\tilde{\mathbf{x}} = \mathbf{P}_p \, \tilde{\mathbf{X}}_c$$

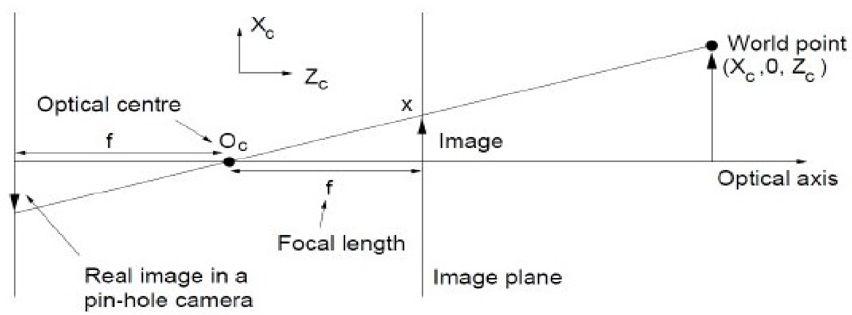
Co-financed by the European Union
Connecting Europe Facility
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Full camera model in matrix form - Perspective

- We can confirm that the expression in the previous slide is equivalent to the perspective equation by recovering **x**, the cartesian equivalent of $\tilde{x}$

$$\tilde{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f\lambda X_C \\ f\lambda Y_C \\ \lambda Z_C \end{bmatrix} \Rightarrow x = \begin{bmatrix} sx/s \\ sy/s \end{bmatrix} = \begin{bmatrix} f\lambda X_C/\lambda Z_C \\ f\lambda Y_C/\lambda Z_C \end{bmatrix} = \begin{bmatrix} fX_C/Z_C \\ fY_C/Z_C \end{bmatrix}$$

$$\left.\begin{aligned} x &= \frac{f}{Z_c} X_c \\ y &= \frac{f}{Z_c} Y_c \end{aligned}\right\}$$

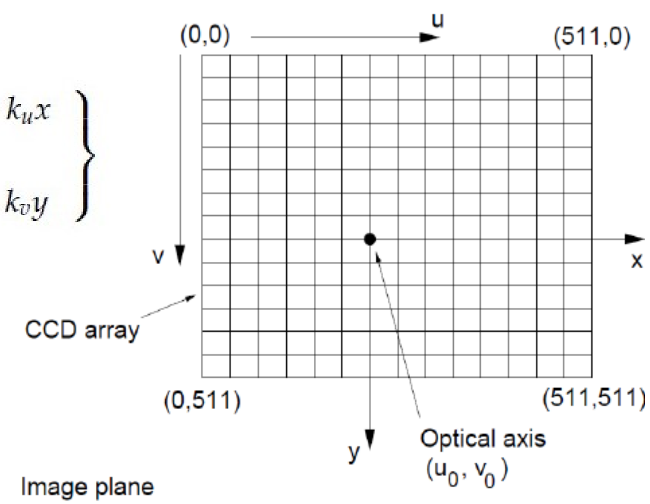


Full camera model in matrix form – CCD Imaging

- Last, wee want to express 2D point **x**(x,y) lying on the image plane in pixel coordinates **w**(u,v)
- In homogeneous coordinates we have $\tilde{x}$(sx,sy,s) and $\widetilde{w}$(su,su,s) – again s can be set to 1
- This is a translation and scaling which can be expressed as:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx \\ sy \\ s \end{bmatrix}$$

$$\left.\begin{aligned} u &= u_0 + k_u x \\ v &= v_0 + k_v y \end{aligned}\right\}$$



- Or equivalently:

$$\tilde{\mathbf{w}} = \mathbf{P}_c \tilde{\mathbf{x}}$$

Full camera model in matrix form – all together

We can now express the whole process, from $\tilde{X}$ to $\tilde{w}$ as a single transformation P_{ps}

$$
\begin{aligned}
\tilde{\mathbf{w}} &= \mathbf{P}_{ps}\,\tilde{\mathbf{X}} \\
\text{where}\quad \mathbf{P}_{ps} &= \mathbf{P}_c\,\mathbf{P}_p\,\mathbf{P}_e \\
&= \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & \mathbf{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
\end{aligned}
$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Full camera model in matrix form – all together

- The transformation P_{ps} is not a general 3x4 matrix, because it has a special structure composed of P_e, P_p,and P_c.
- We can simplify P_{ps} into an upper-triangular matrix K composed of P_e and P_p, and a matrix representing the Euclidean transformation.

