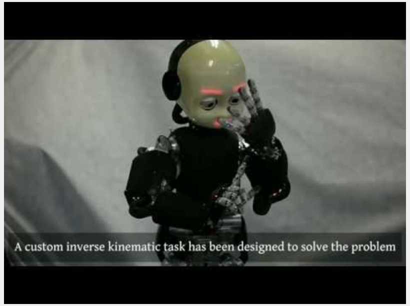
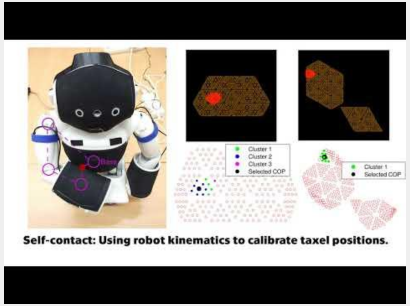
Self-contained Multisensorial Calibration

Lukas Rustler

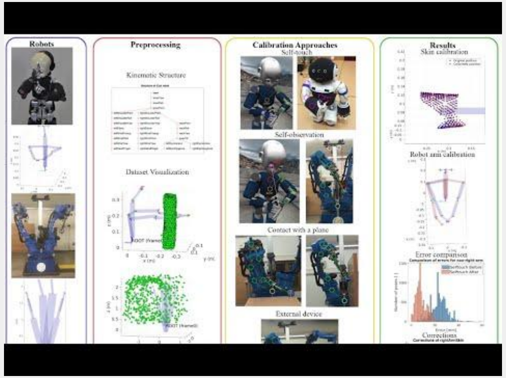
Calibration in our group



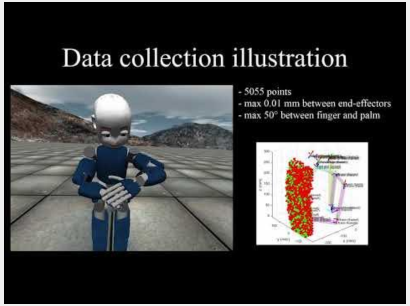
A. Roncone, M. Hoffmann, U. Pattacini and G. Metta, "Automatic kinematic chain calibration using artificial skin: Self-touch in the iCub humanoid robot," 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 2305-2312.



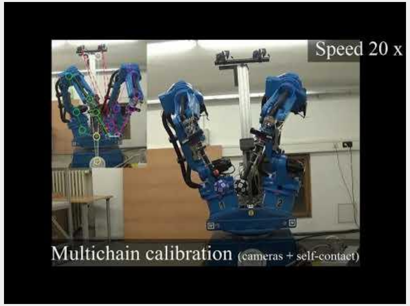
L. Rustler, B. Potocna, M. Polic, K. Stepanova and M. Hoffmann, "Spatial calibration of whole-body artificial skin on a humanoid robot: comparing self-contact, 3D reconstruction, and CAD-based calibration," 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), 2021, pp. 445-452.



J. Rozlivek, L. Rustler, K. Stepanova and M. Hoffmann, "Multisensorial robot calibration framework and toolbox," 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), 2021, pp. 459-466.



K. Stepanova, T. Pajdla and M. Hoffmann, "Robot Self-Calibration Using Multiple Kinematic Chains—A Simulation Study on the iCub Humanoid Robot," in IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1900-1907, April 2019.



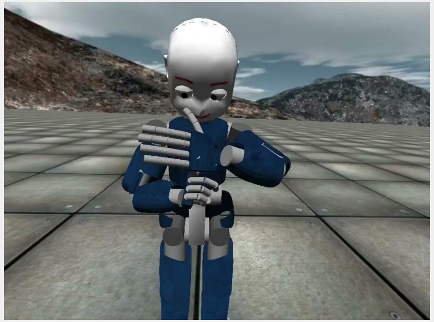
K. Stepanova, J. Rozlivek, F. Puciow, P. Krsek, T. Pajdla, M. Hoffmann, Automatic self-contained calibration of an industrial dual-arm robot with cameras using self-contact, planar constraints, and self-observation, Robotics and Computer-Integrated Manufacturing, Volume 73, 2022.

