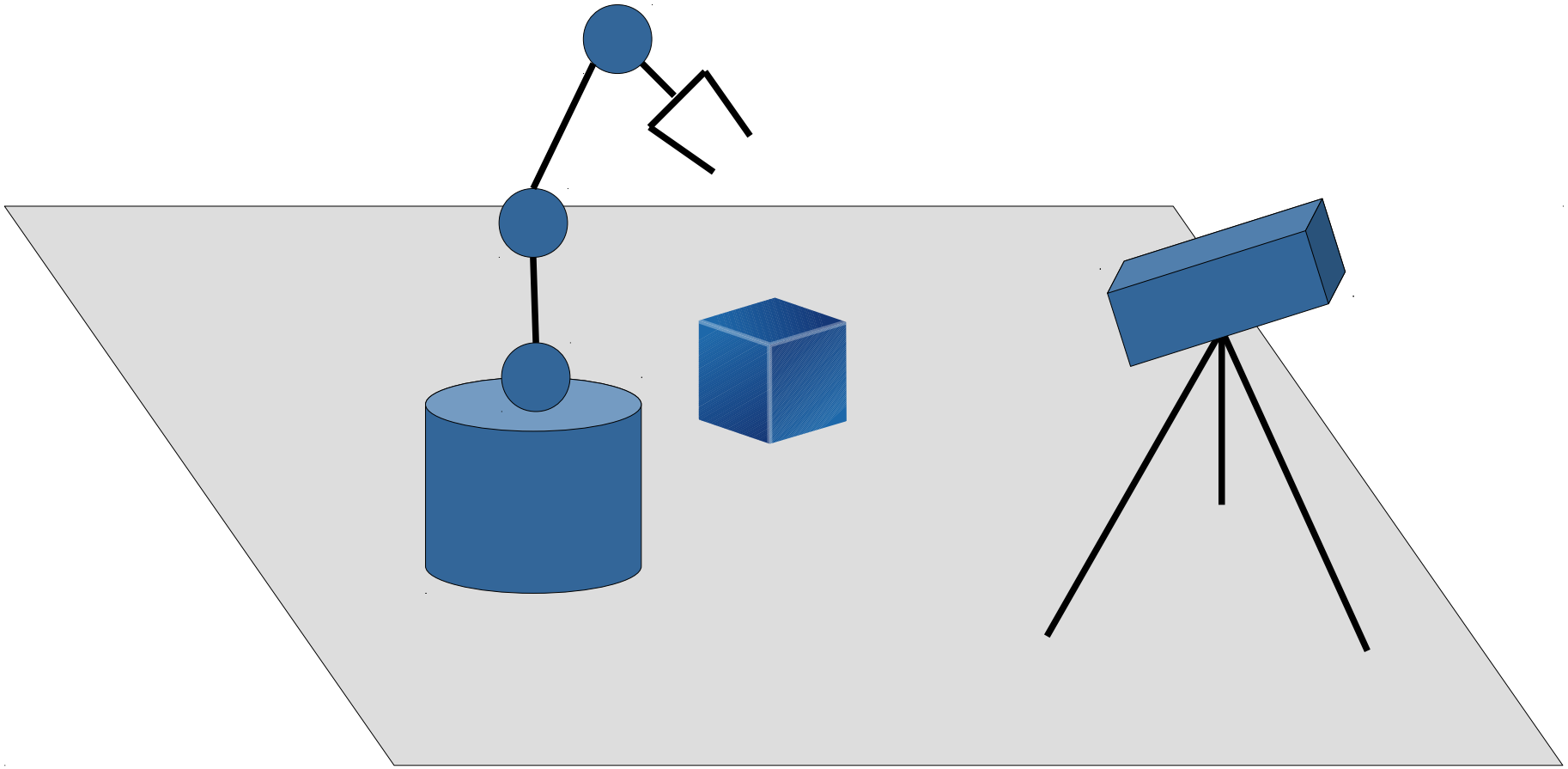
Colour-Based Object Detection, Inverse Kinematics Algorithms and Pinhole Camera Model for Controlling Robotic Arm Movement System

12th International Conference on Electronics, Computer and Computation

27th September — 30 September 2015
Almaty, Kazakhstan

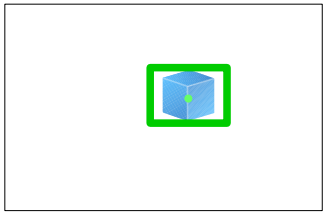
Ravil Mussabayev

Problem Statement

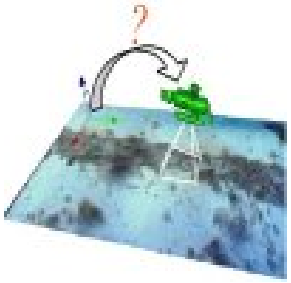


Problem Statement

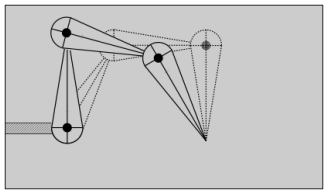
1. Camera calibration



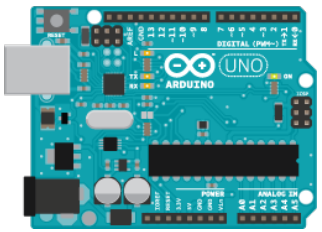
2. Recognition of the object on the input camera frame



3. Projection of the coordinates into the real world



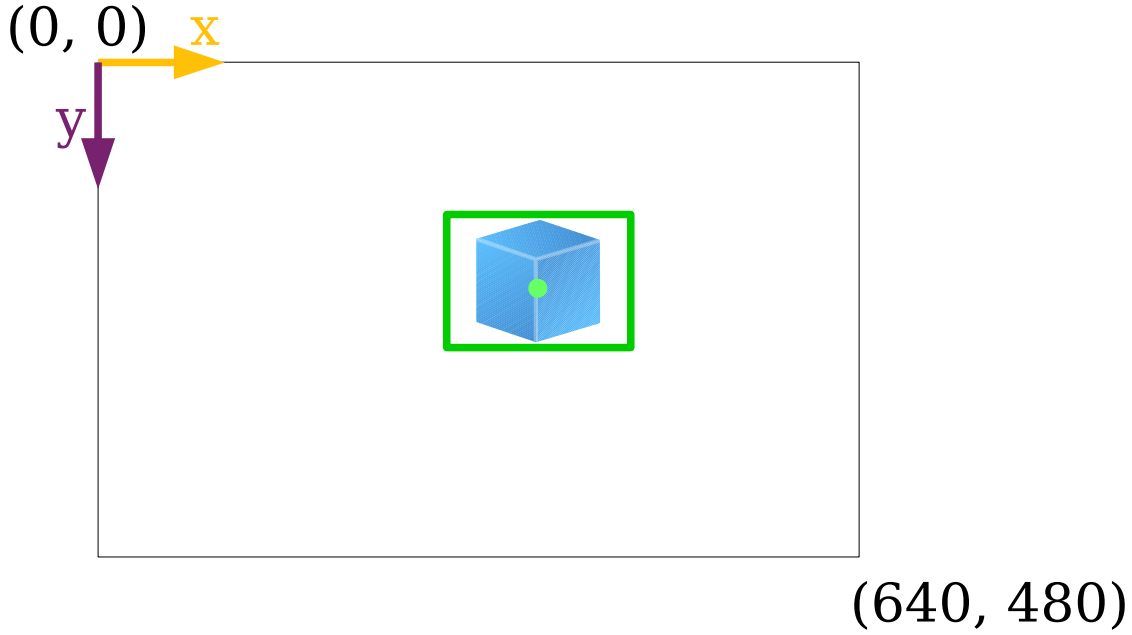
4. Solving the problem of inverse kinematics



5. Developing of the low-level software for Arduino microcontroller

Object Recognition

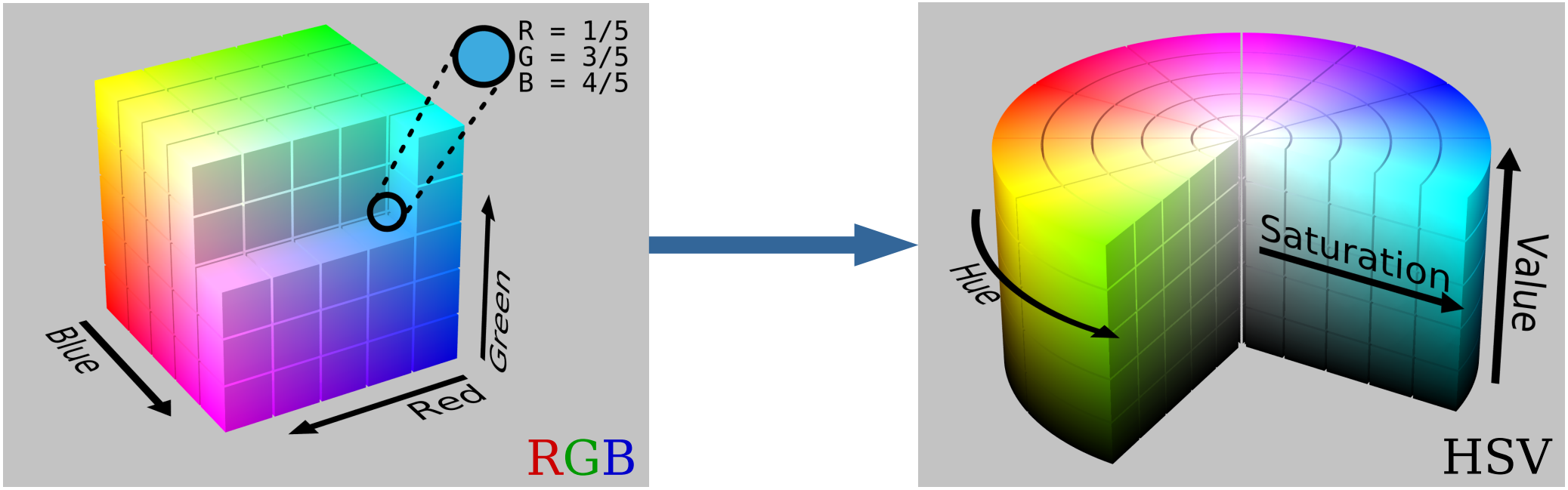
Goal: highlight the object on the input camera frame with bounding rectangle for the purpose of estimating coordinates of the tentative centre of mass relative to the reference frame associated with 2D image.