$$\mathbf{P}_{ps} = \mathbf{K}\,[\mathbf{R}\,|\,\mathbf{T}\,]$$

$$= \begin{bmatrix} m_u & 0 & u_0 & 0 \\ 0 & m_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 & T_x \\ r21 & r22 & r23 & T_y \\ r31 & r32 & r33 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$m_u = k_u\, f$$
$$m_v = k_v\, f$$

Full camera model in matrix form – all together

- The matrix **K** is called the camera calibration matrix and it contains all the viewing parameters (**intrinsics**) coming from inside the camera.
- The Euclidean transformation matrix contains all the viewing parameters (**extrinsics**) that are not controlled by the camera lens or sensor.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Today’s Agenda

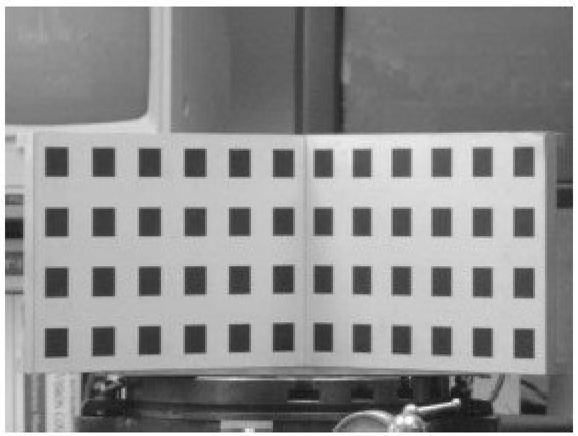
- Full camera model in matrix form
- Camera calibration
- Calibration - Projective camera model
- Calibration – Affine camera model

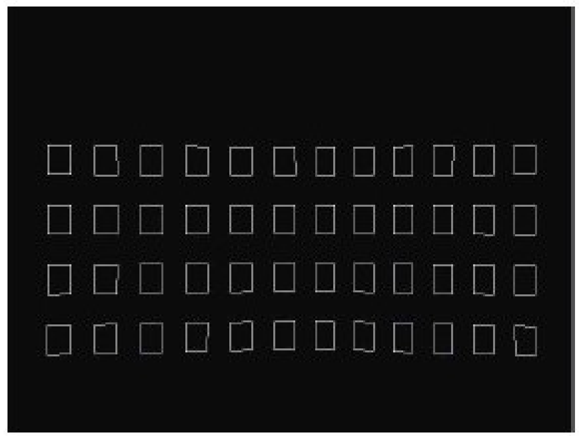
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration

- Camera calibration is the name given to the process of discovering the parameters inside our camera model, i.e., the values inside the camera projection matrix.
- This is done by using an image of a controlled scene.
- We may want to use a scene with some sort of regular pattern.

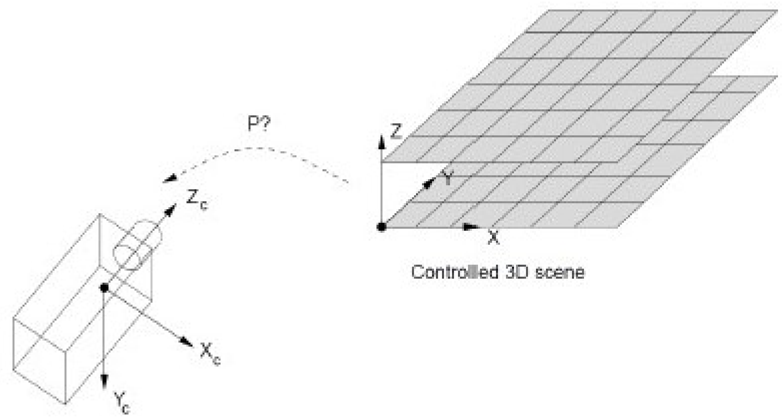




This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration

- Camera calibration is the name given to the process of discovering the parameters inside our camera model, i.e., the values inside the camera projection matrix. This is done by using an image of a controlled scene.
- We may want to use a scene with some sort of regular pattern



Today’s Agenda

- Full camera model in matrix form
- Camera calibration
- Calibration - Projective camera model
- Calibration – Affine camera model

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration – the projective camera

Remember that the **perspective** camera projection matrix P_{ps} is not a general 3x4 matrix, as it has a special structure composed of P_e, P_p,and P_c.