Why is calibration important?

- Parameters can change over time
  - wear and tear
  - encoder errors
  - accidents
- New parts can be added after manufacturing
  - replacements
  - new sensors
- New cheaper and lightweight robots are less accurate
  - however, provide rich set of sensors useful for self-contained calibration

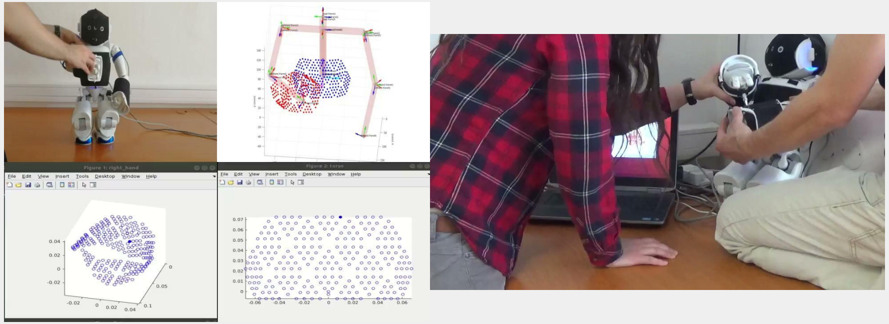
Calibration approaches

Planar constraints



Self-observation

External devices



Self-touch

Parameters of the links

- Estimation of parameter vector $\phi = \{\phi_k\}_{k \in K}$
  - where k are individual links expressed as:
    - Denavit-Hartenberg (DH) notation $\phi_k = \{[a_k, d_k, \alpha_k, o_k]\}$
      - a_k, d_k, α_k are the first three DH parameters and o_k is the offset of the encoders
    - or, translation and rotation vector $\phi_k = \{[\boldsymbol{t_k}, \boldsymbol{r_k}]\}$
      - $\boldsymbol{t_k}$ is three-dimensional vector of translation and $\boldsymbol{r_k}$ three-dimensional vector composed from unit axis of rotation $\boldsymbol{u_k} = \frac{\boldsymbol{r_k}}{||\boldsymbol{r_k}||}$ and rotation angle $\Theta_k = ||\boldsymbol{r_k}||$

Datasets

- Each dataset point $\boldsymbol{D}_i$ from dataset $\boldsymbol{D}$ is defined as $\boldsymbol{D}_i = [pn_i, \boldsymbol{\theta}_i, \boldsymbol{cp}_i, \boldsymbol{rp}_i, \boldsymbol{c}_i]$
  - where pn_i is a pose number, $\boldsymbol{\theta}_i$ are current joint angles, $\boldsymbol{cp}_i$ are the contact points, $\boldsymbol{rp}_i$ are the reference points and $\boldsymbol{c}_i$ are indexes of used cameras
- Each calibration approach has a unique dataset and the datasets are united to one $\boldsymbol{D}^{whole} = \{\boldsymbol{D}^{st}, \boldsymbol{D}^{pl}, \boldsymbol{D}^{so}, \boldsymbol{D}^{ed}\}$
  - where **st** stands for self-touch, **pl** for plane contact, **so** for self-observation and **ed** for external devices
- Usually two points $\boldsymbol{x}^A, \boldsymbol{x}^B$ from sets of points $\boldsymbol{X}^A, \boldsymbol{X}^B$

