上海科技大学
ShanghaiTech University

# CS289: **Mobile Manipulation Fall 2024**

Sören Schwertfeger

ShanghaiTech University

# CALIBRATION

# Calibration

- Intrinsic calibration:
  - Correct raw sensor data such that:
    - It adheres to certain standards
    - Reduces the error/ noise
  - Robotics/ autonomous driving:
    - Camera calibration!
    - LiDARS can also be calibrated (factory calibration typically sufficient)
    - All sensors …
- Extrinsic calibration:
  - Determine the pose (position and orientation) of sensors w.r.t. a frame of reference

# Camera Calibration Parameters

- Different models possible – typical one:
- $\alpha_u$, $\alpha_v$: focal length and size of pixel
- $u_0$, $v_0$: position of the sensor w.r.t optical center
- $k_1$: radial distortion

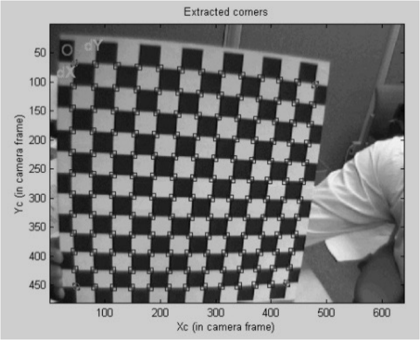
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$

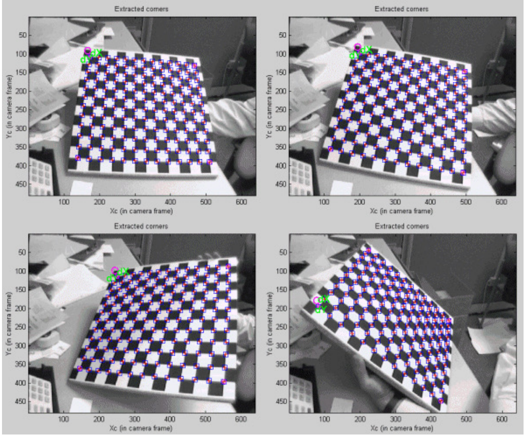
$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 \rho^2) \cdot \begin{bmatrix} u \\ v \end{bmatrix}$$

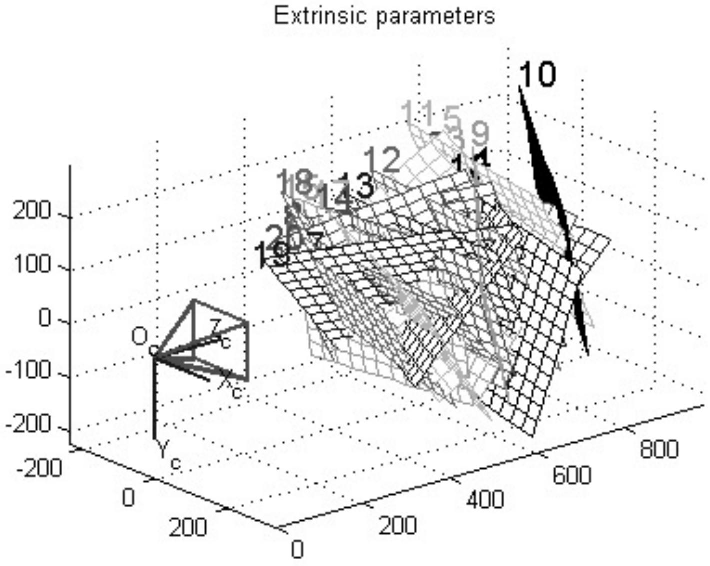
# Camera Calibration

- Parameters: govern mapping from scene points to image points
- Idea: known:
  - pixel coordinates of image points p
  - 3D coordinates of the corresponding scene points P
  - => compute the unknown parameters A, R, T by solving the perspective projection equation

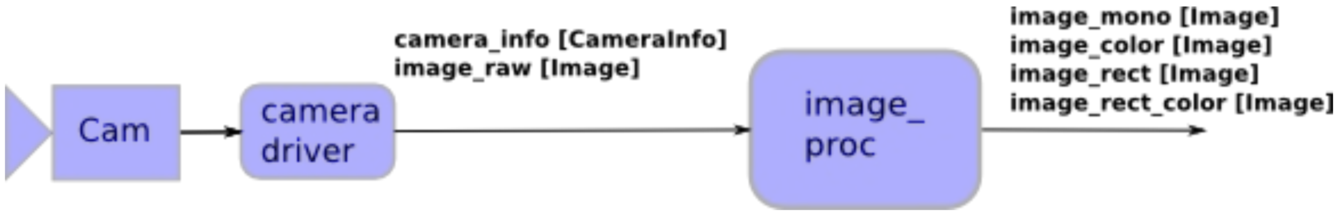
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$