$$\tilde{\mathbf{w}} = \mathbf{P}_{ps}\tilde{\mathbf{X}}$$

$$\text{where } \mathbf{P}_{ps} = \mathbf{P}_c\,\mathbf{P}_p\,\mathbf{P}_e$$

$$= \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & \mathbf{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Camera calibration – the projective camera

- Calibrating the perspective camera model may therefore be difficult. We can instead use the projective camera model which is described by a general 3x4 matrix.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

- The projective camera has **11 degrees of freedom**, since the overall scale of P does not matter when using homogeneous coordinates.
- It is more convenient to deal with a projective camera than a perspective one, since we don't have to worry about placing nonlinear constraints on the elements of P

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration – the projective camera

- Using a projective camera, the whole imaging process is described by

$$\tilde{\mathbf{w}} = \mathbf{P}\tilde{\mathbf{X}}$$

$$\Rightarrow \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- We must estimate 11 parameters (the overall scale does not matter, so let's set $p_{34} = 1$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration – the projective camera

- Because the scene is controlled, we **know** the location of the world points $\mathbf{X}_i$
- We can also find the location of their projections $\mathbf{w}_i$ on the image, e.g., using corner detection.
- Each point we observe gives us two equations

$$u_i = \frac{su_i}{s} = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1}$$

$$v_i = \frac{sv_i}{s} = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + 1}$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration – the projective camera

Rearranging them gives us two linear equations in the unknown parameters of the projection matrix

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & u_iX_i & u_iY_i & u_iZ_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & v_iX_i & v_iY_i & v_iZ_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix} = \begin{bmatrix} -u_i \\ -v_i \end{bmatrix}$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Camera calibration – the projective camera

- As there are 11 unknowns, we need at least 6 points to calibrate this camera model to get enough equations (rows) in the linear system Ax=b.
- Each observed points adds two equations (rows) to the matrix A
- We can solve the system of equations using linear least squares (pseudo-inverse of A)

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \Rightarrow \quad \mathbf{x} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{aligned}$$

Today’s Agenda

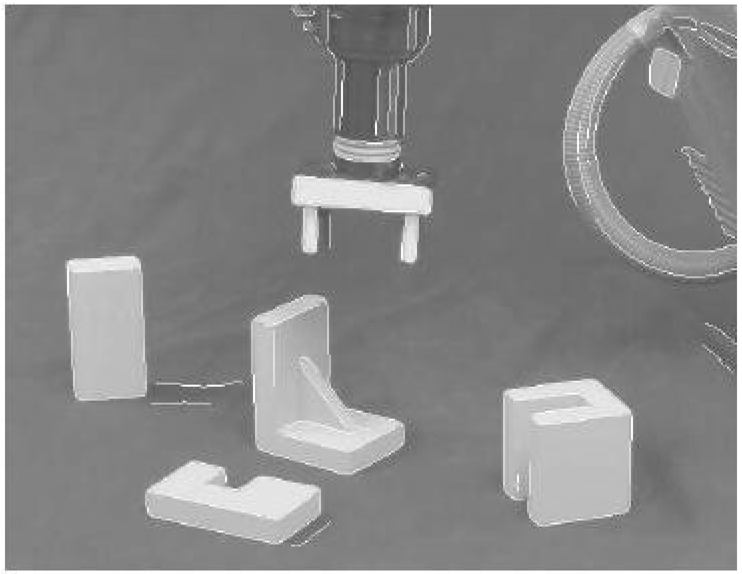
- Full camera model in matrix form
- Camera calibration
- Calibration – Projective camera model
- Calibration – Affine camera model

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Parallel Projection

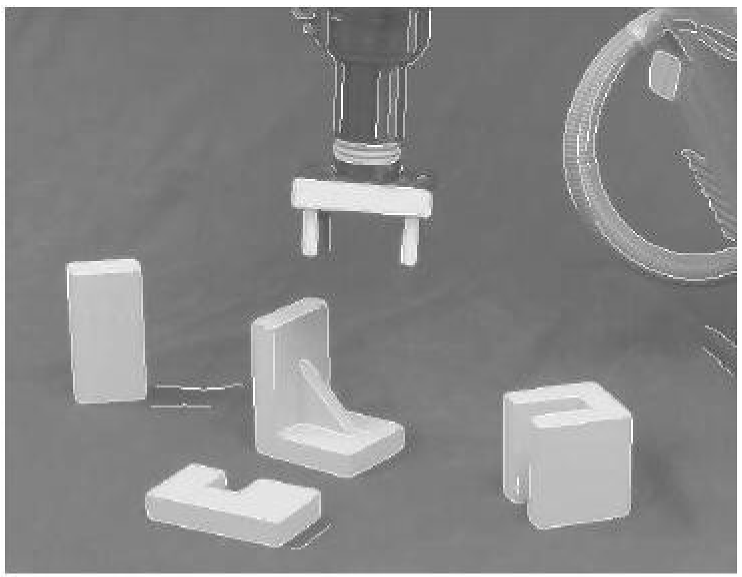
When the depth difference ΔZ_C of objects in the scene is small compared to their distance Z_C to the camera, the resulting image is not described well by perspective projection.