Parameters optimization

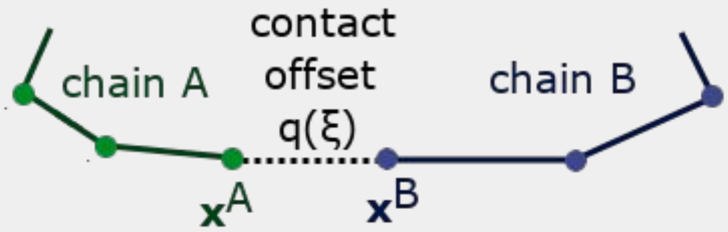
- Optimization of objective function as $\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\, \boldsymbol{f}(\boldsymbol{\phi}, \boldsymbol{D}, \boldsymbol{\zeta})$
    - where $\boldsymbol{f}(\boldsymbol{\phi}, \boldsymbol{D}, \boldsymbol{\zeta}) = \|\boldsymbol{g}(\boldsymbol{\phi}, \boldsymbol{D}, \boldsymbol{\zeta})\|^2 = \sum_{i=1}^{M} g(\boldsymbol{\phi}, \boldsymbol{D}_i, \boldsymbol{\zeta})^2$
        - $\boldsymbol{g}$ is defined for each approach, M is number of configurations (poses) and $\boldsymbol{\zeta}$ are other parameters (fixed transformation, camera calibration, etc.)
- Individual functions $\boldsymbol{g}$ can be united into the overall function

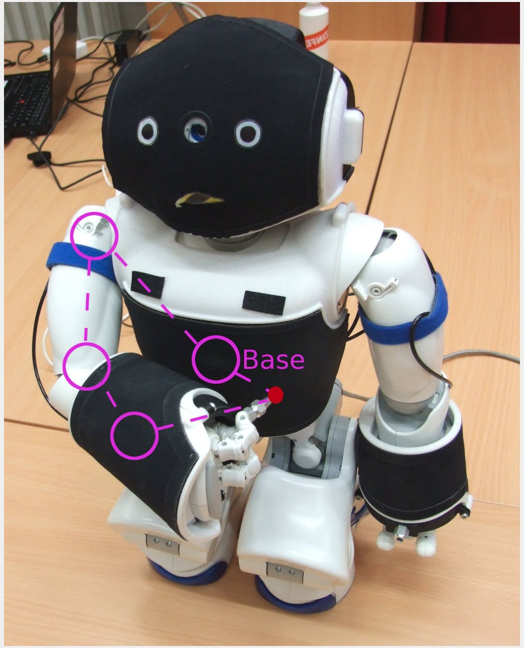
$$\boldsymbol{g}(\boldsymbol{\phi}, D, \boldsymbol{\zeta}) = \Big[\boldsymbol{k}^{st} \odot \boldsymbol{g}^{st}(\boldsymbol{\phi}, \boldsymbol{D}^{st}, \boldsymbol{\zeta}), \boldsymbol{k}^{p} \odot \boldsymbol{g}^{p}(\boldsymbol{\phi}, \boldsymbol{D}^{p}, \boldsymbol{\zeta}),$$
$$\boldsymbol{k}^{so} \odot \boldsymbol{g}^{so}(\boldsymbol{\phi}, \boldsymbol{D}^{so}, \boldsymbol{\zeta}), \boldsymbol{k}^{ed} \odot \boldsymbol{g}^{ed}(\boldsymbol{\phi}, \boldsymbol{D}^{ed}, \boldsymbol{\zeta})\Big],$$

    - where $\odot$ marks the Hadamard product (e.g., $(\boldsymbol{k}^{st} \odot \boldsymbol{g}^{st})_i = k_i^{st} \cdot g_i^{st}$), $\boldsymbol{k}^j$ ($j \in \{st, p, so, ed\}$) are the scale factors allowing combination of approaches

