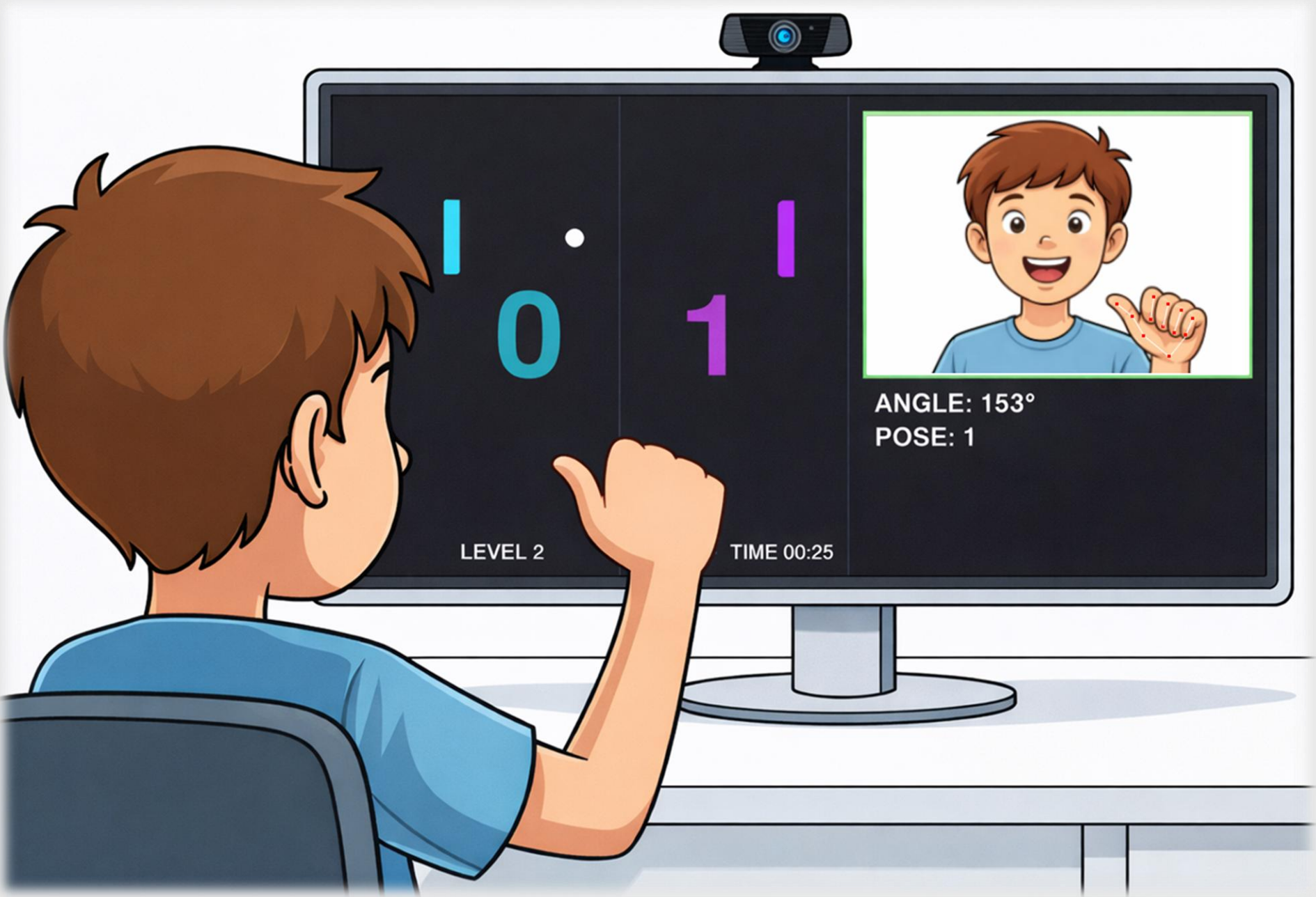


01. Einführung



Dieses Projekt entwickelt ein kontaktloses Steuerungssystem basierend auf Hand-Tracking mit Anwendung in einem interaktiven Pong-Spiel. Das System nutzt MediaPipe zur Echtzeit-Erkennung von Handbewegungen und setzt diese in Spielkommandos um. Die Implementierung demonstriert die praktische Anwendung von Computer Vision und Gesture Recognition in Gaming und Human-Computer Interaction (HCI).