- Is used for detection of an object of the preliminarily specified color;
- Unsusceptible to presence of regions on the object's surface of colour other than that of specified.

Object Recognition

1. Three-channel matrix of the input image is transformed from the colour space RGB (Red, Green, Blue) into HSV (Hue, Saturation, Value), since this allows to most efficiently specify the colour range of required hue;

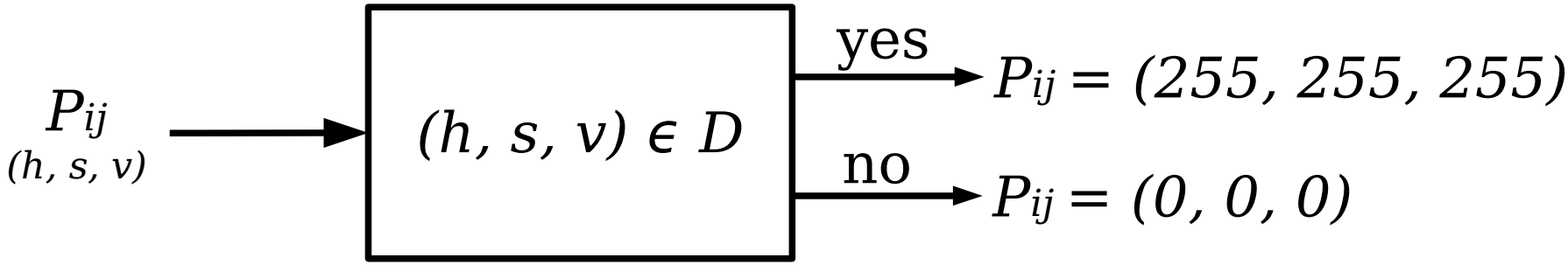


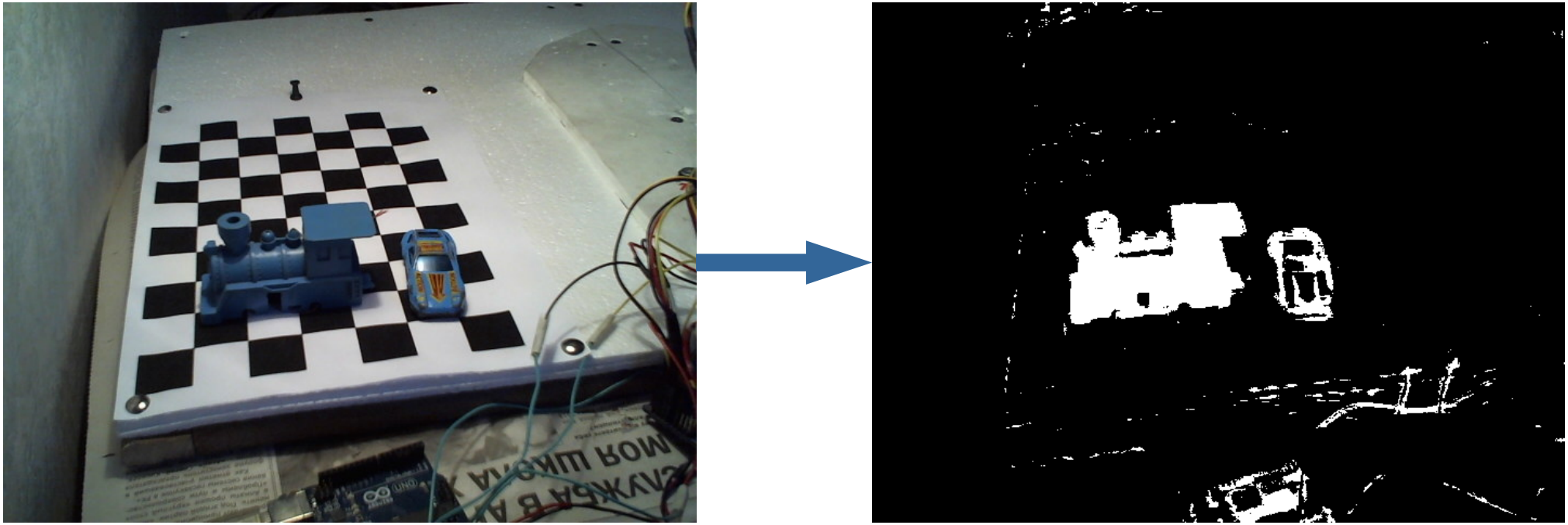
For instance, the range D for colours of blue hue:

$D =$ (200, 30, 90) ——— (250, 100, 100)

Object Recognition

2. Then the image is rendered binary by checking belonging of each pixel P_{ij} to the range D.

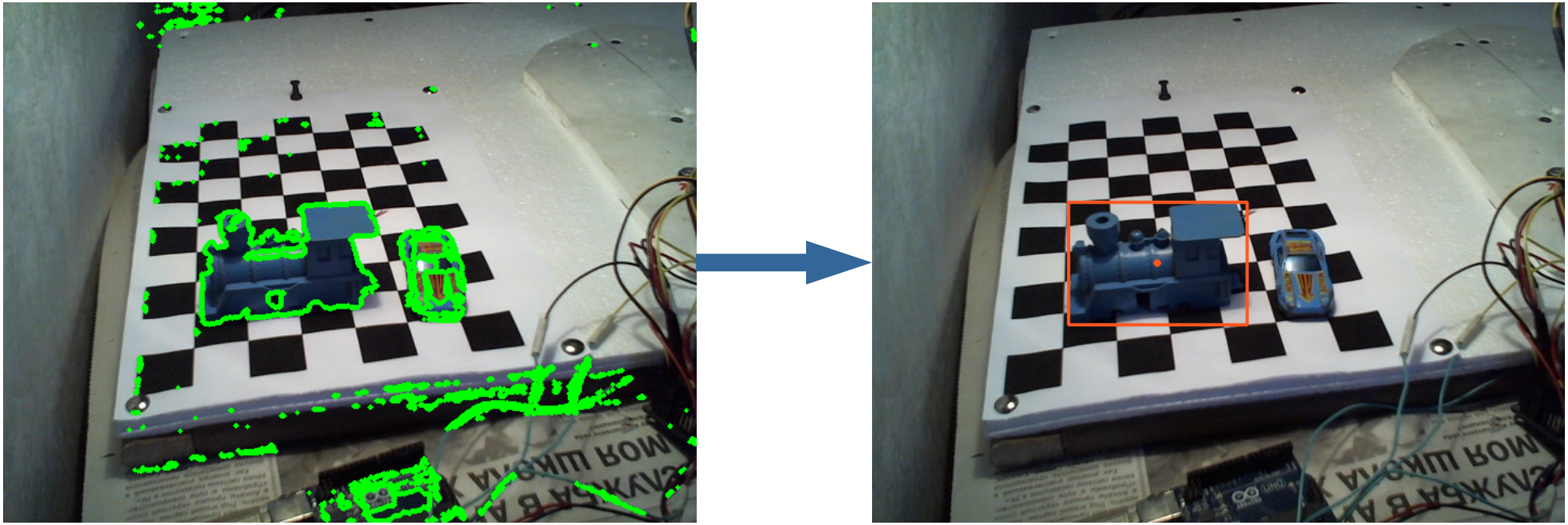




Object Recognition

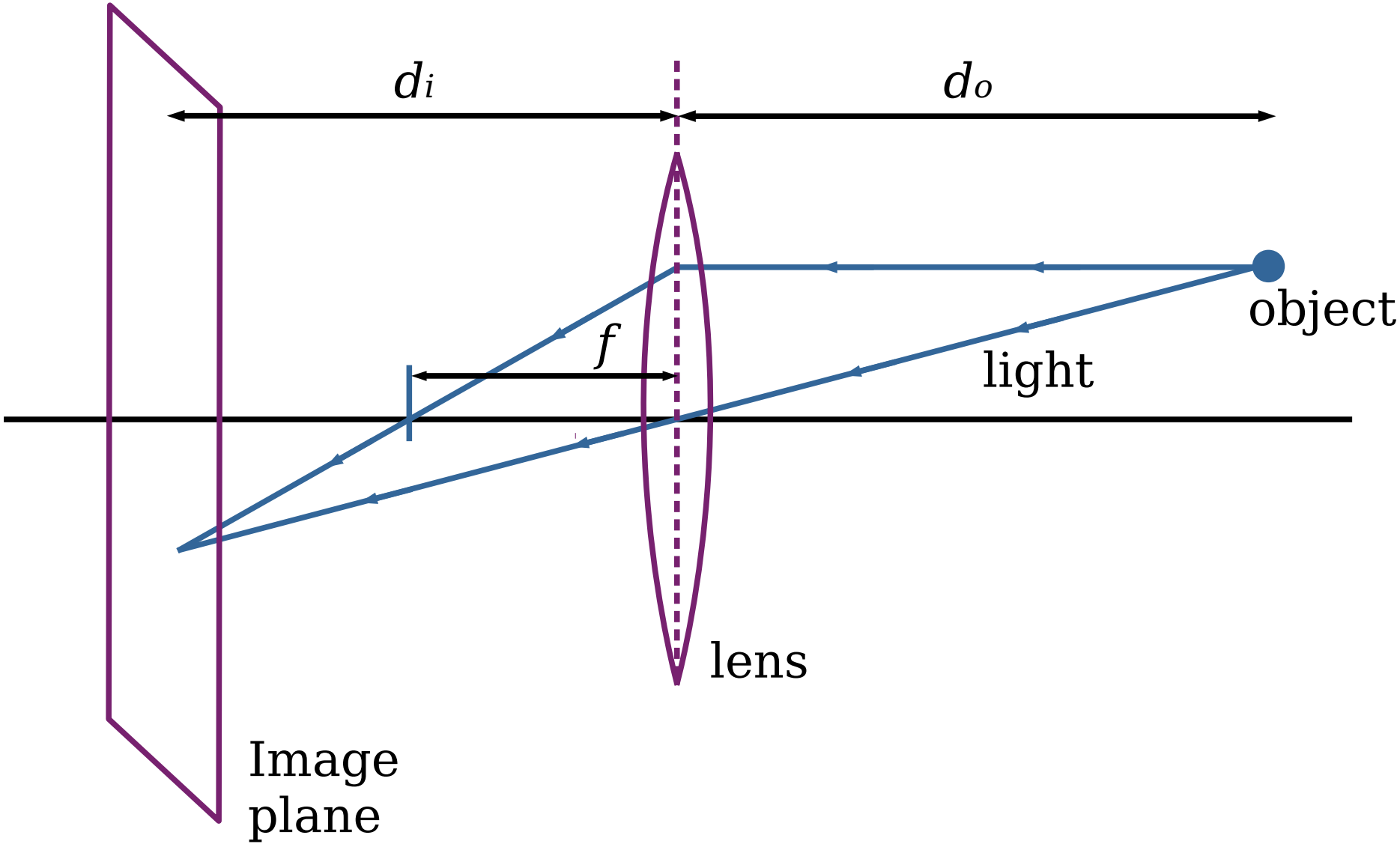
3. Realized in the OpenCV library border following algorithm is subsequently applied to the obtained mask, so that the algorithm returns an array of recognized contours, from which the one with the largest area is to be selected.