Self-touch

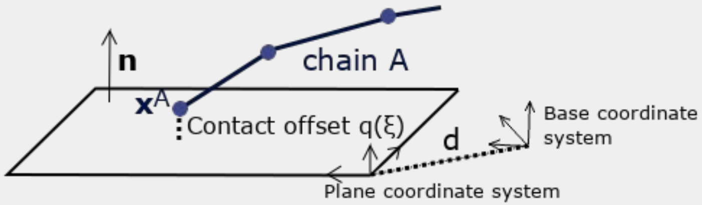
- Chains *A, B*
  - Parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}^A, \boldsymbol{\phi}^B\}$
  - minimization of the distance between two points $\boldsymbol{x}^A, \boldsymbol{x}^B$
    - tactile sensors, fingers, tool, etc.
- Function $\boldsymbol{g}^{st}(\boldsymbol{\phi}, \boldsymbol{D}^{st}, \zeta) = \quad [c(\boldsymbol{\phi}, \boldsymbol{D}_1, \zeta) - q(\zeta), ..., c(\boldsymbol{\phi}, \boldsymbol{D}_M, \zeta) - q(\zeta)]$
  - where $c(\boldsymbol{\phi}, \boldsymbol{D}_i, \zeta) = ||\boldsymbol{X}_i^A(\boldsymbol{\phi}^A, \boldsymbol{D}_i, \zeta) - \boldsymbol{X}_i^B(\boldsymbol{\phi}^B, \boldsymbol{D}_i, \zeta)||$ is a distance between $\boldsymbol{x}^A, \boldsymbol{x}^B$ and $q(\zeta)$ is the contact offset (e.g., thickness of the skin)

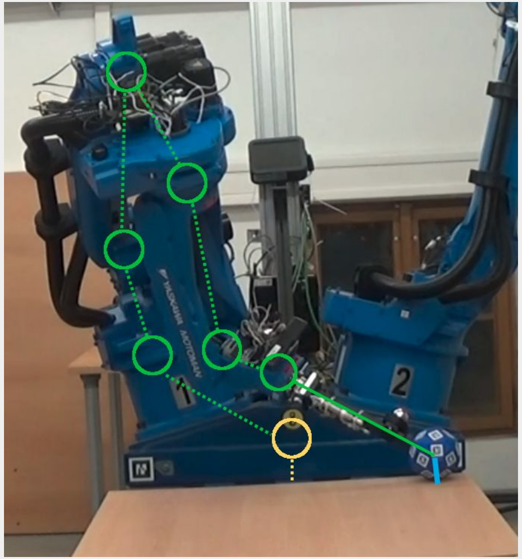




Plane constraints

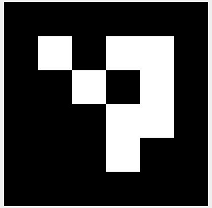
- Chain *A* and plane *p*
  - parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}^A, \mathbf{n}, d\}$
    - $\mathbf{n} = (a, b, c)$ is a normal vector from plane equation $ax + by + cz + d = 0$
  - minimization of the distance between end-effector $\boldsymbol{x}^A$ and a plane
- Function $\boldsymbol{g}^p(\boldsymbol{\phi}, \boldsymbol{D}^p, \boldsymbol{\zeta}) = [\boldsymbol{c}(\boldsymbol{\phi}^{p_1}, \boldsymbol{D}^{p_1}, \boldsymbol{\zeta}) - q(\boldsymbol{\zeta}), ..., \boldsymbol{c}(\boldsymbol{\phi}^{p_n}, \boldsymbol{D}^{p_n}, \boldsymbol{\zeta}) - q(\boldsymbol{\zeta})]$
  - where $c(\boldsymbol{\phi}^{p_j,A}, \boldsymbol{D}_i^{p_j}, \boldsymbol{\zeta}) = ||\boldsymbol{n}^{p_j} \boldsymbol{p}_i^{p_j}(\boldsymbol{\phi}^{p_j,A}) + d^{p_j}||$ is the distance for plane *j* and $q(\boldsymbol{\zeta})$ is the contact offset