02. Motivation

- **Pandemie-getrieben:** Kontaktlose Bedienung für Bankautomaten, Fahrstühle, öffentliche Terminals – eliminiert Virenübertragung durch Oberflächen
- **Barrierefreiheit:** Ermöglicht Menschen mit motorischen Einschränkungen spielerische Interaktion ohne physische Hilfsmittel
- **Natürliche HCI:** Gesten sind intuitiver als Tastatur/Maus
- **Low-Cost-Deployment:** Nur Standard-Webcam nötig
- **Zukunftsweisend:** Grundlage für immersive Interfaces u. A. in Medizin & Gaming,

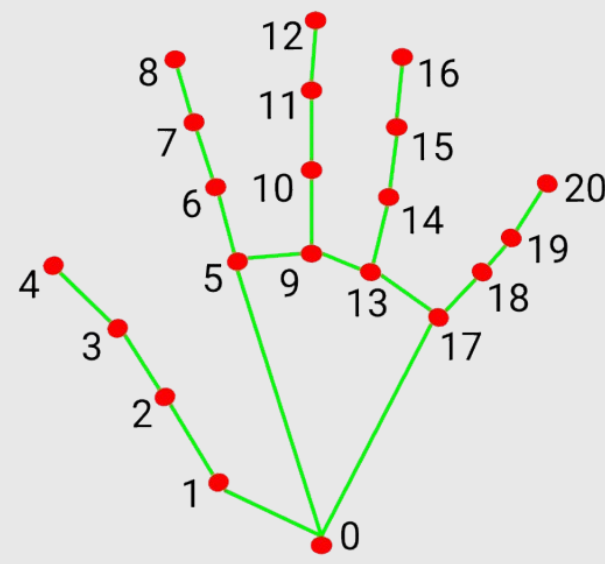
03. Spielmechanik & Interaktionsdesign

- **Menüsteuerung:** Fingerzahlen (0-5) wählen → kurzes Halten bestätigt
- **Spielmodi:** VS Bot (verschiedene Schwierigkeitsgrade) oder VS 2. Spieler (Tastatur)
- **Paddle-Steuerung:** Daumenwinkel bestimmt präzise vertikale Position
- **Dual-Tasks:** Mathe-PopUps (zeitlich begrenzt); Lösung per Finger-Geste
- **Richtig:** Spieler erhält **PowerUp** (z.B. Enlarge oder Trampoline - teils nachteilig!)
- **Falsch/Timeout:** Bot erhält PowerUp
- **PVP:** 2. Spieler löst via **Zahlen-Tasten**
- **Gewinn:** Erster auf 5 Punkte oder nach Ablauf der Zeit, der mit mehr Punkten gewinnt