# ROS Image Processing




- image_proc : Image processing node
  - Take calibration data from sensor_msgs/CameraInfo to calibrate/ rectify images
  - http://wiki.ros.org/image_proc

sensor_msgs/CameraInfo

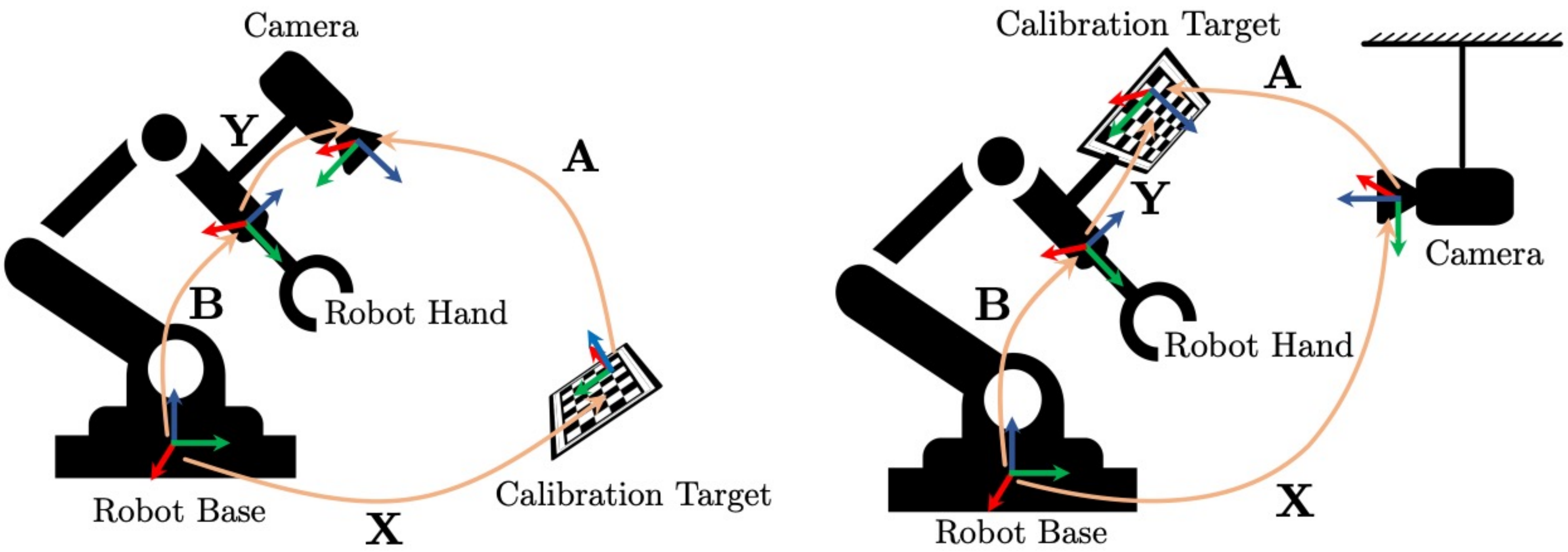
std_msgs/Header header
uint32 height
uint32 width
string distortion_model
float64[] D
float64[9] K
float64[9] R
float64[12] P
uint32 binning_x
uint32 binning_y
sensor_msgs/RegionOfInterest roi

- Calibrate cameras:
  - http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration
  - http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration

# Extrinsic Calibration

- Find pose of sensor w.r.t:
  - another sensor of same type
    - registration/ scan matching
  - another sensor of a different type
    - Heterogeneous calibration (e.g. LiDAR, camera, IMU)
  - a robot/ the arm frame – arm coordinate system
    - Hand-eye calibration
  - many other sensors:
    - Multi-sensor calibration