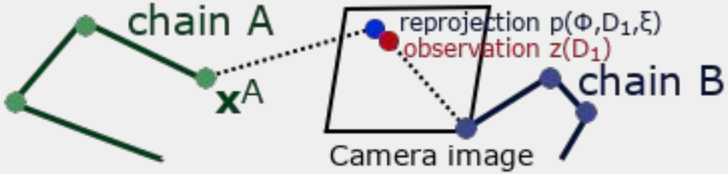


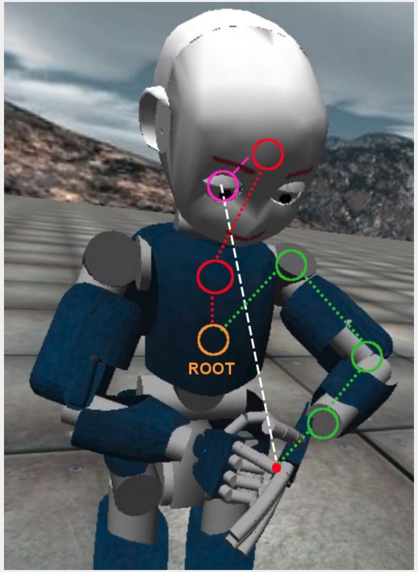


Self-observation

- Chain *A* (observed) and chain *B* (camera)
  - parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}^A, \boldsymbol{\phi}^B\}$
  - minimization of projection of point $\boldsymbol{x}^A$ into the camera plane and point $\boldsymbol{x}^B$ in the camera image
- Function $\boldsymbol{g}^{so}(\boldsymbol{\phi}, \boldsymbol{D}^{so}, \boldsymbol{\zeta}) = \quad [p(\boldsymbol{\phi}, \boldsymbol{D}_1, \boldsymbol{\zeta}) - z(\boldsymbol{D}_1), ..., p(\boldsymbol{\phi}, \boldsymbol{D}_{M'}, \boldsymbol{\zeta}) - z(\boldsymbol{D}_{M'})]$
  - where $p(\boldsymbol{\phi}, \boldsymbol{D}_i, \boldsymbol{\zeta})$ is the projection and $z(\boldsymbol{D}_i)$ is the actual observed point in the camera image
    - the observed point in the image is found using ArUco markers
    - the projection is done by using pinhole camera model

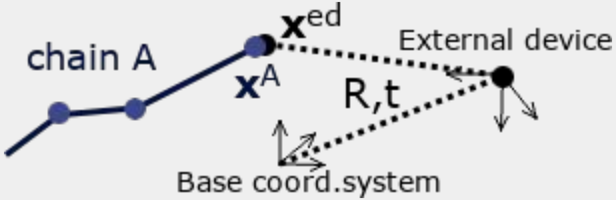


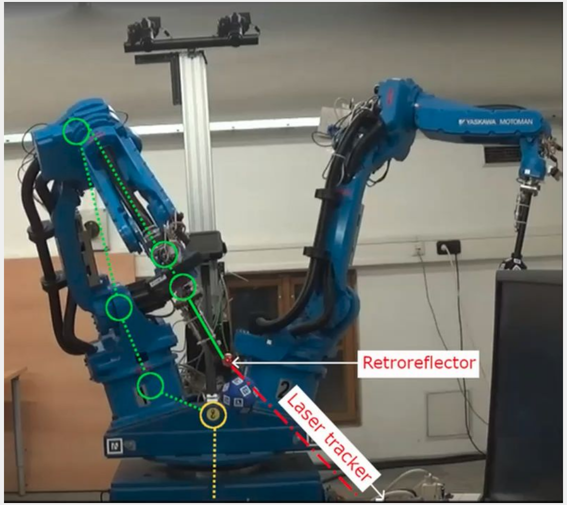




External devices

- Chain *A*
  - parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}^A, \boldsymbol{R}, \boldsymbol{t}\}$
    - **R, t** are the rotation matrix and translation vector of the external device w.r.t. The robot base frame
      - **R, t** can be found with SVD (*Arun et al. 1987*)
  - Minimization of the distance between $\boldsymbol{x}^A, \boldsymbol{x}^{ed}$
    - retroreflector on the robot
- Function $\boldsymbol{g}^{ed}(\boldsymbol{\phi}, \boldsymbol{D}^{ed}, \zeta) = \left[p(\boldsymbol{\phi}, \boldsymbol{D}_1, \zeta), ..., \; p(\boldsymbol{\phi}, \boldsymbol{D}_M, \zeta)\right]$
  - where $p(\boldsymbol{\phi}, \boldsymbol{D}_i, \zeta) = ||\boldsymbol{X}_i^{ed} - \boldsymbol{X}_i^A||$ is the distance between $\boldsymbol{x}^A, \boldsymbol{x}^{ed}$
  - $\boldsymbol{x}^{ed}$ is usually acquired with laser measuring device