4. Utmost points of the contour are chosen as the coordinates for the upper left and lower right corners of the bounding rectangle respectively.



Camera

- Light beams passing through the lens in the objective.



Thin Lens Equation

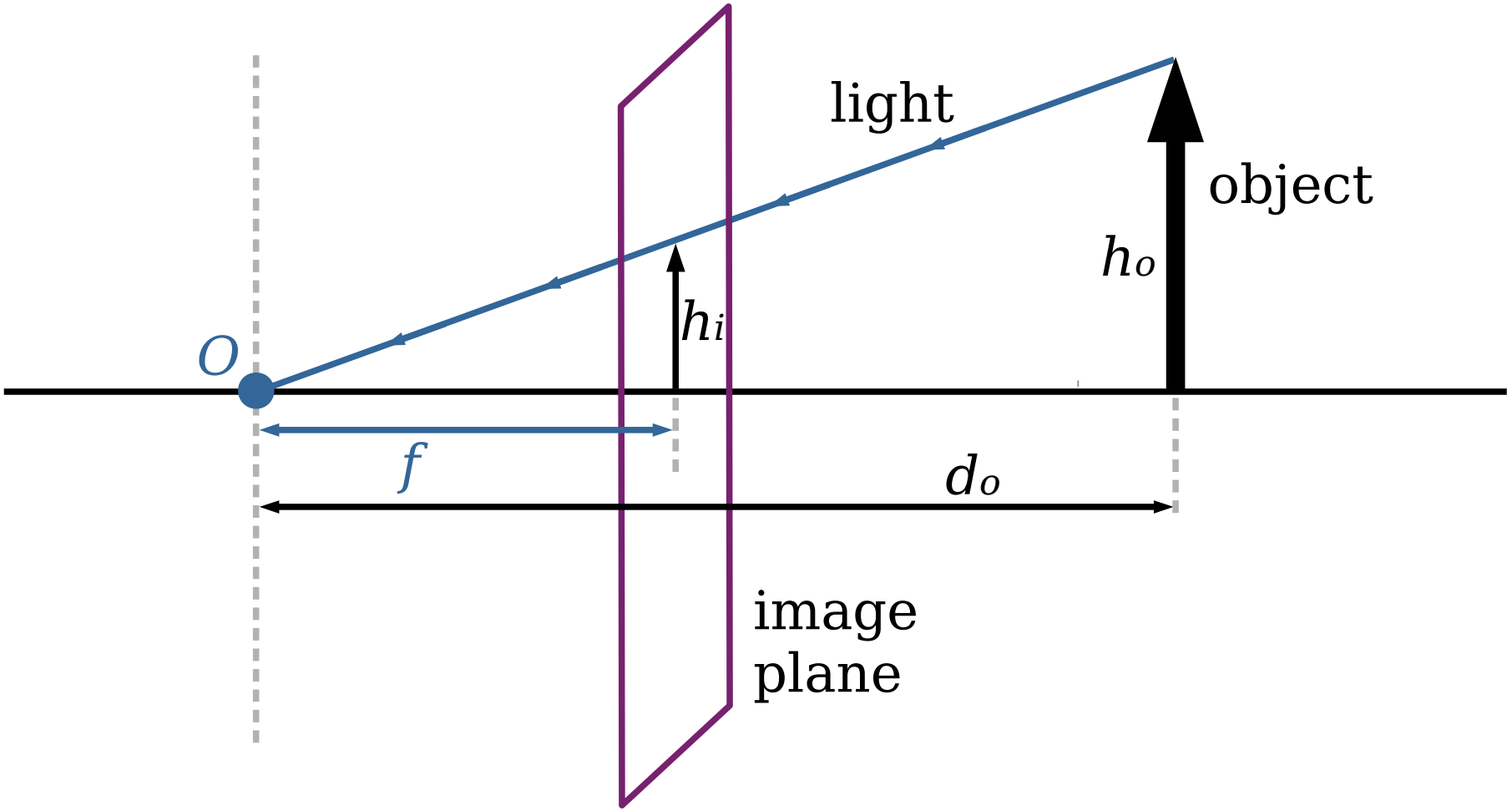
$$\frac{1}{f} = \frac{1}{d_0} + \frac{1}{d_i} \tag{1}$$

- d_o — distance from the object to the lens;
- d_i — distance from the lens to the image plane;
- f — focal distance.

Pinhole Camera Model

Simplified model of camera used in computer vision:

- The lens effect is neglected;
- $d_o >> d_i$. Then we may assume that $d_i = f$;
- Image is imaginary and upright.

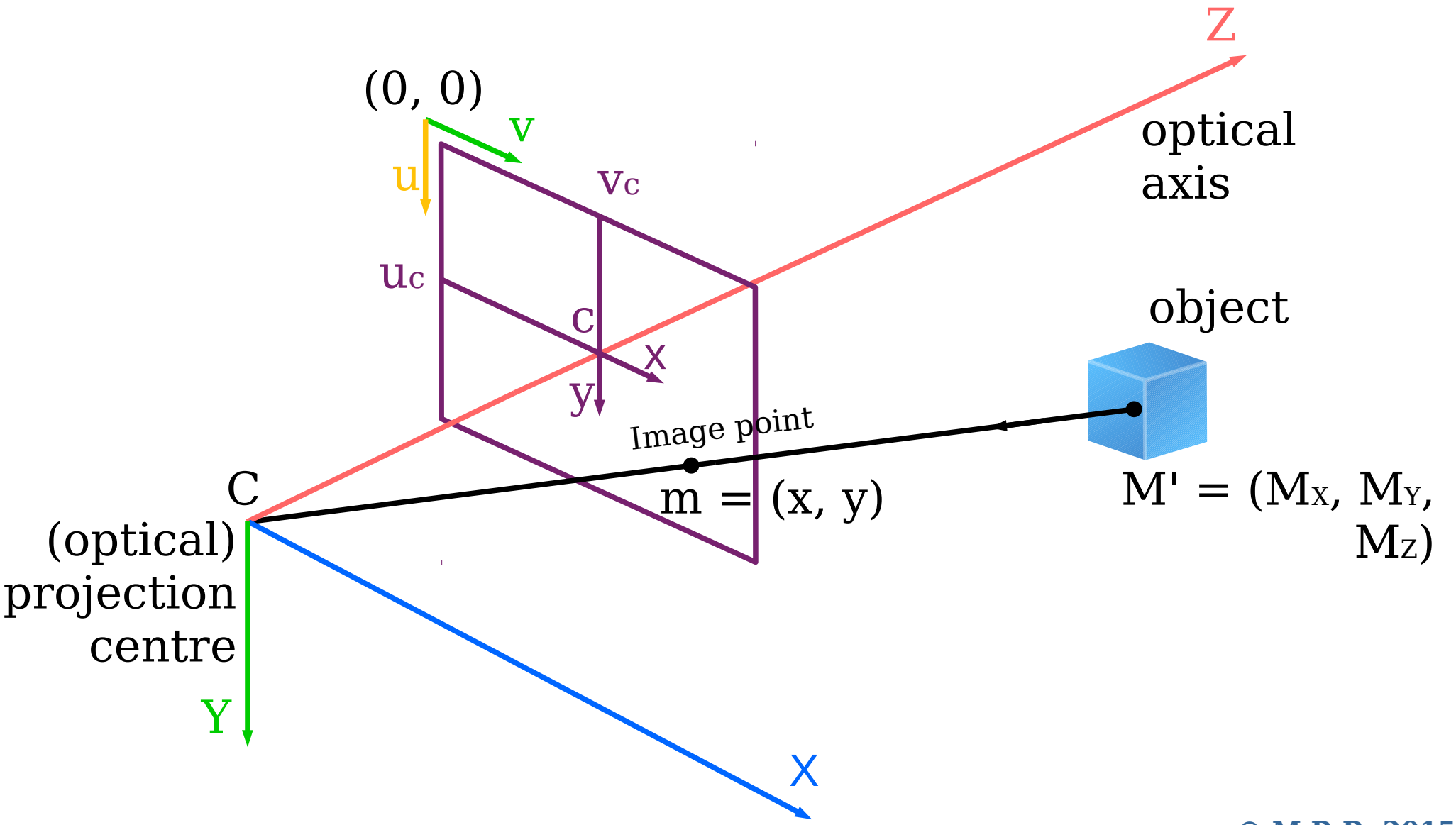


Pinhole Camera Model

$$h_i = f \frac{h_0}{d_0} \tag{2}$$

- h_i — height of the image;
- d_o — distance from the world object to the oprical centre;
- h_o — height of the world object;
- f — focal distance.