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Parallel Projection

For example, parallel lines on the left image remain parallel after projection. In this case, the 3D to 2D projection is better described by **parallel projection**.

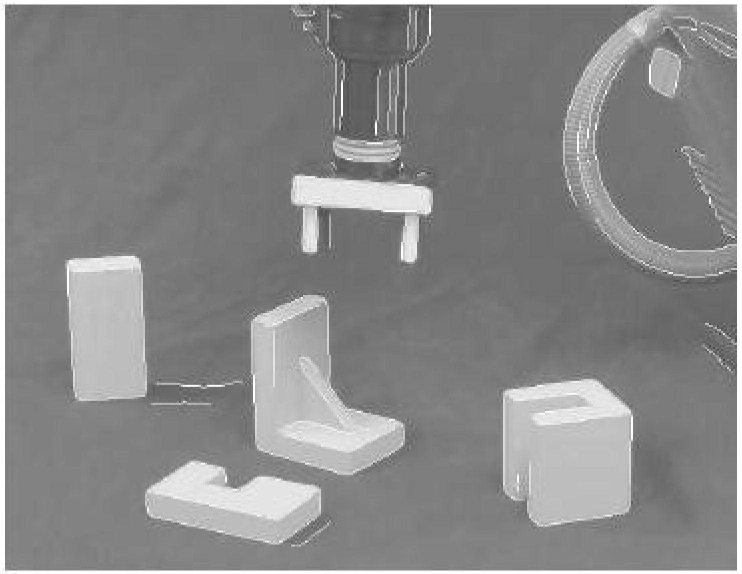


Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Parallel Projection

A camera which creates images like the one on the left is known as a **weak-perspective camera.**



Parallel Projection

What changes in our imaging transformations is the perspective projection matrix. Remember P_p:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Parallel Projection

What changes in our imaging transformations is the perspective projection matrix.

If the depth variation in the scene is small, then $Z_C \approx Z_C^{avg}$ and the perspective projection matrix P_p can be changed to the parallel projection matrix P_{pll}

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & Z_c^{\text{avg}} \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{x}} = \mathbf{P}_{pll} \tilde{\mathbf{X}}_c$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Weak-Perspective Camera Model

This leads to the **weak perspective** camera model:

$$\tilde{\mathbf{w}} = \mathbf{P}_{wp}\,\tilde{\mathbf{X}}$$

where

$$\begin{aligned}\mathbf{P}_{wp} &= \mathbf{P}_c\,\mathbf{P}_{pll}\,\mathbf{P}_e \\ &= \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & Z_c^{\text{avg}} \end{bmatrix} \left[\begin{array}{ccc|c} & \mathbf{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]\end{aligned}$$

Weak-Perspective Camera Model

P_{wp} is the projection matrix for a weak-perspective camera. It is a 3x4 matrix with a special structure composed of P_e, P_{pll}, and P_c.

$$\tilde{\mathbf{w}} = \mathbf{P}_{wp}\,\tilde{\mathbf{X}}$$

where

$$\begin{aligned}\mathbf{P}_{wp} &= \mathbf{P}_c\,\mathbf{P}_{pll}\,\mathbf{P}_e \\ &= \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & Z_c^{\text{avg}} \end{bmatrix} \left[\begin{array}{ccc|c} & \mathbf{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]\end{aligned}$$

Camera calibration – the affine camera

- The special structure of the weak-perspective camera model makes it difficult to calibrate. This is why the affine camera model is often used instead:

$$\mathbf{P}_{aff} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{bmatrix}$$

- P_{aff} is the projection matrix for the affine camera. It has 8 degrees of freedom (remember the overall scale does not matter so we can set p_{34} to 1).
- It can be calibrated in the same way as the projective camera. There are eight degrees of freedom, so we need a minimum of 4 points.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU | Master programmes in Artificial Intelligence 4 Careers in Europe

Visual Computing Group

CYENS CENTRE OF EXCELLENCE

Thank you.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423