Minimization

- Non-linear least squares
  - sines and cosines
  - solved by numerical methods
    - Gauss-Newton, Levenberg-Marquardt, Trust Region Reflective
- Gauss-Newton
  - $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$, $\boldsymbol{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$
    - *m* equations and *n* variables, no solution
    - criterion function $f(\boldsymbol{x}) = ||\boldsymbol{g}(\boldsymbol{x})||^2 = \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{m} g_i(\boldsymbol{x})^2$
  - $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{g}'(\boldsymbol{x}_k)^+ \boldsymbol{g}(\boldsymbol{x}_k) = \boldsymbol{x}_k - \left(\boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}'(\boldsymbol{x}_k)\right)^{-1} \boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}(\boldsymbol{x}_k)$
- Levenberg-Marquardt
  - $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left(\boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}'(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}\right)^{-1} \boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}(\boldsymbol{x}_k)$

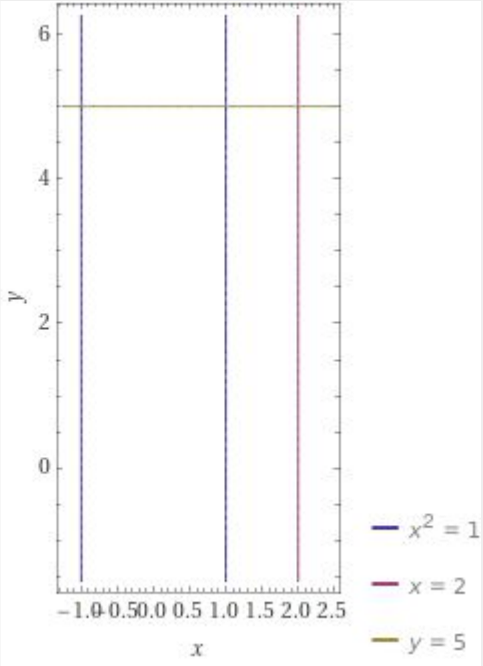
Things to avoid

- Wrong dataset
    - bad distribution of training examples
        - check joint distribution plots
        - check identifiability of parameters
            - computed from identification matrix
            - the closer the values to zero, the worse identifiability -> the parameter cannot be calibrated
        - (Optional) observability may be checked as well
- Compensation
    - if the problem is badly scaled, individual links may compensate
        - two link are shifted for few centimeters, but the calibration will move only one for the sum of the errors
        - can be solved by using bounds

Levenberg-Marquardt

- $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left(\boldsymbol{g}^{'}(\boldsymbol{x}_k)^T \boldsymbol{g}^{'}(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}\right)^{-1} \boldsymbol{g}^{'}(\boldsymbol{x}_k)^T \boldsymbol{g}(\boldsymbol{x}_k)$
    - $\mu_k \boldsymbol{I}$ can be seen as regularization
    - μ_k should get smaller when criterion decreases and higher otherwise
        - with multiples of 10 usually
- Exercise: Compute one iteration of LM algorithm
    - System of equations $x^2 = 1$
      $x = 2$
      $y = 5$

1. Define the cost function
2. Write matrix **g** and its derivative
3. Compute $\boldsymbol{x}_1$ and μ_1 for $\boldsymbol{x_0} = (0,0)$ and $\mu_0 = 10$



