POMDP Model for Continuous Calibration on Interactive Surfaces

Bastian Migge, Tim Schmidt, Andreas Kunz
Content

- Use case: Interactive surfaces
- Problem: Parallax error
- Continuous calibration using POMDPs
- Conclusion and outlook
Interactive Surfaces

- Pros
  - Intuitive operation, easy to learn
  - Real interaction (Input and output close)

- Requisition
  - Good alignment between image plane and tracking system is essential
Static Calibration

- Initially to correct geometric distortion
- A-priori setup
- Is biased by user characteristics (height, arm length)
- Depends on a single viewpoint

- Can not deal with user’s motion
- Can not handle multiple users
- Can not overcome the parallax error stemming from changing viewpoints (VP)
Parallax Error

- \( V_z \): Distance between image plane and interaction plane
- \( V_x \): Resulting parallax distortion in \( x \)-dimension
- \( a_x \): Distance from user to interaction point in \( x \)-dimension
- \( a_z \): Distance from user to interaction point in \( z \)-dimension

\[ V_x = \frac{V_z \cdot a_x}{a_z} \]

(Analogous for \( y \))
Continuous Calibration

- Parallax correction \((-V_x)\): shift pointing device information
- Information sources
  - **Target** can be assumed to be next to hit point
  - **Viewpoint** is not directly observable
  - VP can inferred from GUI interactions \((\Leftarrow V_x, V_z)\)
- Benefits
  - User dependent and self adapting error correction
  - Increases the Pointing accuracy
Problem Description

- Relative Model
  Position on axis represents deviation between actual and inferred view, the system currently correct for

- Assumptions and Limitations
  - Z axis distance of viewpoint fix
  - Infinite screen
  - Interaction (touch point) limited to Viewpoint

![Diagram showing interaction area, target on display, interaction zone, display pane, deviation space, and arm length.]

Bastian Migge © 03/2010 inspire AG
## POMDP\(^1\) Model

| State space | \(S = \{s_1, \ldots, s_{|S|}\}\) | Deviation of users actual and inferred viewpoint |
|-------------|---------------------------------|-----------------------------------------------|
| Action space | \(A = \{a_1, \ldots, a_{|A|}\}\) | Parallax correction/adjustments |
| Observations | \(O = \{o_1, \ldots, o_{|O|}\}\) | User’s pointing error (from interaction with widgets) |
| Transition model | \(P(s' \mid s, a)\) | Probability of viewpoint change between interactions |
| Observation model | \(P(o \mid s, a)\) | Likelihood of receiving an observation |
| Rewards | \(R(s, a) \in /R\) | Cost for applying correction |

---

\(^1\)Partially Observable Markov Decision Process
POMDP State space

- Model deviation of users actual and inferred viewpoint (\( S_0 \equiv \) both viewpoints align)
- Map atomic deviation space to **accumulated deviation space** (POMDP model simplicity)
3 Observations: target hit, miss left/right
   - Relative to the closest target within tolerance interval
   - Intervals mapped to small, discrete set of observations
   - Depends on the actual applied error correction

Null Observation
   - receive observation each time interval (time synchronous)
POMDP Observation Model

- Derive $P(o \mid s,a)$

User's pointing accuracy

Angle adjustment
Two independent factors for \( P(s'|s,a) \)

1) Realignment of the error compensation (actions)
2) Change of user’s position in front of the screen

1) \( P(s'|s,a) \) - **Action**: shift of the ramp functions prob. Mass

\[
\begin{array}{c|ccc}
   & s_{\text{left}} & 0.1 & 0.2 & 0.3 & 0.4 \\
\hline
s_{\text{left}} & 0.1 & 0.2 & 0.3 & 0.4 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
   & s_0 & s_{\text{right}} \\
\hline
s_0 & ... & ... & ... \\
s_{\text{right}} & ... & ... & ... \\
\end{array}
\]

an index \( a_{n,m} \) signifies \( P(m|n) \)

(0.1+0.2+0.3) stays at \( s_{\text{left}} \) and 0.4 moves to \( s_0 \)
2) $P(s'|s,a)$ - User’s position

- Static User movement model $P(s_{t+1} | s_t)$
- Convolute with $s \rightarrow s'$

Convolution of correction influence and user’s movement
Two Dimensional Parallax Correction

- X,Y screen dimensions can be treaded
  - Y analogously (Symmetry)
  - Independent

2D correction matrix (Action Space)
Implementation

- Decision maker with one step backup on value function
- Value function calculated previously by offline solver
- Implementation as embedded system
Conclusion and Outlook

- Focus was model structure
- Described basic parameterization

- More detailed model (accuracy vs. computational effort)
- User Studies to describe
  - The user’s behavior
  - The correlation between pointing accuracy and target size
  - Independent probability distr. of target positions

- Online Planner
  - redefine model during runtime
    (adapt to user’s specific behavior)
  - Apply search heuristics
Thank you for attention

Bastian Migge
miggeb@ethz.ch