A Machine Learning Approach to Hand-Eye Calibration

Nils, Nikolay, and Marc

Machine Learning & Robotics Group – TU Berlin

February 03, 2010

Outline

Introduction

Stand-alone Calibration: Robot Arm

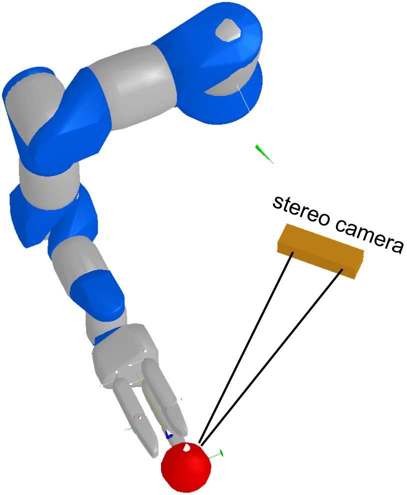
Stand-alone Calibration: Camera

Hand-Eye Calibration
- Calibration Setup
- Data Acquisition And Structure
- Minimization Problem

Next steps

Conclusions

Introduction



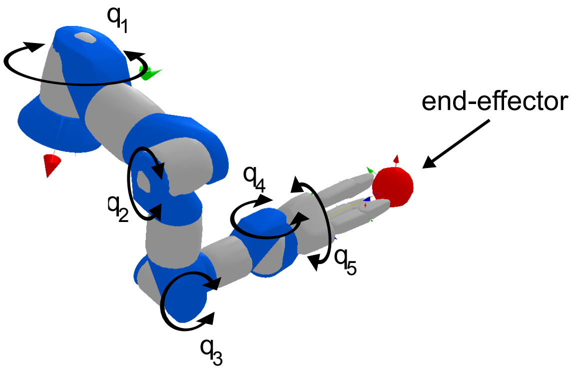
Introduction

- pre-step to high level algorithms, e.g. object recognition or grasping
- calibrate each piece of hardware by itself
- after calibration measure relative offset between arm and camera (tricky)
- non-linearities, e.g. through distortions
- our idea: reformulation of these problems as a single problem
- simultaneously learn:
  - mapping from image coordinates to world coordinates
  - mapping from state vector q to world coordinates
- minimize error between both mappings

Stand-alone Calibration: Robot Arm

Forward Kinematics

- n-dof robot arm:
  - controlled via state vector $\vec{q} \in \mathbb{R}^n$, e.g. joint angles
  - foward kinematics: given $\vec{q}$ determine world coord. $\vec{X}$ of end-effector
  - ideal world: $\vec{X} = \omega(\vec{q})$, actual world: $\vec{X}_\delta = \omega(\vec{q} + \vec{\delta})$
  - unknown $\vec{\delta}$ due to physical phenomena, e.g. cog wheels



Stand-alone Calibration: Camera

Pinhole Camera Model

- projection matrix $P \in \mathbb{R}^{3\times 4}$:

$$P = KR[I| - \tilde{C}] \quad (1)$$

- projection of world point $\vec{X} = (x, y, z, v)^t$ to image point $\vec{x} = (u, v, w)$:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = P \cdot \begin{pmatrix} x \\ y \\ z \\ v \end{pmatrix} \quad (2)$$

- intrinsics $K \in \mathbb{R}^{3\times 3}$, extrinsics $R \in \mathbb{R}^{3\times 3}$, $-\tilde{C} \in \mathbb{R}^3$:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Stand-alone Calibration: Camera

Gold Standard algorithm[1] to determine int./ext. parameters

- solvable through Direct Linear Transform(ation) (DLT)
- given: collect $n \geq 6$ point correspondances: $\vec{x}_i \leftrightarrow \vec{X}_i$
- wanted: matrix P, so that

$$\vec{x}_i = P\vec{X}_i \quad \Leftrightarrow \vec{x}_i \times (P\vec{X}_i) = \vec{0} \quad \Leftrightarrow \vec{x}_i \times \begin{pmatrix} P^1\vec{X}_i \\ P^2\vec{X}_i \\ P^3\vec{X}_i \end{pmatrix} = \vec{0} \tag{3}$$