Levenberg-Marquardt

- Exercise: Compute one iteration of LM algorithm
  - System of equations

$$\begin{aligned} x^2 &= 1 \\ x &= 2 \\ y &= 5 \end{aligned}$$

1. Define the cost function

$$f(x,y) = \boldsymbol{g}(x,y)^T \boldsymbol{g}(x,y)$$

Levenberg-Marquardt

- Exercise: Compute one iteration of LM algorithm
  - System of equations $$x^2 = 1$$ $$x = 2$$ $$y = 5$$

2. Write matrix **g** and its derivative

$$\boldsymbol{g} = \begin{bmatrix} x^2 - 1 \\ x - 2 \\ y - 5 \end{bmatrix} \qquad \boldsymbol{g}^{'} = \begin{bmatrix} 2x & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Levenberg-Marquardt

- Exercise: Compute one iteration of LM algorithm
  - System of equations

$$x^2 = 1$$
$$x = 2$$
$$y = 5$$

$$\boldsymbol{g} = \begin{bmatrix} x^2 - 1 \\ x - 2 \\ y - 5 \end{bmatrix}$$

3. Compute $\boldsymbol{x}_1$ and μ_1 for $\boldsymbol{x_0} = (0,0)$ and $\mu_0 = 10$

$$\boldsymbol{g}' = \begin{bmatrix} 2x & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left(\boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}'(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}\right)^{-1} \boldsymbol{g}'(\boldsymbol{x}_k)^T \boldsymbol{g}(\boldsymbol{x}_k)$$

$$\boldsymbol{x_1} = (0.1818, 0.4545)$$

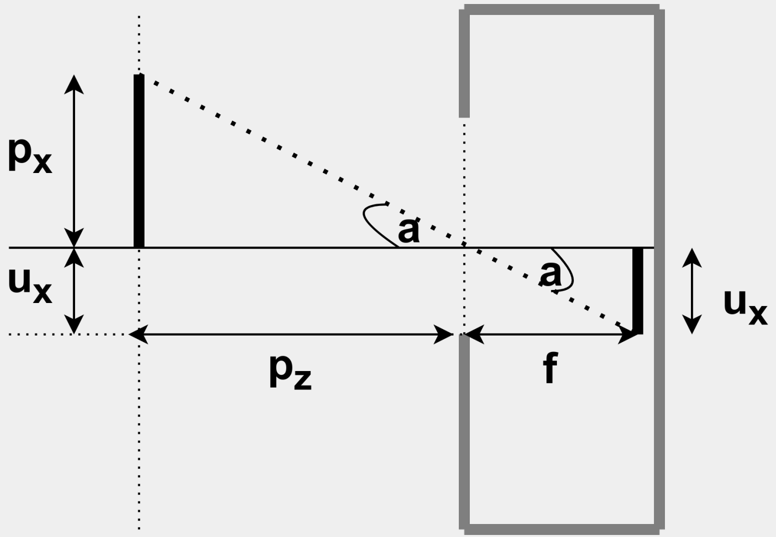
$$f(x_0, y_0) = f(0,0) = 30$$

$$f(x_1, y_1) = f(0.1818, 0.4545) = 24.9019$$

$\longrightarrow$ e.g., $\mu_1 = 1$

Pinhole camera model

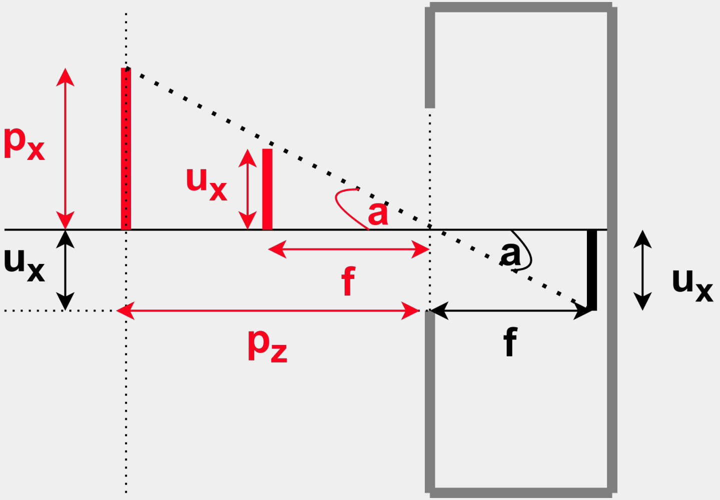
Exercise: Derive the pinhole camera model



