

Tracking juggling balls using the Kinect

Rolf Boomgaarden · Thiemo Gries · Florian Letsch

Universität Hamburg

30. Mai 2013

1 A Juggling Robot

2 Hough Transform

3 Kalman Filter

4 Our plan

5 Our plan

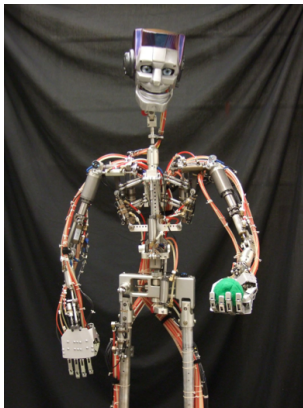
A juggling robot

Playing Catch and Juggling with a Humanoid Robot Jens Kober¹,
Matthew Glisson, and Michael Mistry²

Disney Research, Pittsburgh, USA

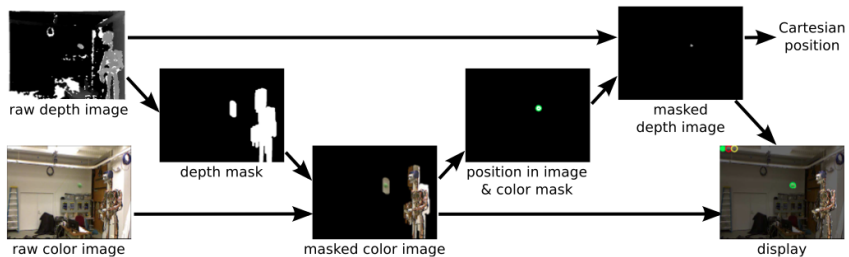
¹ Bielefeld University, Germany

² University of Birmingham, UK

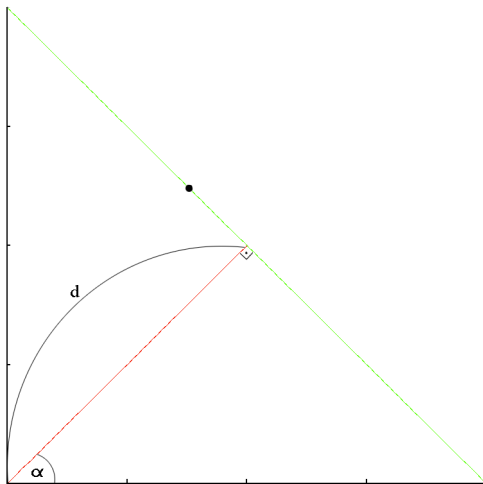


- 1 Robot hardware
- 2 Vision system
- 3 Ball Filtering and Prediction
- 4 Robot State Machine
- 5 Catching Algorithm
- 6 Throwing Approach
- 7 Juggling

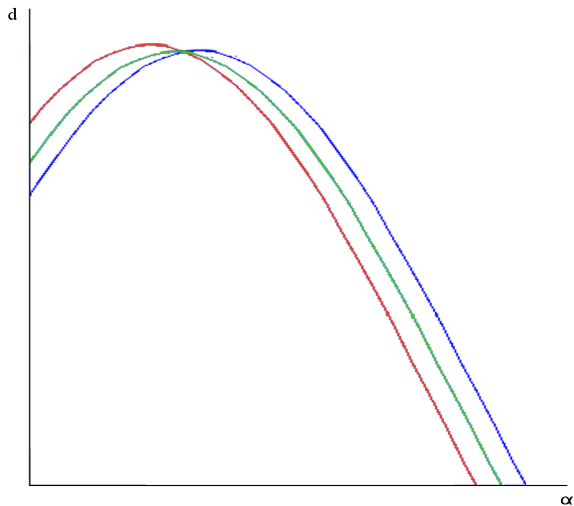
Image processing pipeline



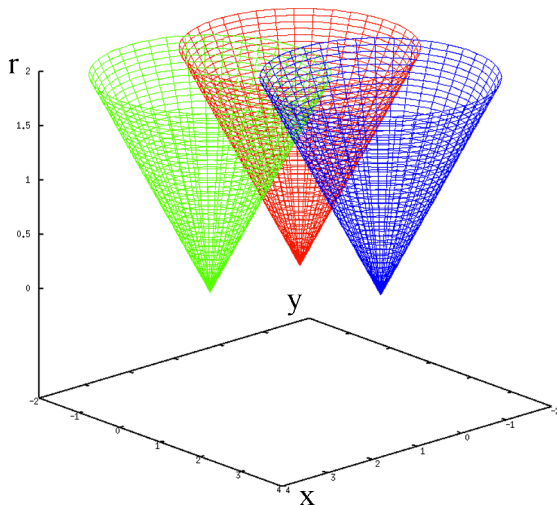
Hough Transform



Hough Transform



Hough Transform



- nutzt verschiedene Daten, um den aktuellen Zustand zu schätzen:
 - Daten aus Wissen über die Zustandsänderung
 - Sensordaten
- kann mit diesen auch zukünftige Zustände schätzen