Coordinate Transformations



Coordinate Transformations

$$x = f\frac{M_X}{M_Z}; y = f\frac{M_Y}{M_Z} \tag{3}$$

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} M_X \\ M_Y \\ M_Z \\ 1 \end{bmatrix} \tag{4}$$

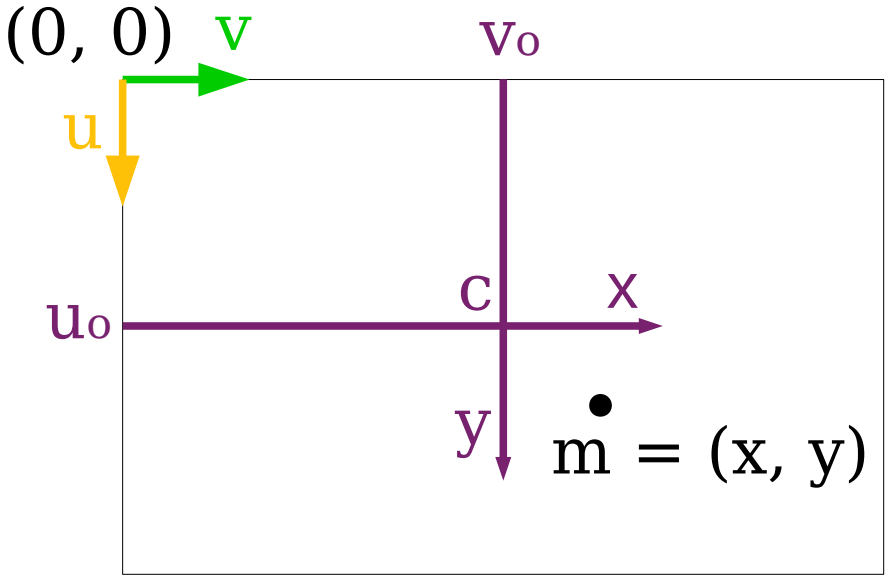
- x, y — coordinates of the image point in the system cxy;
- M_X, M_Y, M_Z — coordinates of the object in CXYZ;
- s — scalar factor different from zero;
- f — focal distance.

Coordinate Transformations

$$u = u_0 + \frac{x}{pixel\,width};\ v = v_0 + \frac{y}{pixel\,height} \tag{5}$$

(3) → (5): $$M_Z u = M_Z u_0 + f_x M_X;\ M_Z v = M_Z v_0 + f_y M_Y \tag{6}$$

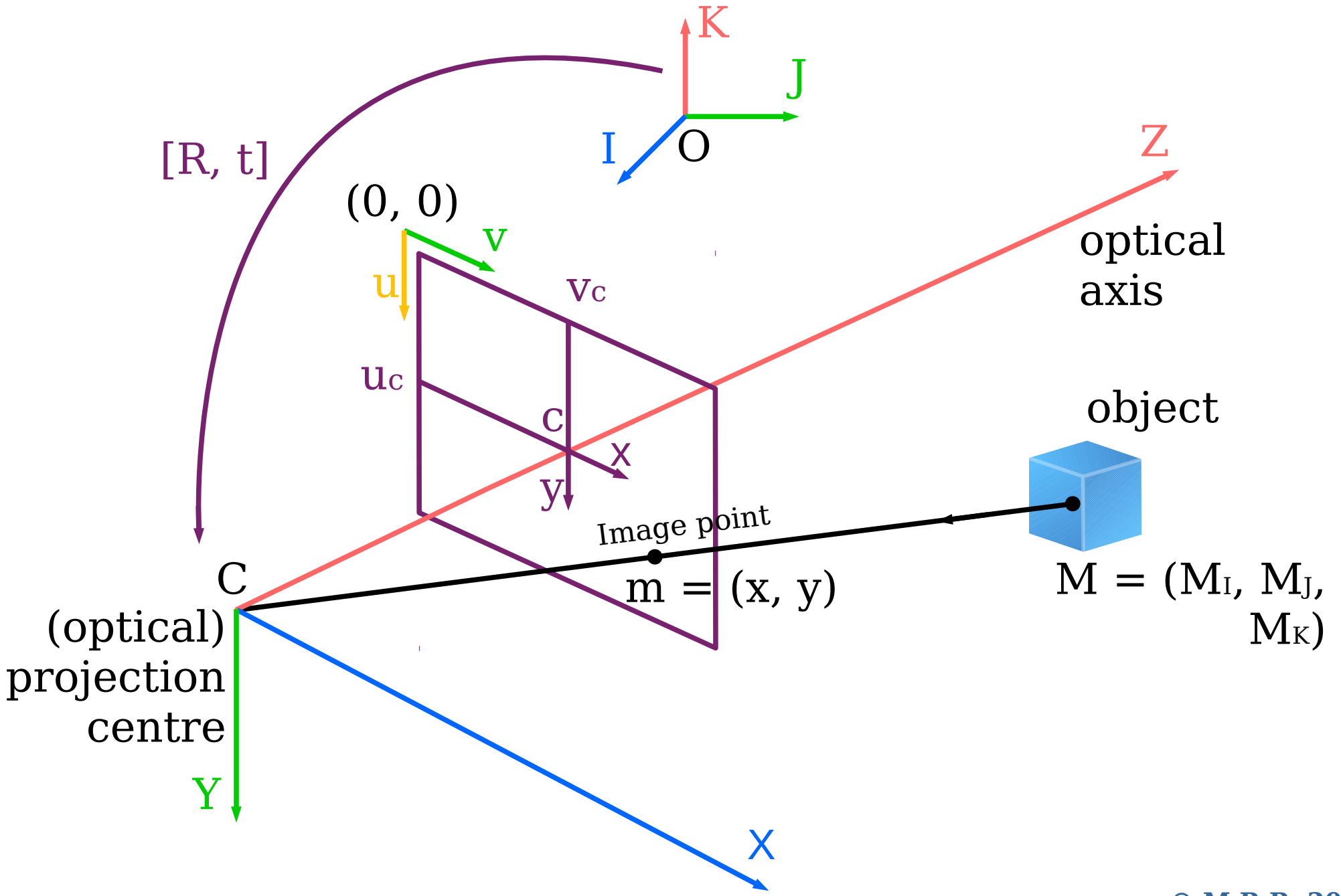
Where: $$f_x = \frac{f}{pixel\,width};\ f_y = \frac{f}{pixel\,height} \tag{7}$$



$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} M_X \\ M_Y \\ M_Z \\ 1 \end{bmatrix} \tag{8}$$

$$sm' = AM' \tag{9}$$

Coordinate Transformations



Coordinate Transformations

$$M = \begin{bmatrix} M_I \\ M_J \\ M_K \\ 1 \end{bmatrix} \xrightarrow{[R,\ t]} M' = \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix} \tag{10}$$

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{11}$$

$$M' = [R|t]\, M \tag{12}$$

Coordinate Transformations

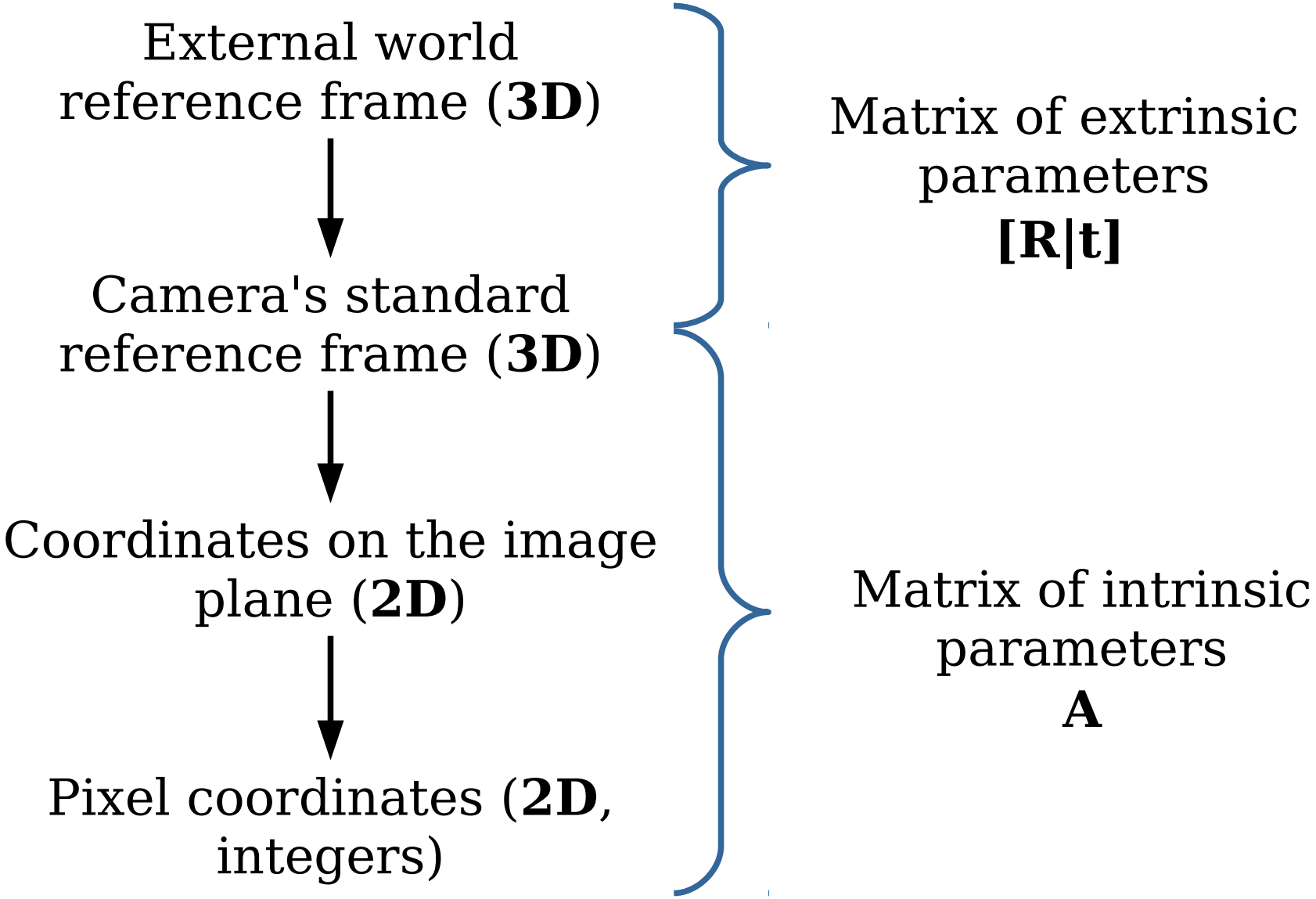
- Fundamental equation, establishing the relation between the 2D image and the real world.