Pinhole camera model

Exercise: Derive the pinhole camera model

$$tg(a) = \frac{p_x}{p_z} = \frac{u_x}{f} \rightarrow u_x = f\frac{p_x}{p_z}$$



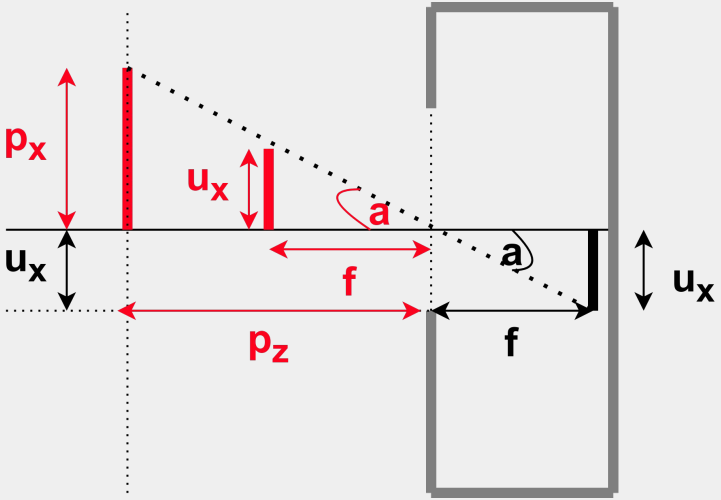
Pinhole camera model

Exercise: Derive the pinhole camera model

$$u_x = f\frac{p_x}{p_z} \qquad\qquad \lambda u_x = f p_x$$

$$\longrightarrow \qquad \lambda u_y = f p_y$$

$$u_y = f\frac{p_y}{p_z} \qquad\qquad \lambda = p_z$$



Pinhole camera model

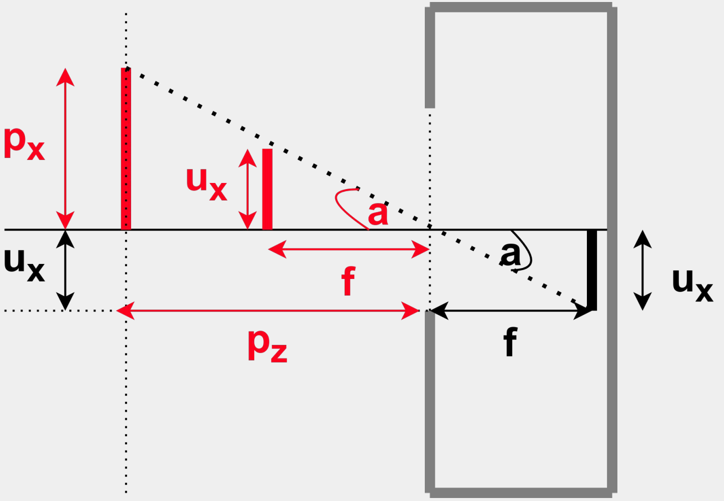
Exercise: Derive the pinhole camera model

$$u_x = f\frac{p_x}{p_z} \qquad \lambda u_x = f p_x$$

$$u_y = f\frac{p_y}{p_z} \quad \rightarrow \quad \lambda u_y = f p_y$$

$$\lambda = p_z$$

$$\lambda \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$



Pinhole camera model

Exercise: Derive the pinhole camera model

$$u_x = o_x + s_x f \frac{p_x}{p_z}$$

$$u_y = o_y + s_y f \frac{p_y}{p_z}$$

$$\lambda \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & o_x \\ 0 & s_y f & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$