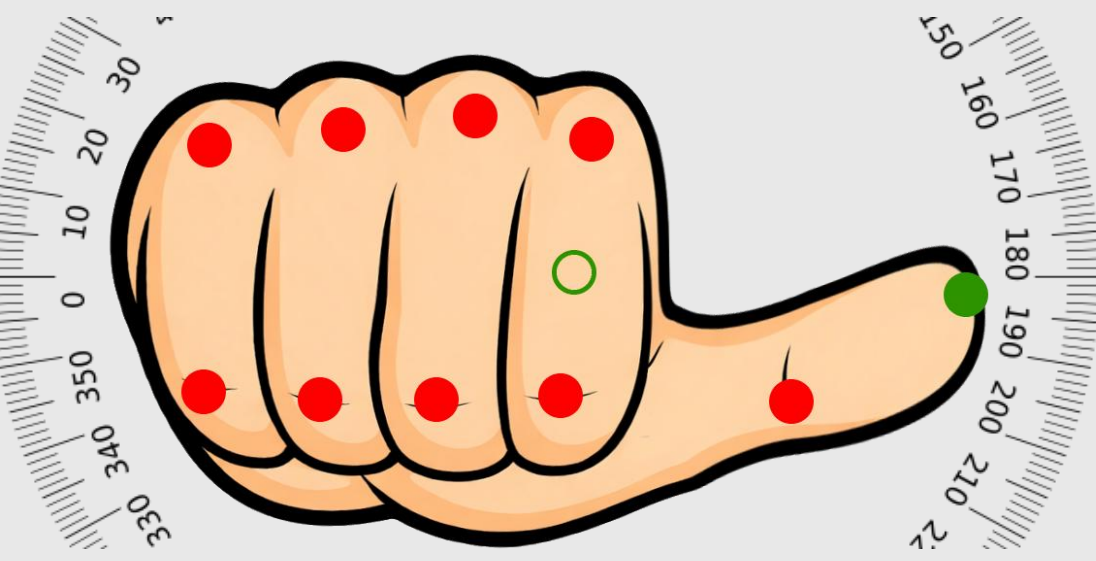
04. Methodik

MediaPipe Hands:

- **Palm Detector:** Ein spezialisierter Objekterkennungs-Modell (BlazePalm), das eine initiale Bounding-Box der Hand liefert.
- **Landmark Model:** Ein Regressionsmodell, das innerhalb der Bounding-Box eine präzise Skelett-Struktur aus 21 3D-Landmarks (Keypoints) vorhersagt.



Winkelberechnung:

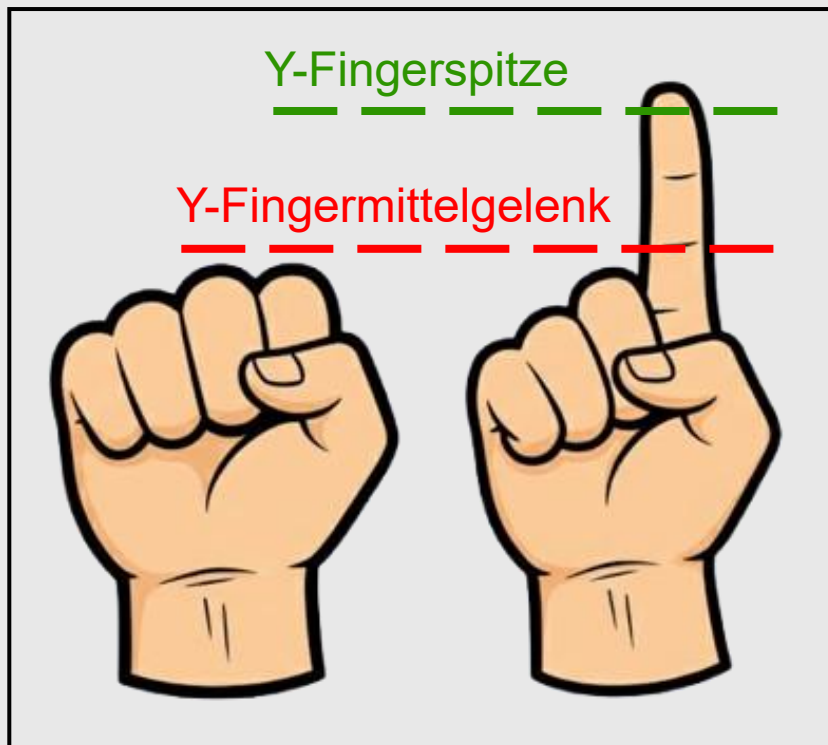


$P_{Anchor}$   $P_{Target}$

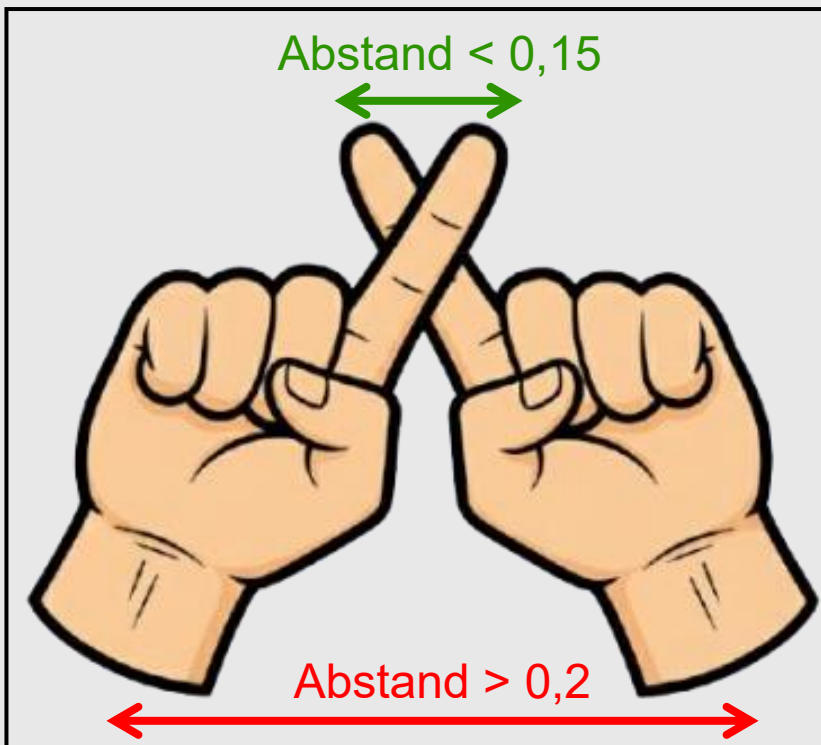
$$\Delta x = x_{Target} - x_{Anchor}$$
$$\Delta y = y_{Target} - y_{Anchor}$$
$$\theta = \text{atan2}(\Delta y, \Delta x) \pmod{360^\circ}$$

**Paddle-Mapping:**  $y_{Pos} = \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}} * (H_{GameArea} - H_{Paddle})$

Gestenerkennung und Interaktionslogik:



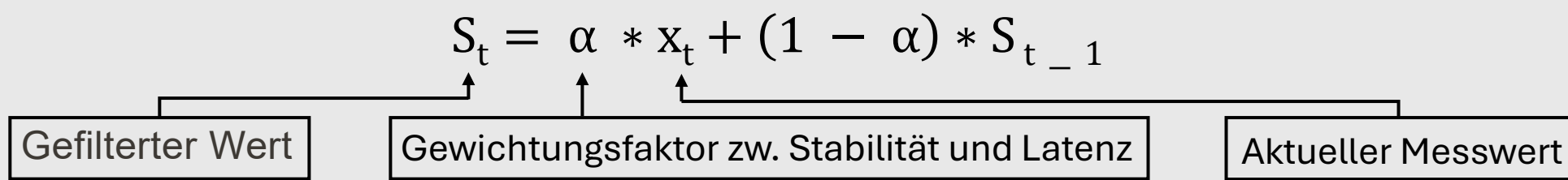
Fingerzahl-Erkennung



Pause-Geste

Adaptive Signalglättung (Jitter-Reduktion):

1. **Exponential Moving Average (EMA):** Ein Glättungsfilter zur Rauschunterdrückung durch Gewichtung von aktuellem Messwert und vorherigem Zustand.