$$s\,m' = A\,[R|t]\,M \tag{13}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} M_I \\ M_J \\ M_K \\ 1 \end{bmatrix} \tag{14}$$

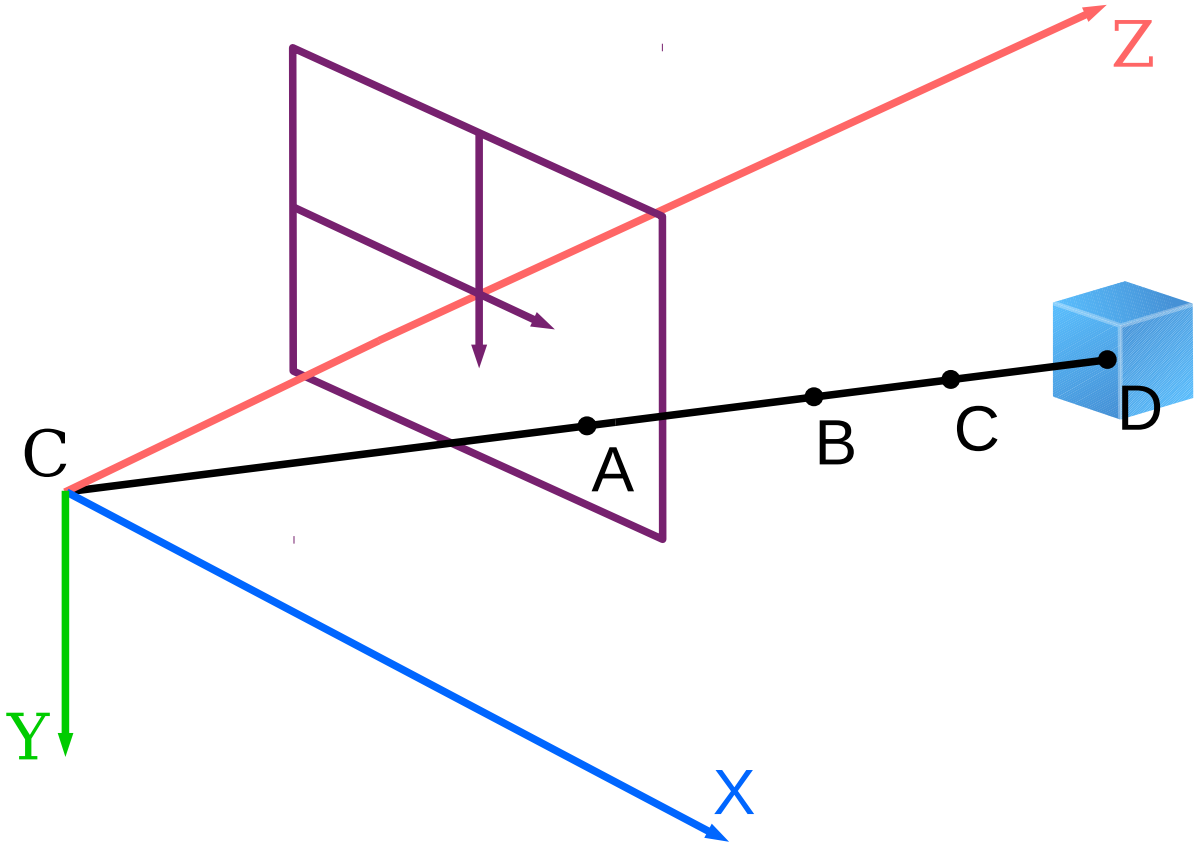
- m' — coordinate vector in the system of 2D image;
- A — matrix of intrinsic parameters;
- $[R|t]$ — matrix of extrinsic parameters;
- M — object's coordinate vector in the real world.

Coordinate Transformations

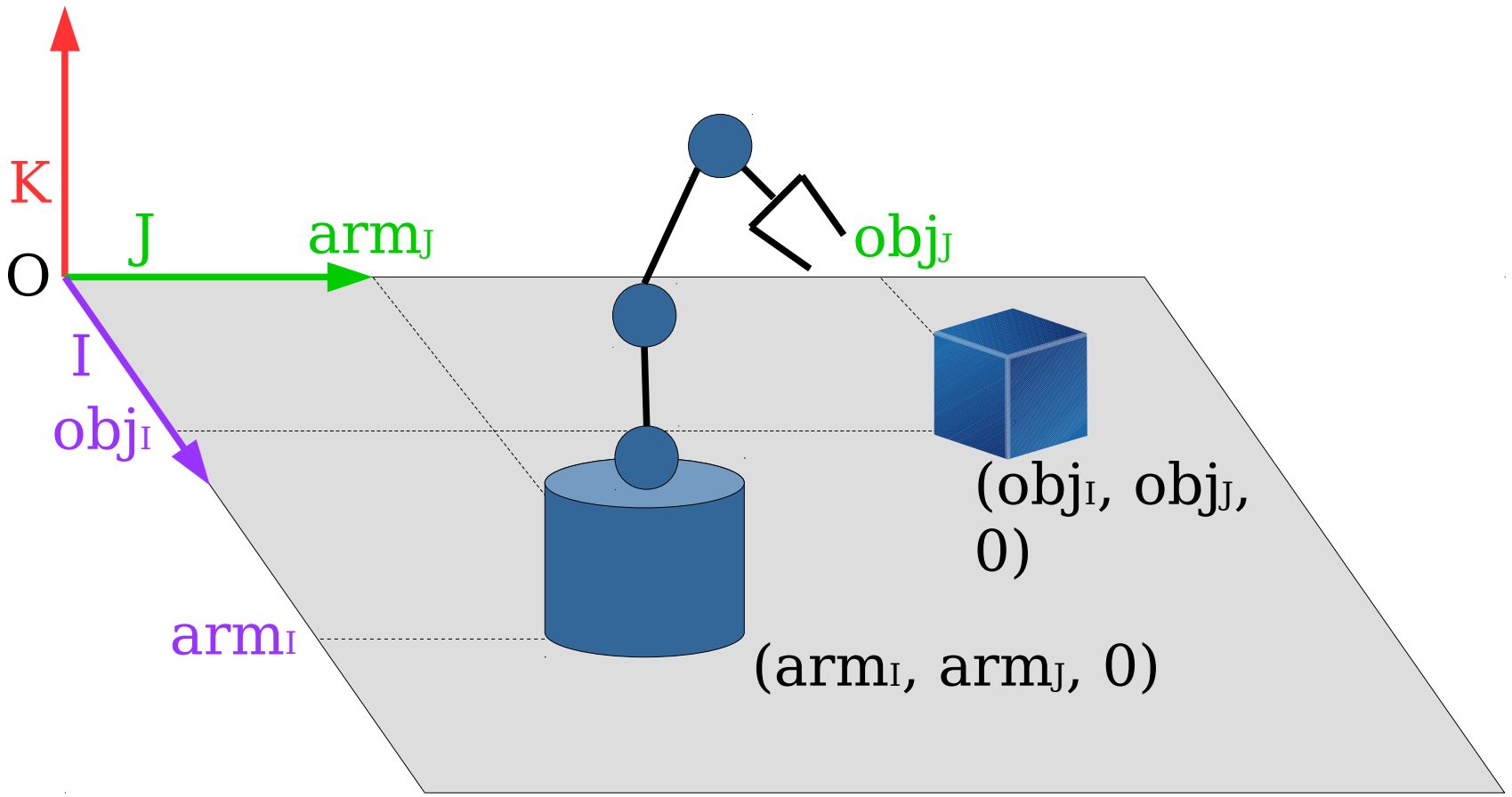


Coordinate Transformations

- Fundamental equation (14) enables to project points from the real world onto the image plane with just a single camera observing the scene;
- But for solving the inverse problem it is essential to either know one of the 3D coordinates beforehand or receive data simultaneously from two cameras viewing the scene.



Coordinate Transformations



Coordinate Transformations

$M_K=0 \rightarrow (5)$:

$$s\,m' = A \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} M_I \\ M_J \\ 0 \\ 1 \end{bmatrix} \tag{15}$$

$$s\,m' = A \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} M_I \\ M_J \\ 1 \end{bmatrix}; \qquad \begin{bmatrix} M_I \\ M_J \\ 1 \end{bmatrix} = s \left(A \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \right)^{-1} m' \tag{16}$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \left(A \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \right)^{-1} m'; \qquad \begin{bmatrix} M_I \\ M_J \\ 1 \end{bmatrix} = \begin{bmatrix} s\,e_1 \\ s\,e_2 \\ s\,e_3 \end{bmatrix} \tag{17}$$

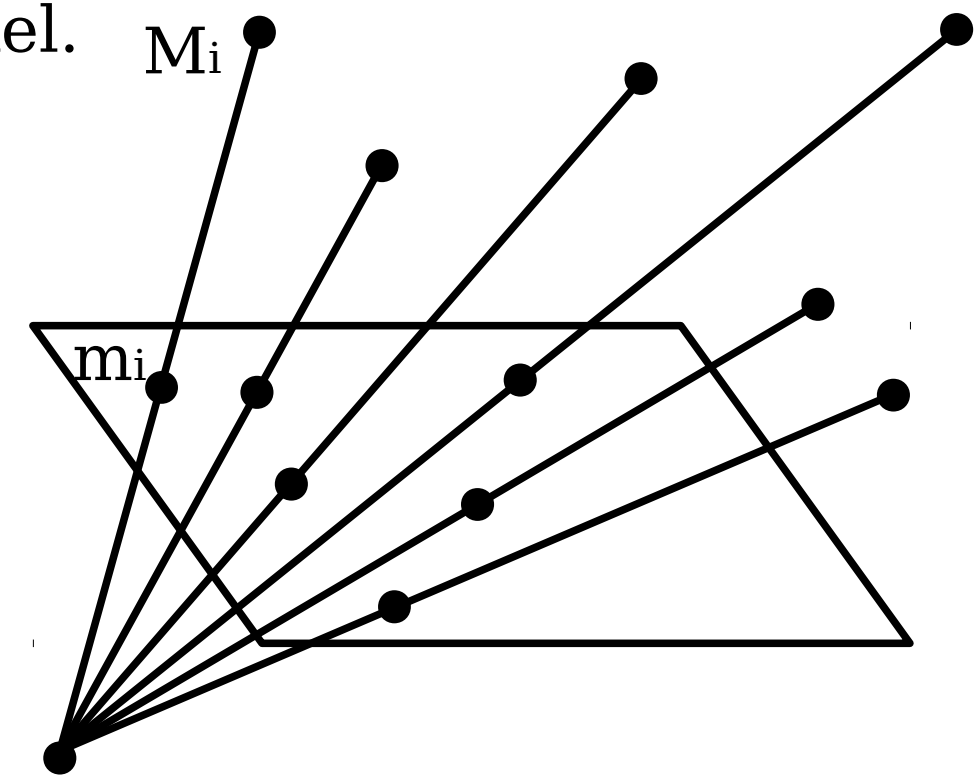
$$s = \frac{1}{e_3}; M_I = \frac{e_1}{e_3}; M_J = \frac{e_2}{e_3} \tag{18}$$