Kalman Filter

Anwendungsfall

$t-1$

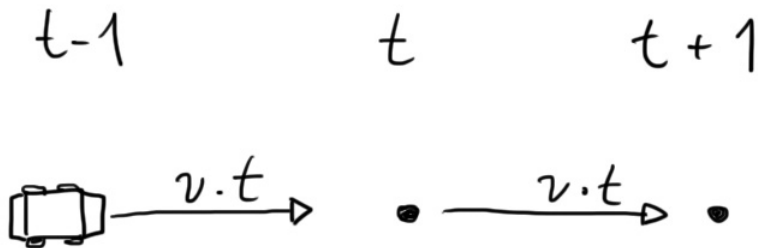
t

$t+1$



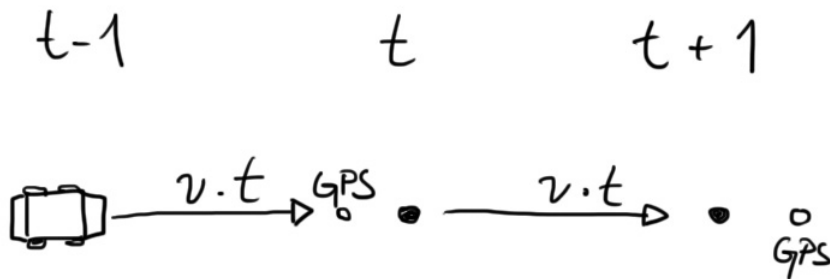
Kalman Filter

Anwendungsfall



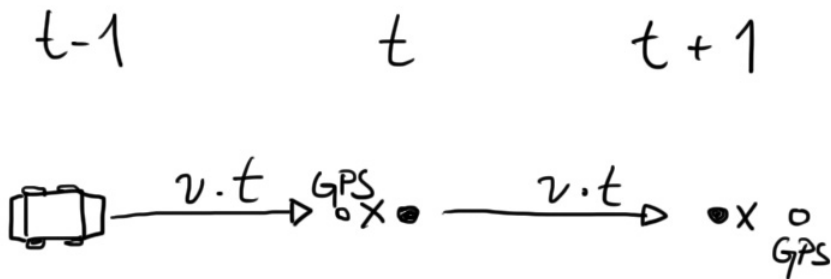
Kalman Filter

Anwendungsfall



Kalman Filter

Anwendungsfall



■ predict

- mit Wissen über die Art der Bewegung wird eine voraussichtliche Position berechnet:

$$\bar{x}_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

- zusätzlich werden die Sensordaten (hier GPS) geschätzt:

$$\bar{z}_t = H_t \bar{x}_t + \varepsilon_t$$

■ update

- geschätzte Positionen werden mit Sensordaten korrigiert:

$$x_{estimate} = \bar{x}_t + K(z_t - \bar{z}_t)$$

■ predict

- mit Wissen über die Art der Bewegung wird eine voraussichtliche Position berechnet:

$$\bar{x}_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

- zusätzlich werden die Sensordaten (hier GPS) geschätzt:

$$\bar{z}_t = H_t \bar{x}_t + \varepsilon_t$$

■ update

- geschätzte Positionen werden mit Sensordaten korrigiert:

$$x_{estimate} = \bar{x}_t + K(z_t - \bar{z}_t)$$

■ predict

- mit Wissen über die Art der Bewegung wird eine voraussichtliche Position berechnet:

$$\bar{x}_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

- zusätzlich werden die Sensordaten (hier GPS) geschätzt:

$$\bar{z}_t = H_t \bar{x}_t + \varepsilon_t$$

■ update

- geschätzte Positionen werden mit Sensordaten korrigiert:

$$x_{estimate} = \bar{x}_t + K(z_t - \bar{z}_t)$$

- Wissen über Art der Bewegung → Wurfparabel
- Sensordaten → Kinect

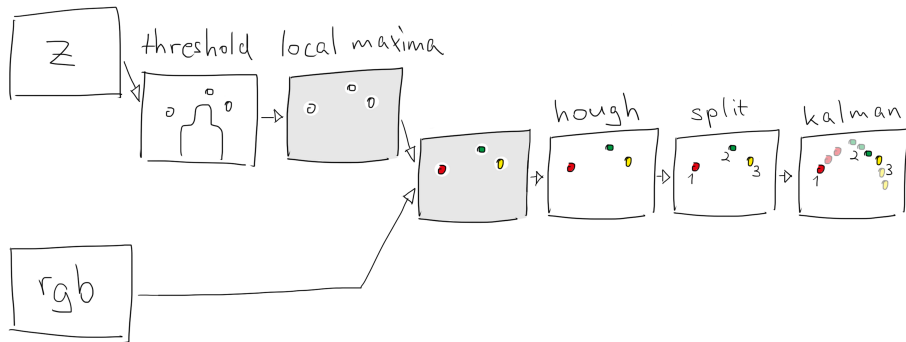
Conclusion

Video + paper conclusion

IN: $\text{rgb} + z$

- eliminate background in z
- search local maxima in $z \rightarrow \text{ROIs}$
- use ROI mask on rgb
- Hough transform on rgb
- Match corresponding balls in multiple frames
- Kalman filter for each ball

The flow



Challenges

- Local maxima
- Multiple balls (colour?)

The end

EOF