2. **Adaptive Optimierung:** Der Faktor  $\alpha$  wird in jedem Frame dynamisch an die Bewegungsgeschwindigkeit angepasst:

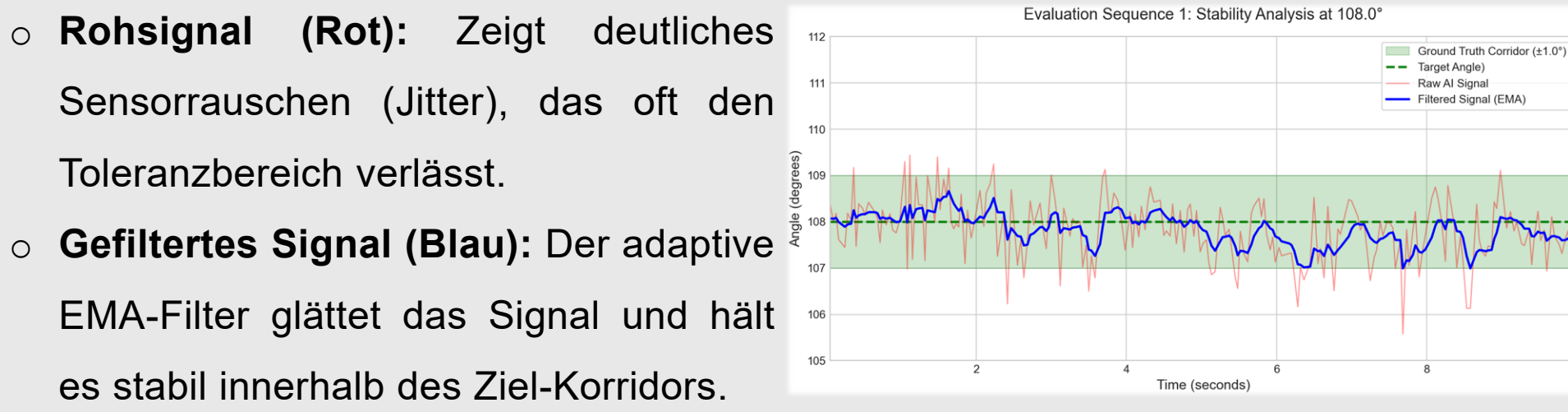
⇒ **Abweichung:**  $\Delta = |x_t - S_{t-1}|$