Camera Calibration

Camera calibration process helps to get:

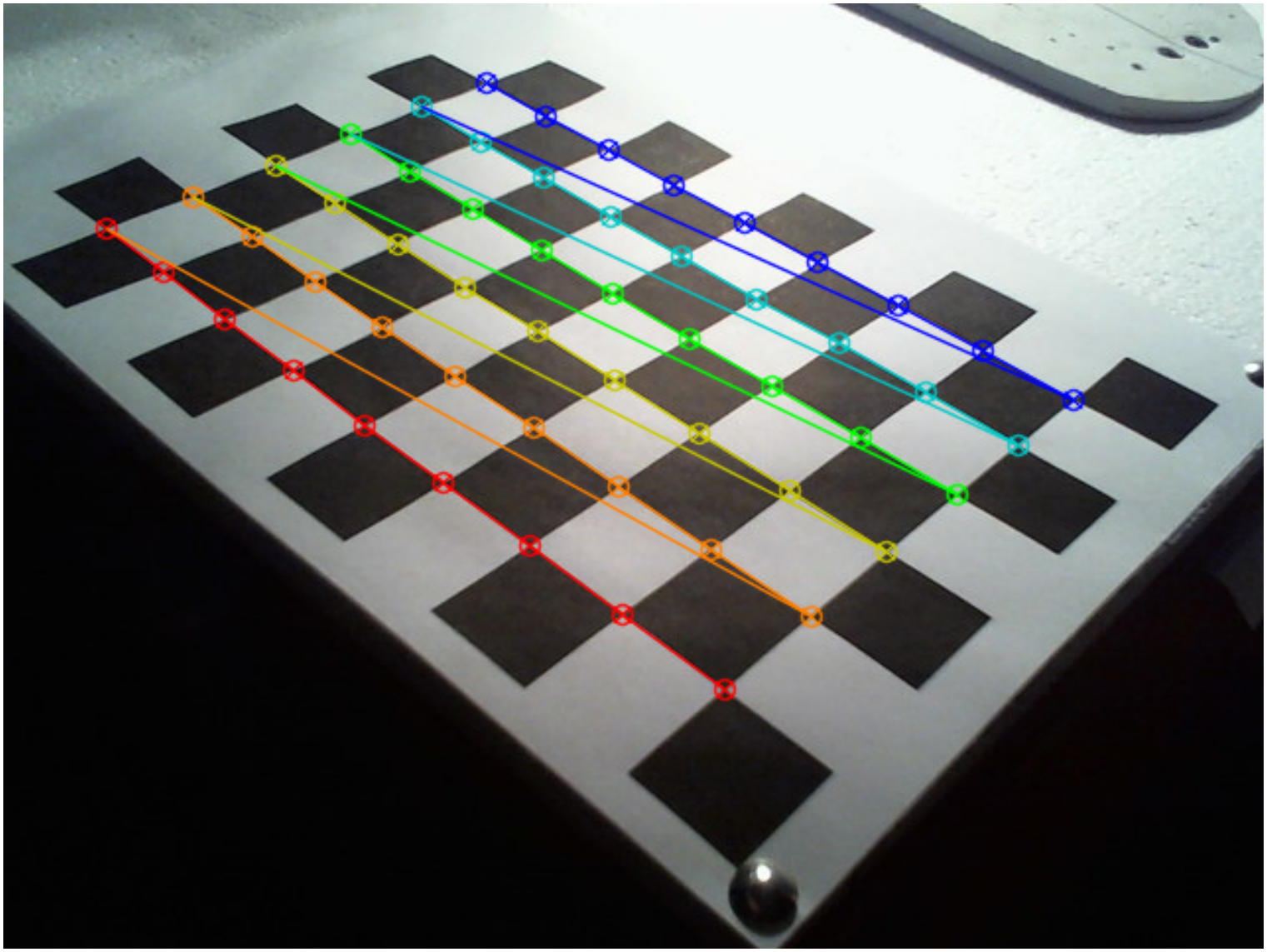
1. Intrinsic parameters of the camera:
   - Focal distance of the lens *f*;
   - Coordinates of the principal point of the image (u_o, v_o);
   - Width and height of the pixel.
2. Extrinsic parameters:
   - Rotation vector;
   - Translation vector.

11 parameters overall.



Calibration requires at least 6 correspondences of the points on the image and in the space, viz.
each correspondence of the spacial point M_i to the point on the image m_i generates 2 independent equations.

Realization in OpenCV



Recognition of the intersection points of the square corners on the chessboard

Realization in OpenCV

Algorithm has found object position

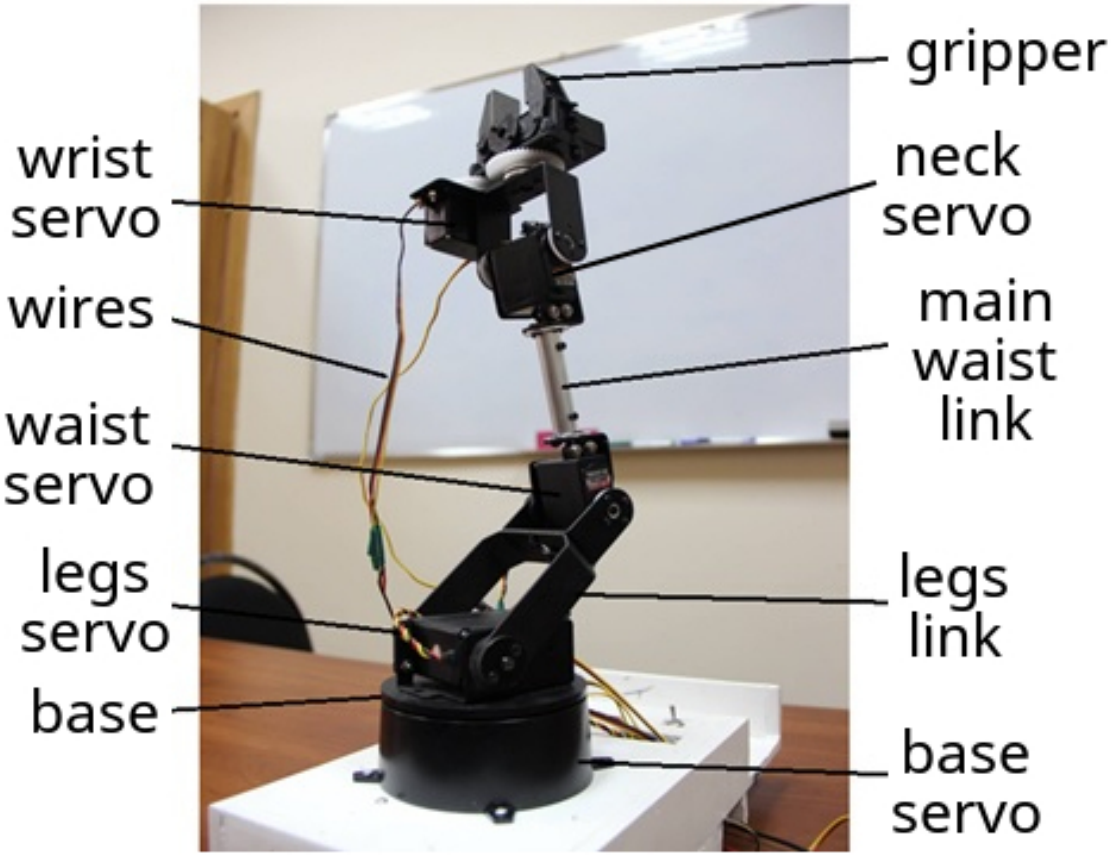
Kinematics

So far the following is known:

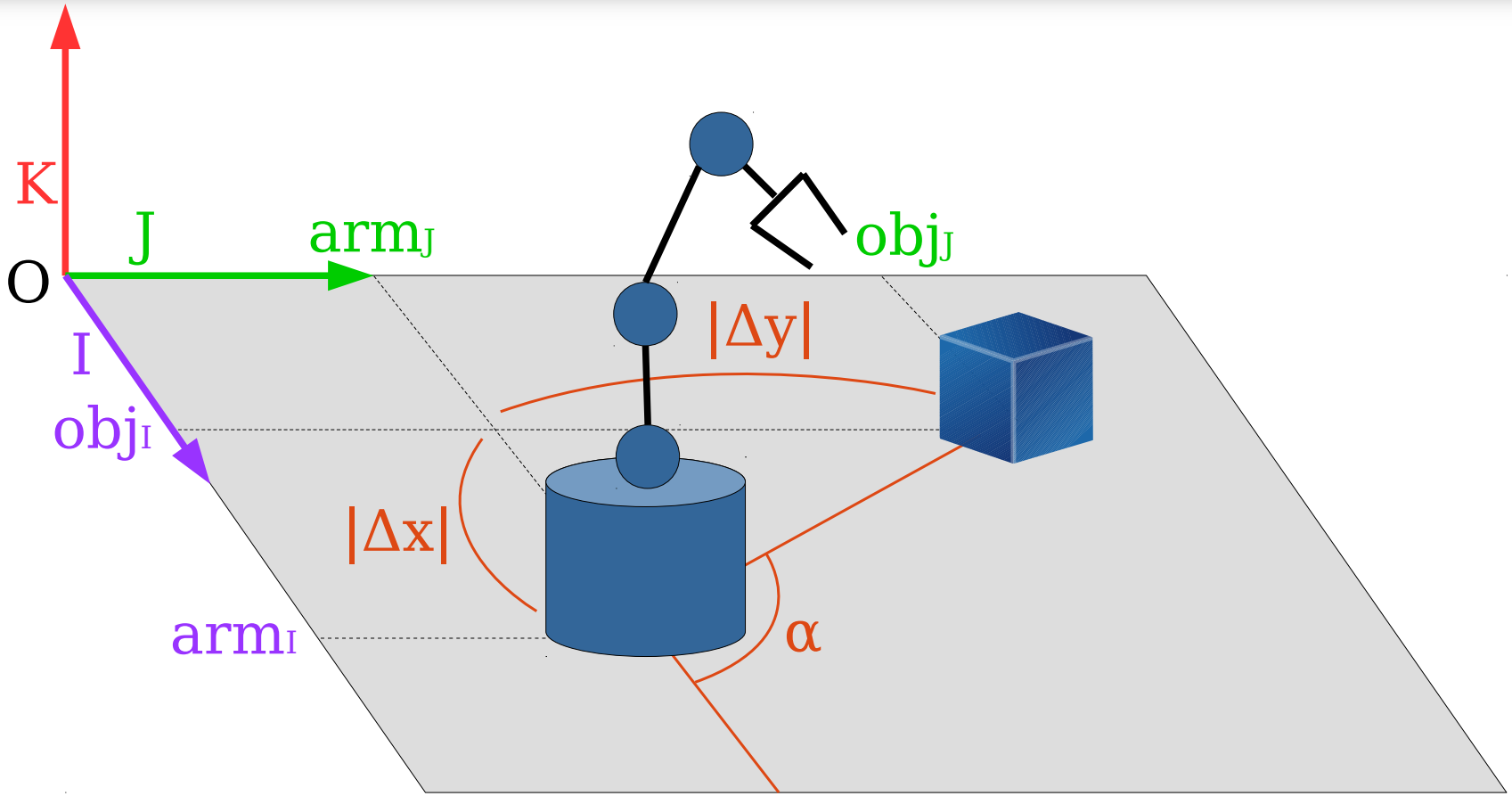
- Object coordinates (M_I, M_J);
- Coordinates of the robotic arm;
- Parameters of the robotic arm.

We have to compute:

- Respective rotation angles of the servo motors disposed at the joints, which would ensure precise positioning the end-effector (gripper) so that the object could be reached.



Kinematics

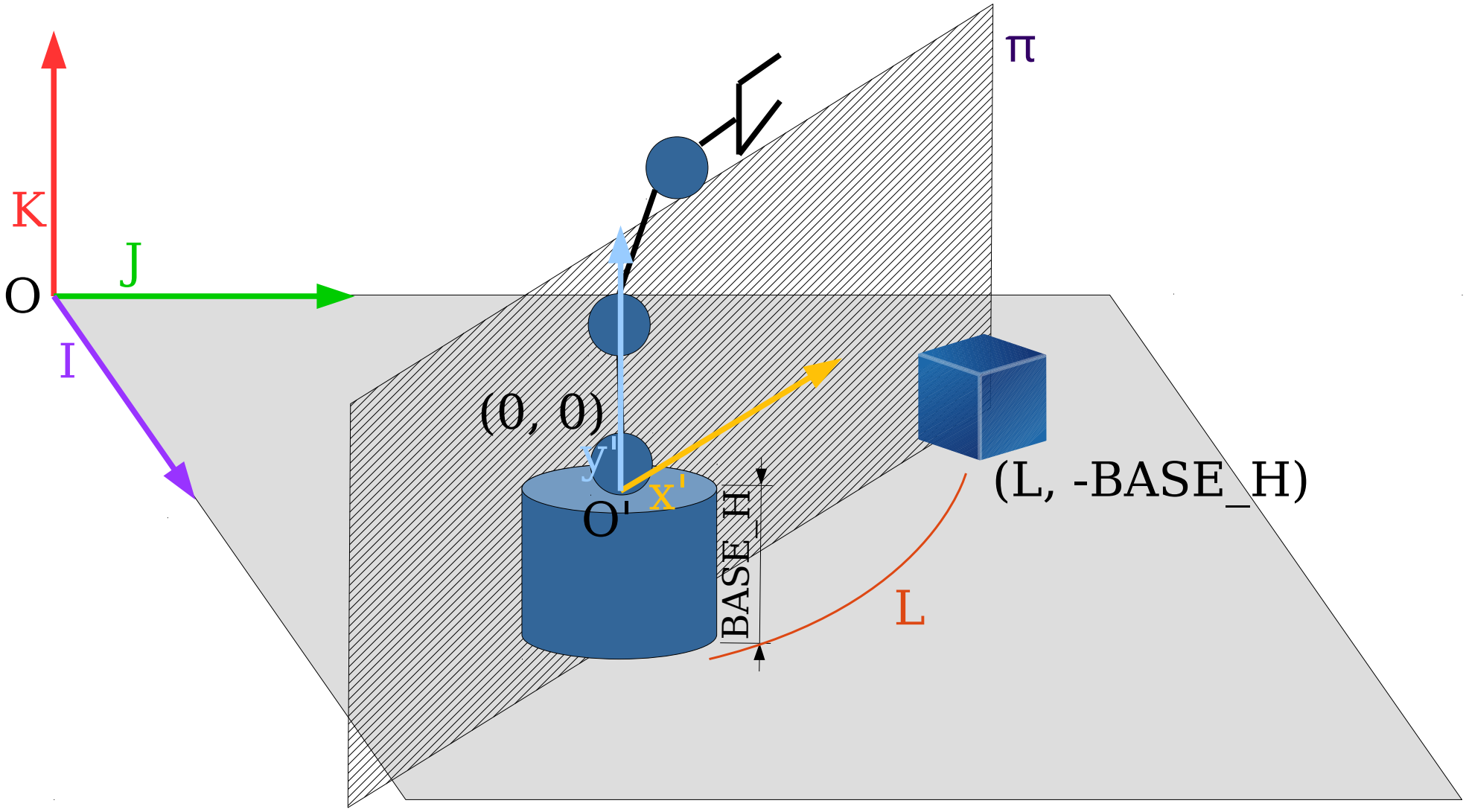


$$\Delta x = obj_I - arm_I; \quad \Delta y = obj_J - arm_J \tag{19}$$

$$L = \sqrt{\Delta x^2 + \Delta y^2}; \quad \alpha_d = \arccos\frac{\Delta x}{L} rad \tag{20}$$

$$\alpha = \alpha_d \frac{180^o}{\pi rad} \tag{21}$$

Kinematics



Problem passes from 3D space to 2D plane

Forward Kinematics

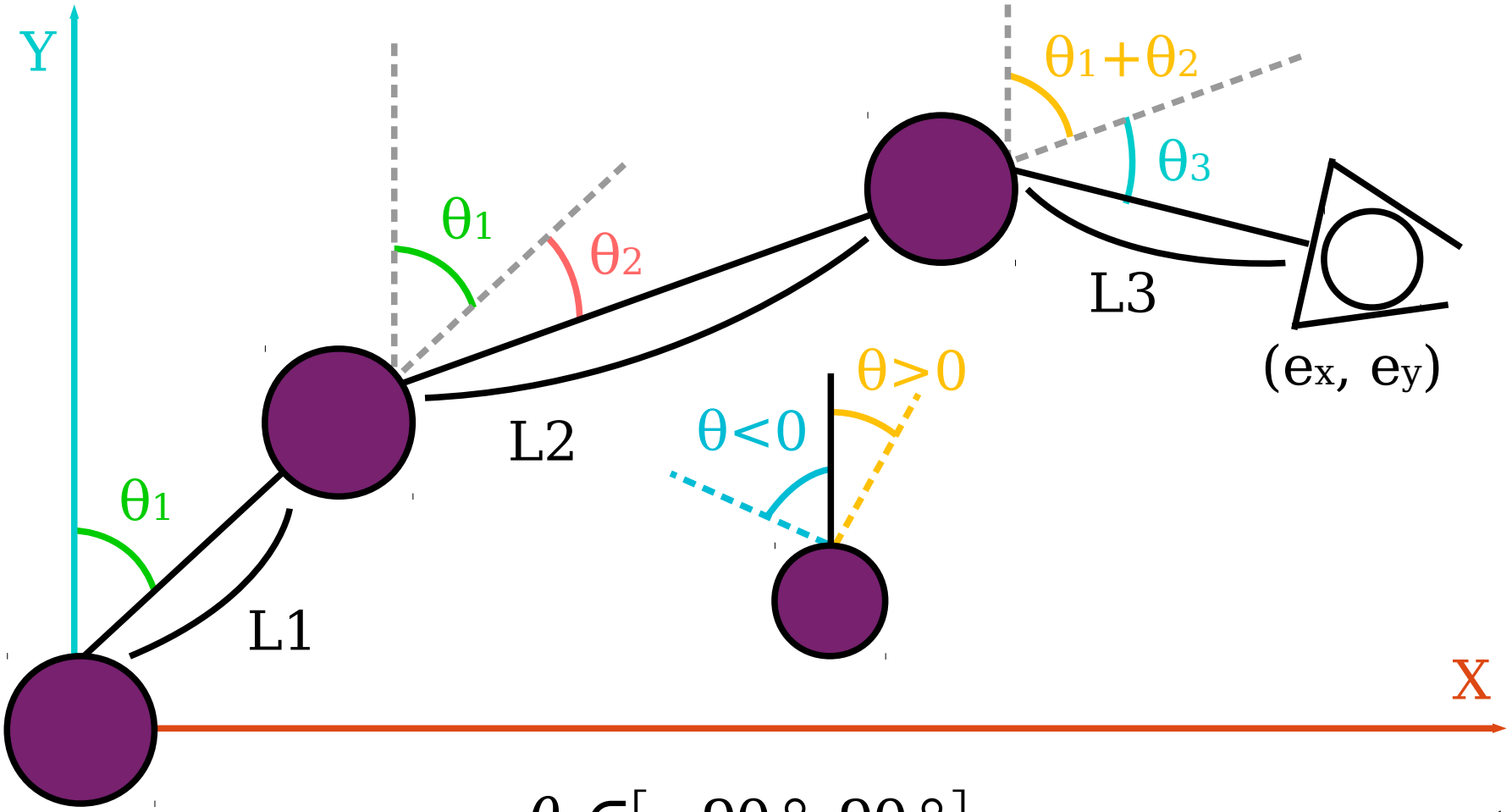
Forward kinematics describes the technique of determining the spatial location of the end-effector depending on the known lengths of links and positions of joints.