- rewrite eq. (3) in matrix notation, $A \in \mathbb{R}^{12\times 12}$, $p \in \mathbb{R}^{12}$:

$$A\vec{p} = \vec{0}, \text{ s.t. } \|\vec{p}\| = 1 \tag{4}$$

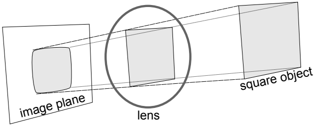
- solve with SVD: $A = UDV^T$
- $p = V(:, end)$, if diagonal values of D in descending order

[1] Zissermann, Multiple View Geometry

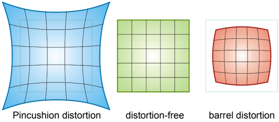
Stand-alone Calibration: Camera

Optical Aberration/Distortion

- camera with lens $\neq$ pin-hole camera model (!)
  - distortion (radial, tangential, ...)
  - significance increases while focal length (and price) decreases



(a) lens distortion



(b) types of (radial) distortion

Stand-alone Calibration: Camera

Optical Aberration/Distortion

- correction of radial distortion

$$\begin{aligned} x' &= x(1+k_1r^2+k_2r^4+k_3r^6) \\ y' &= y(1+k_1r^2+k_2r^4+k_3r^6) \end{aligned}$$

- correction of tangential distortion

$$\begin{aligned} x' &= x+[2p_1y+p_2(r^2+2x^2)] \\ y' &= y+[p_1(r^2+2y^2)+2p_2x] \end{aligned}$$

- minimizing geometric error of a calibration object (chessboard) based on deviation from linear mappings

Stand-alone Calibration: Camera

Stereo Calibration

- first, single-camera calibration to obtain

$$P_l = K_l R_l [I | -\tilde{C}_l]$$

and

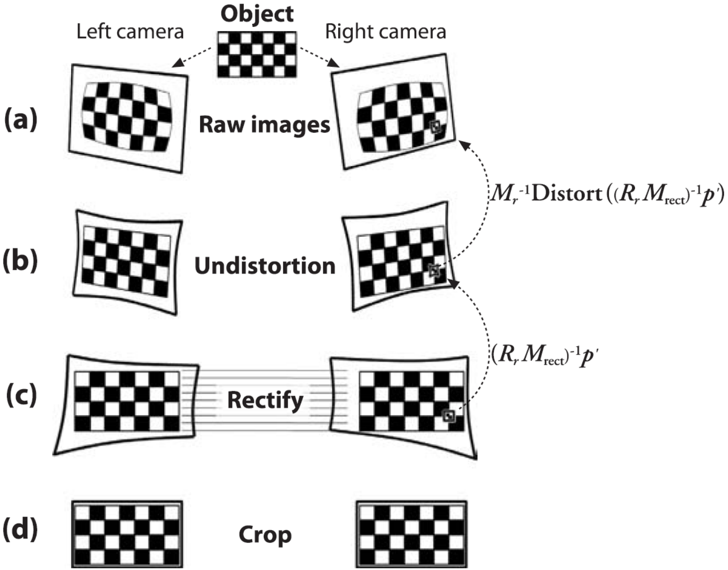
$$P_r = K_r R_r [I | -\tilde{C}_r]$$

- then, translation matrix R and translation T from right to left camera coordinate system:

$$\begin{aligned} R &= R_r (R_l)^T \\ T &= \tilde{C}_r - R\tilde{C}_l \end{aligned} \qquad (5)$$