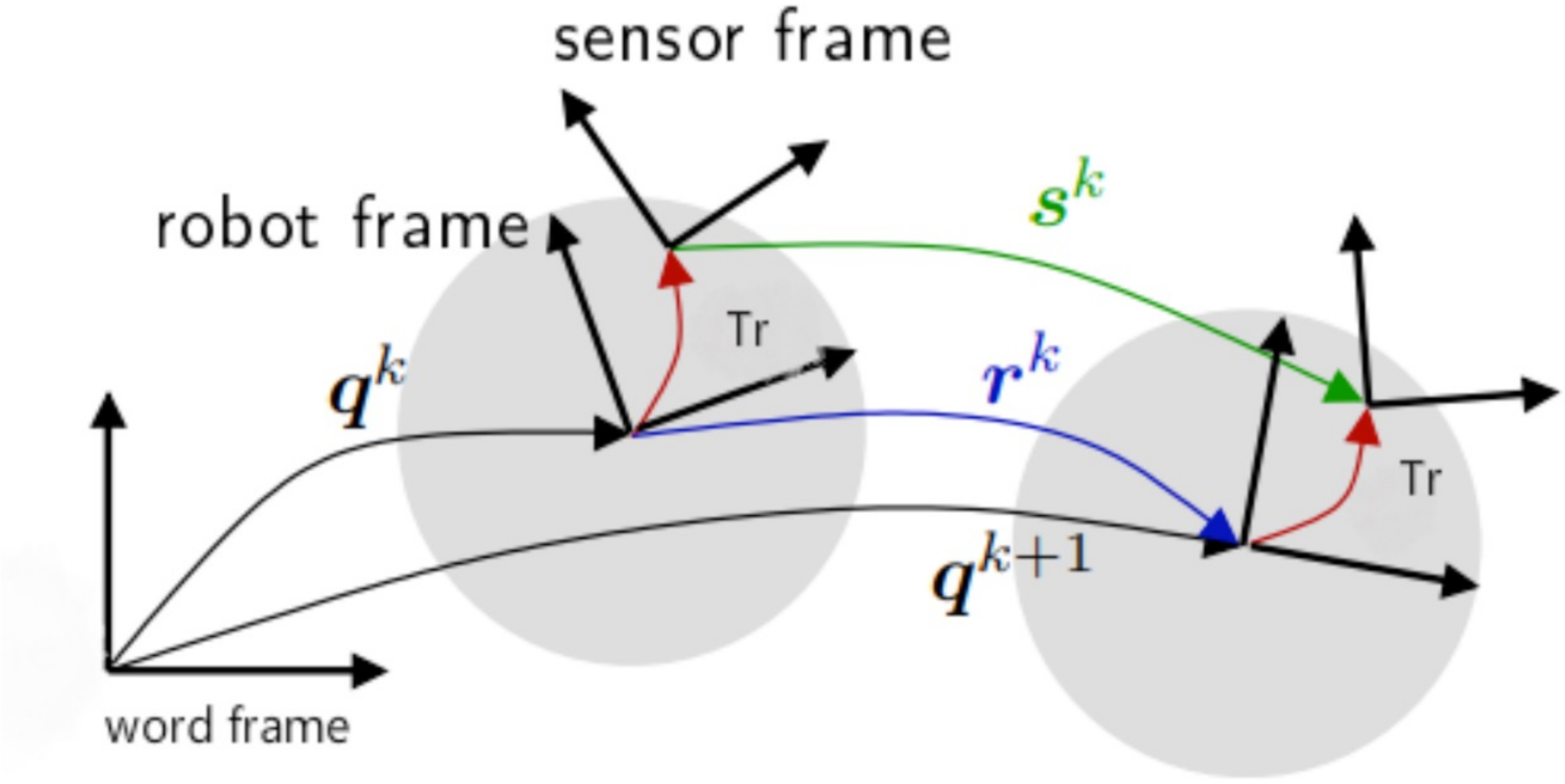
# Manipulation: Hand-Eye-Calibration



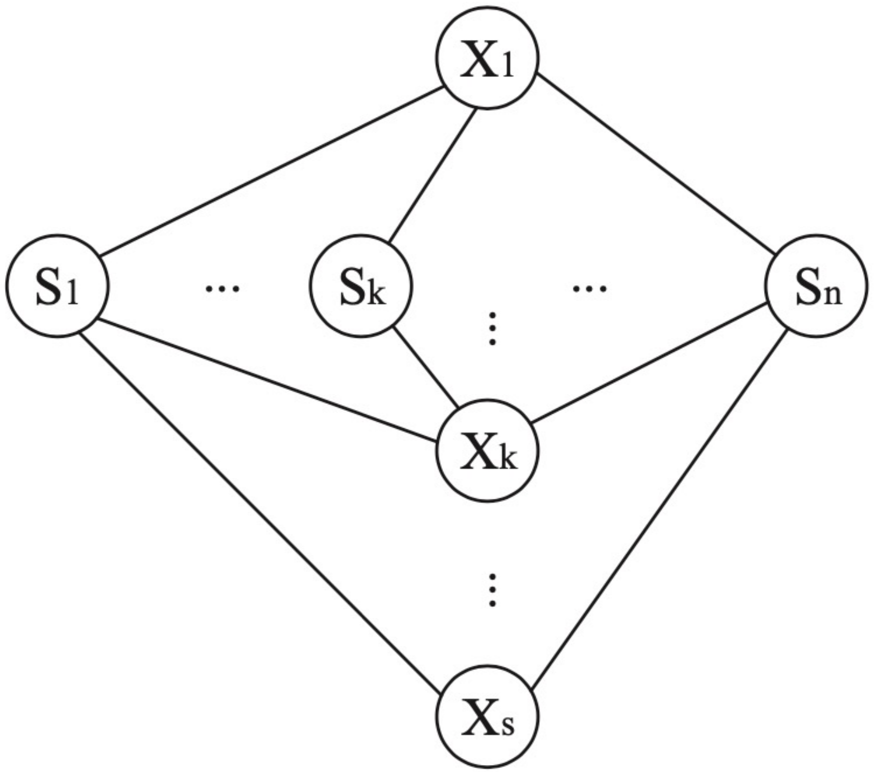

(a) eye-on-hand; aka: eye-in-hand (a) eye-to-base; aka: eye-to-hand

Fig. 2. Visualization of the hand-eye/robot-world calibration problem. Both (a) *eye-on-hand* and (b) *eye-to-base* cases are constrained by $\mathbf{AX} = \mathbf{YB}$.

# Hand-Eye calibration: robot to sensor transform



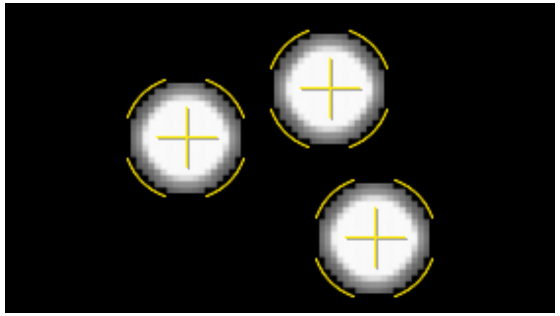
# Heterogeneous Multi-sensor Calibration based on Graph Optimization




**5.6 The graph representing the relationships between different sensors.** $X_1 \ldots X_s$ **and** $S_1 \ldots S_n$ **are different types of sensors such as camera and Velodyne. An edge between two nodes represents a direct sensor-to-sensor calibration between these two devices.**

# Segway: Tracking Systems


- Use cameras to detect infrared markers
- Triangulate 3D marker position from multiple cameras
- Create groups of markers to estimate 3D poses of groups
- Cameras need to be intrinsically & extrinsically calibrated
- Cameras report only marker (image) positions to system => one system can handle many cameras
- Famous vendors: Vicon, Optitrack
- E.g. Optitrack system at STAR Center:
  - 240Hz
  - Error: +-0.20 mm
  - 21 cameras



Video next slide: https://www.youtube.com/watch?v=o2xufLT8hUs
Video next next slide: https://www.youtube.com/watch?v=m0szHx3St-M