$$\theta \in \mathbb{R}^3; \quad e \in \mathbb{R}^2 \tag{22}$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}; \quad e = \begin{bmatrix} e_x \\ e_y \end{bmatrix} \tag{23}$$

$$f(\theta) = e \tag{24}$$

Forward Kinematics



$$\theta_i \in [-90°, 90°] \tag{25}$$

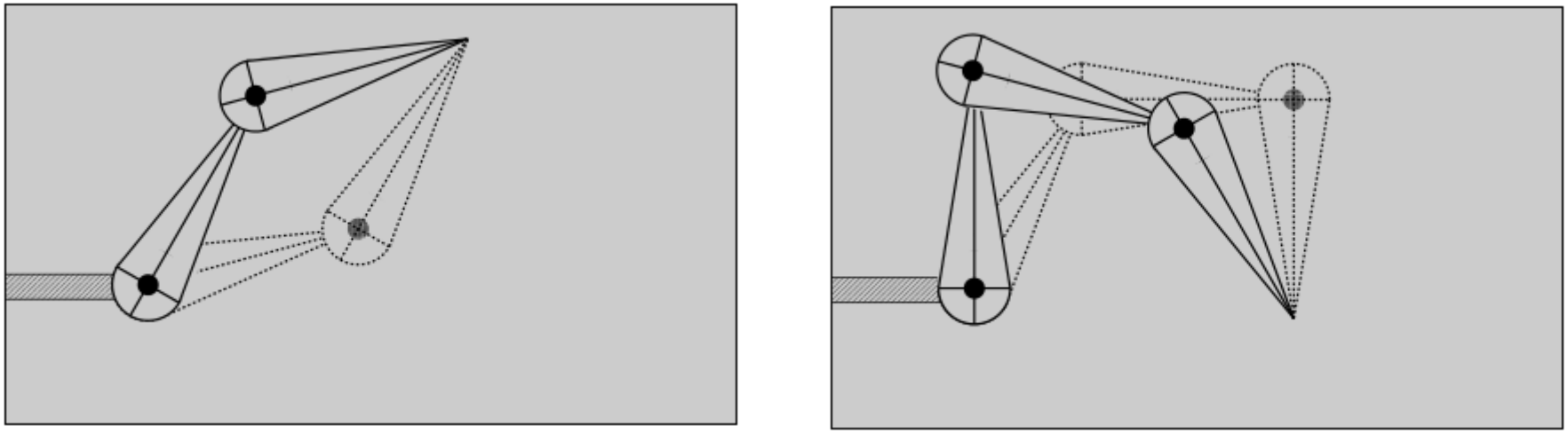
$$e_x = L_1 \sin\theta_1 + L_2 \sin\left(\theta_1+\theta_2\right) + L_3 \sin\left(\theta_1+\theta_2+\theta_3\right) \tag{26}$$

$$e_y = L_1 \cos\theta_1 + L_2 \cos\left(\theta_1+\theta_2\right) + L_3 \cos\left(\theta_1+\theta_2+\theta_3\right) \tag{27}$$

Inverse Kinematics

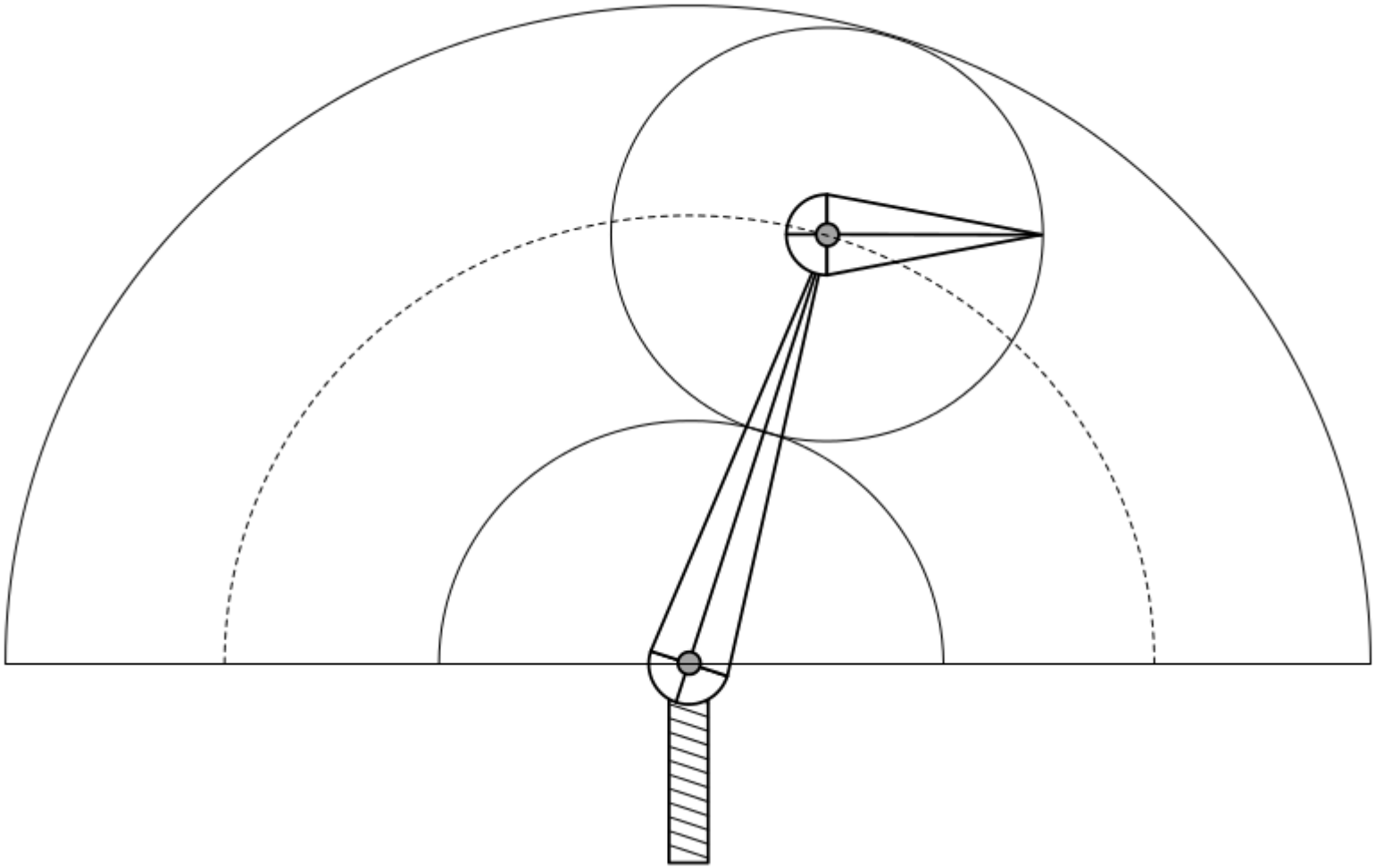
Inverse kinematics is a field of robotics whose subject is computing parameters of the joints sufficient to ensure reach of the end-effector to the predefined position.

$$\theta = f^{-1}(e) \tag{28}$$



Several possible solutions

Inverse Kinematics



Probable absence of solutions

Method of Jacobian Inverse

$$f:\mathbb{R}^3 \rightarrow \mathbb{R}^2;\quad e=f(\theta)\in\mathbb{R}^2;\quad \theta\in\mathbb{R}^3 \tag{29}$$

$$J(\theta)=\frac{de}{d\theta}=\begin{bmatrix}\frac{\partial e}{\partial \theta_1} & \frac{\partial e}{\partial \theta_2} & \frac{\partial e}{\partial \theta_3}\end{bmatrix}=\begin{bmatrix}\frac{\partial e_x}{\partial \theta_1} & \frac{\partial e_x}{\partial \theta_2} & \frac{\partial e_x}{\partial \theta_3}\\ \frac{\partial e_y}{\partial \theta_1} & \frac{\partial e_y}{\partial \theta_2} & \frac{\partial e_y}{\partial \theta_3}\end{bmatrix} \tag{30}$$

$$de=J(\theta)d\theta \tag{31}$$

$$de=e_{current}-e_{goal} \tag{32}$$

$$\theta_{i+1}=\theta_i+d\theta \tag{33}$$

$$d\theta=J^{-1}(\theta)de \tag{34}$$

$$\theta_{i+1}=\theta_i+J^{-1}(\theta)de \tag{35}$$

Method of Jacobian Inverse

$$de = J\, d\theta \tag{36}$$

$$J^T de = J^T J\, d\theta \tag{37}$$

$$\left(J^T J\right)^{-1} J^T de = \left(J^T J\right)^{-1}\left(J^T J\right) d\theta \tag{38}$$

$$\left(J^T J\right)^{-1} J^T de = d\theta \tag{39}$$

$$J^{pi} = \left(J^T J\right)^{-1} J^T \tag{40}$$

$$J_{11} = \frac{\partial e_x}{\partial \theta_1} = L_1 \cos\theta_1 + L_2 \cos\left(\theta_1 + \theta_2\right) + L_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right)$$

$$J_{12} = \frac{\partial e_x}{\partial \theta_2} = L_2 \cos\left(\theta_1 + \theta_2\right) + L_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right)$$

$$J_{13} = \frac{\partial e_x}{\partial \theta_3} = L_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right)$$

$$J_{21} = \frac{\partial e_y}{\partial \theta_1} = -L_1 \sin\theta_1 - L_2 \sin\left(\theta_1 + \theta_2\right) - L_3 \sin\left(\theta_1 + \theta_2 + \theta_3\right)$$

$$J_{22} = \frac{\partial e_y}{\partial \theta_2} = -L_2 \sin\left(\theta_1 + \theta_2\right) - L_3 \sin\left(\theta_1 + \theta_2 + \theta_3\right)$$

$$J_{23} = \frac{\partial e_y}{\partial \theta_3} = -L_3 \sin\left(\theta_1 + \theta_2 + \theta_3\right)$$

Thanks for watching