⇒ **Berechnung:**  $\alpha_{adaptive} = 0,15 + \min(\Delta/10, 1) * 0,75$

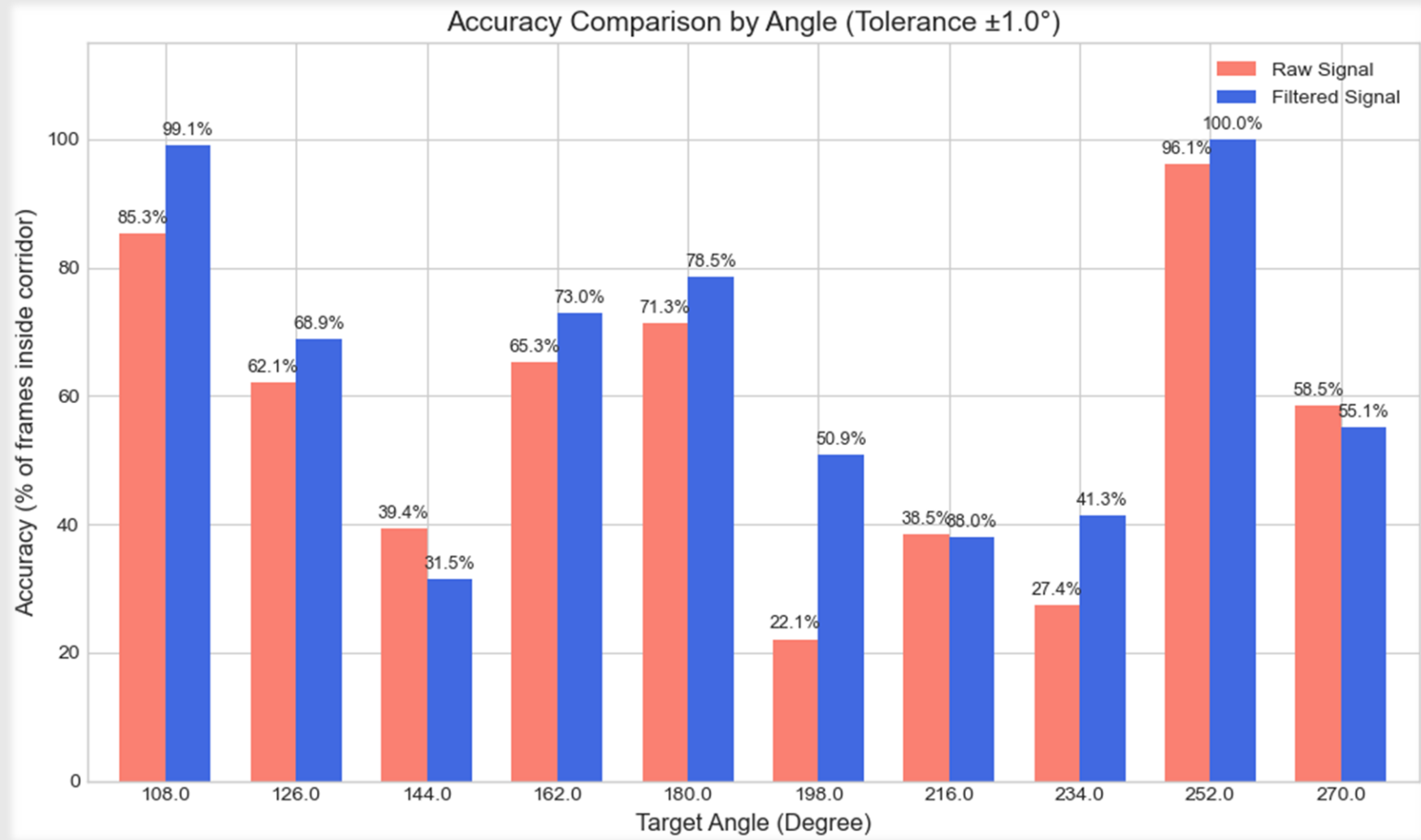
05. Ergebnisse

- Es wurden **10 zufällig gewählte Zielwinkel** aufgenommen; pro Zielwinkel wurde die Hand **10 Sekunden** möglichst stabil gehalten (insgesamt **2160 Frames**).
- Für die Auswertung wurde ein **Toleranzkorridor von ±1.0°** um den jeweiligen Zielwinkel definiert (Frame zählt als „stabil“, wenn er im Korridor liegt).

1. **Stabilitätsanalyse (Exemplarisch 108°):** Die Zeitreihenanalyse verdeutlicht die Effizienz der Rauschunterdrückung im stationären Zustand.



2. **Zuverlässigkeitsanalyse:** Das Balkendiagramm zeigt die Trefferrate (Frames im ±1.0° Korridor) über 10 Zielwinkel.



3. Quantitative Metriken:

MAE Rohsignal	MAE gefiltert	Fehlerreduktion
1,13°	0,95°	15,9% (0,18°)

06. Nächste Schritte

- **Ghost-Input beheben:** Klare Trennung Bewegung vs. Dual-Tasks; robustere Mathe-Gesten trotz dynamischer Handhaltung.
- **Multi-User:** Netzwerkfähige Engine + Multi-Kamera-Support

Referenzen

- [1] Zhang, F. et al. (2020): MediaPipe Hands: On-device Real-time Hand Tracking. Google Research.
- [2] Rautaray & Agrawal (2015). Vision-based Hand Gesture Recognition for HCI: A Survey. AI Review.

GitHub-Repo