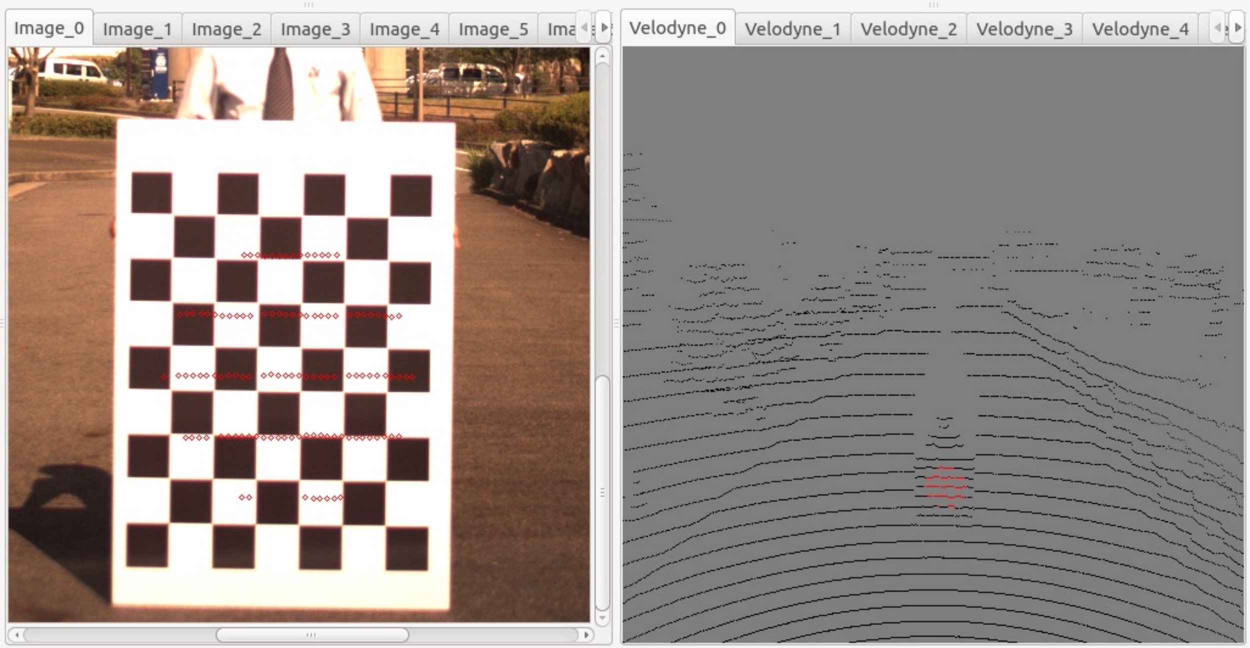
# Calibration

## Manipulator Calibration

- Move manipulator -> track "tool center point (tcp)" motion (or link to which camera is attached)
  - Use arm forward kinematics OR
  - Use tracking system. Difficulty: Calibration from tracking markers to tool needed
- Eye-on-hand:
  - Observe static target -> use Perspective-n-Point (PnP) to estimate camera transforms
- Eye-to-base
  - Static camera: observe moving target to estimate target transforms

## Multi-Sensor Calibration

- Robot with multiple sensors
- Option 1: Hand-eye approach:
  - Move robot (with sensors)
  - Estimate motions in all sensor frames
  - Use optimization to find transforms that explain all motions
- Option 2: Moving target approach:
  - Move a target around robot
  - Target needs to be observable by the sensors
  - Ideally: track target pose in tracking system
  - Use optimization to find transforms
  - Cannot estimate IMU pose




# Tools & Tutorials I

- Camera to Camera:
  - Stereo calibration (see previous)
- Camera to LiDAR:
  - https://github.com/CPFL/Autoware/wiki/Calibration
  - https://www.youtube.com/watch?v=pfBmfgHf6zg
  - Needs Autoware: https://github.com/cpfl/Autoware
- Point Cloud to Point Cloud (3D LiDAR):
  - Any point cloud registration, e.g. ICP
  - Example: Normal Distributions Transform https://pointclouds.org/documentation/tutorials/normal_distributions_transform.html

# Tools & Tutorials II

- Camera to IMU:
  - https://github.com/ethz-asl/kalibr
  - Offers multi-camera calibration; camera IMU calibration
  - Tutorial: https://github.com/ethz-asl/kalibr/wiki/Camera-IMU-calibration
  - Install: https://github.com/ethz-asl/kalibr/wiki/installation
- Hand-Eye Calibration:
  - E.g. Simultaneous Hand-Eye Calibration and Reconstruction
    - Doesn't use calibration targets
    - https://github.com/STAR-Center/shecar
  - Or: https://github.com/IFL-CAMP/easy_handeye
- Multi-sensor Calibration:
  - Optimize the results of a graph of individual calibrations…
  - "Heterogeneous Multi-sensor Calibration based on Graph Optimization" https://arxiv.org/pdf/1905.11167.pdf
  - **Multical: Spatiotemporal Calibration for Multiple IMUs, Cameras and LiDARs** https://github.com/zhixy/multical