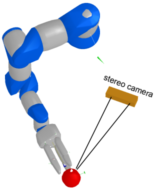
Stand-alone Calibration: Camera

Calibration Pipeline



Hand-Eye Calibration

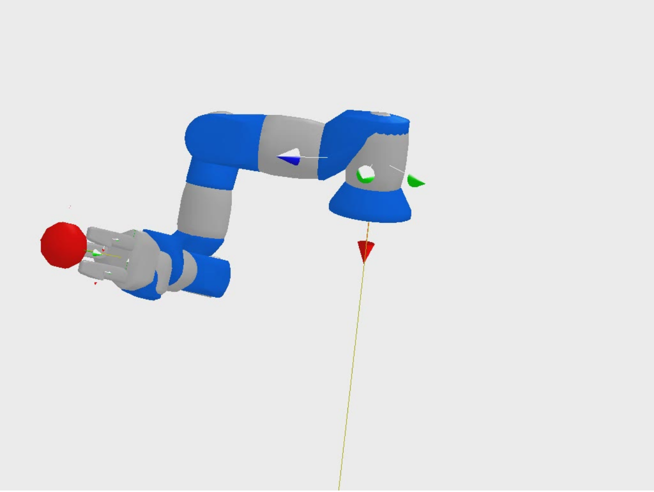
General Setup



- unknown calibrations parameters: initially, hardware is uncalibrated!
- unknown rotation & translation between robot and camera
- fiducal marker ("lolly pop") in robot's hand
- visually track lolly pop and record image coordinates $\vec{x}_i^l$ and $\vec{x}_i^r$
- record state vector $\vec{q}_i$ of robot
- ...

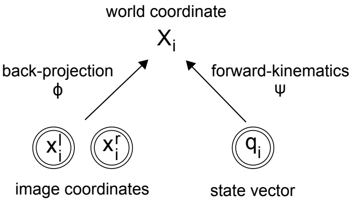
Hand-Eye Calibration

Data Acquisition



Hand-Eye Calibration

Minimization Problem



- given: set of image points $\vec{x}_i^l$, $\vec{x}_i^r$, and state vectors $\vec{q}_i$
- wanted: world coordinates $\vec{X}_i$ from non-linear mappings, such that

$$\Phi(\vec{x}_i^l, \vec{x}_i^r) = \vec{X}_i = \psi(\vec{q}_i) \tag{6}$$

Hand-Eye Calibration

Minimization Problem

- global cost function:

$$L_{\text{global}} = \sum_i |\psi(\vec{q}_i) - \Phi(\vec{x}_i^l,\ \vec{x}_i^r)|^2 \tag{7}$$

- hard to minimize eq. (7) directly
- local cost function for image points:

$$L_{\vec{x}} = \sum_i |X_i - \Phi(\vec{x}_i^l,\ \vec{x}_i^r)|^2 \tag{8}$$

- local cost function for state vector:

$$L_{\vec{q}} = \sum_i |X_i - \psi(\vec{q}_i)|^2 \tag{9}$$

Hand-Eye Calibration

Suggested Algorithm

Input: image point pairs $(\vec{x}_i^l, \vec{x}_i^r)$, state vectors $\vec{q}_i$
Output: mappings $\psi(\vec{q}_i)$ and $\Phi(\vec{x}_i^l, \vec{x}_i^r)$

1: define $\psi'(\cdot)$ as regular forward kinematics
2: **repeat**
3: learn $\Phi'(\cdot)$ given $\psi'(\vec{q}_i)$ (eq. (8))
4: learn $\psi'(\cdot)$ given $\Phi'(\vec{x}_i^l, \vec{x}_i^r)$ (eq. (9))
5: **until** $L_{\text{global}} < \varepsilon$
6: **return** $\Phi = \Phi',\ \psi = \psi'$

Hand-Eye Calibration

Learning Step

- SVM regression
- Gaussian Process
- Multi-layer Perceptron
- [your algorithm]
- ...

Next steps

- Proof of concept in simulator
- Verification with actual camera and arm
- make data available to IDA members
- ...

Conclusions

- iterative approach to the Hand-Eye calibration problem
- solving problem in “EM-fashion”
- no assumption on parameters
- no need to measure exact camera-to-robot offset
- interesting for people with algorithm but no data/problem

10q!

Questions?