

Tracking mit Pilot

„Leonie come here to me“

Erstellt von Jens Nill und Thomas Kärcher

WS2014/15

Agenda

Aufgabenstellung

Projektübersicht

Lösungswege

Projektausblick und Erweiterungen

Aufgabenstellung

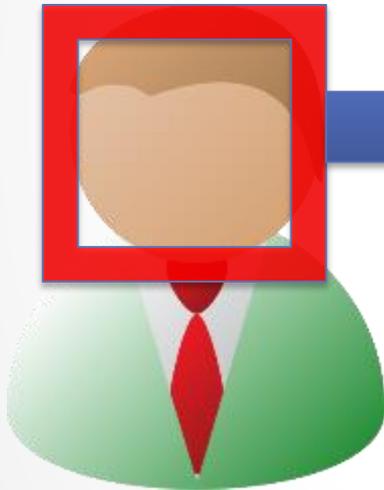
SCITOS (Leonie) soll eine Person bzw. ein Objekt im Kamerabild tracken und anschließend mittels Pilot verfolgen.

Teilaufgaben:

- Adaptive Face Tracking gemeinsam mit dem Mira Center auf einer VM lauffähig machen.
- Auslesen der benötigten Werte aus dem „Adaptive Face Tracking“.
- Umrechnung dieser Werte, um die Position der Person bzw. des Objekts zu erhalten.
- Ansteuerung des Roboters per Pilot anhand der berechneten Werte.

Aufbau des Projekts

FaceTracking



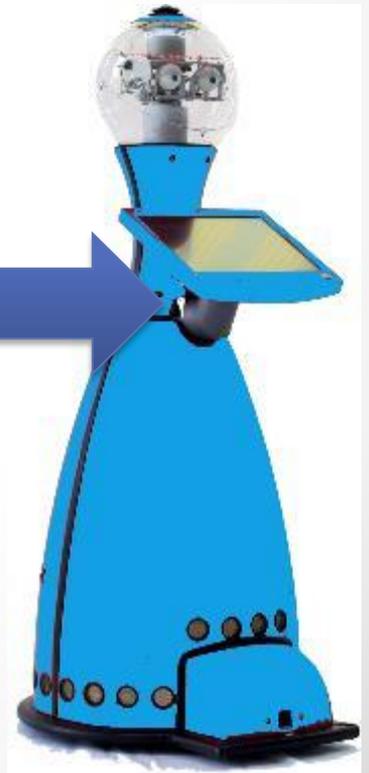
Höhe
Breite



Umrechnen
in
benötigte
Werte

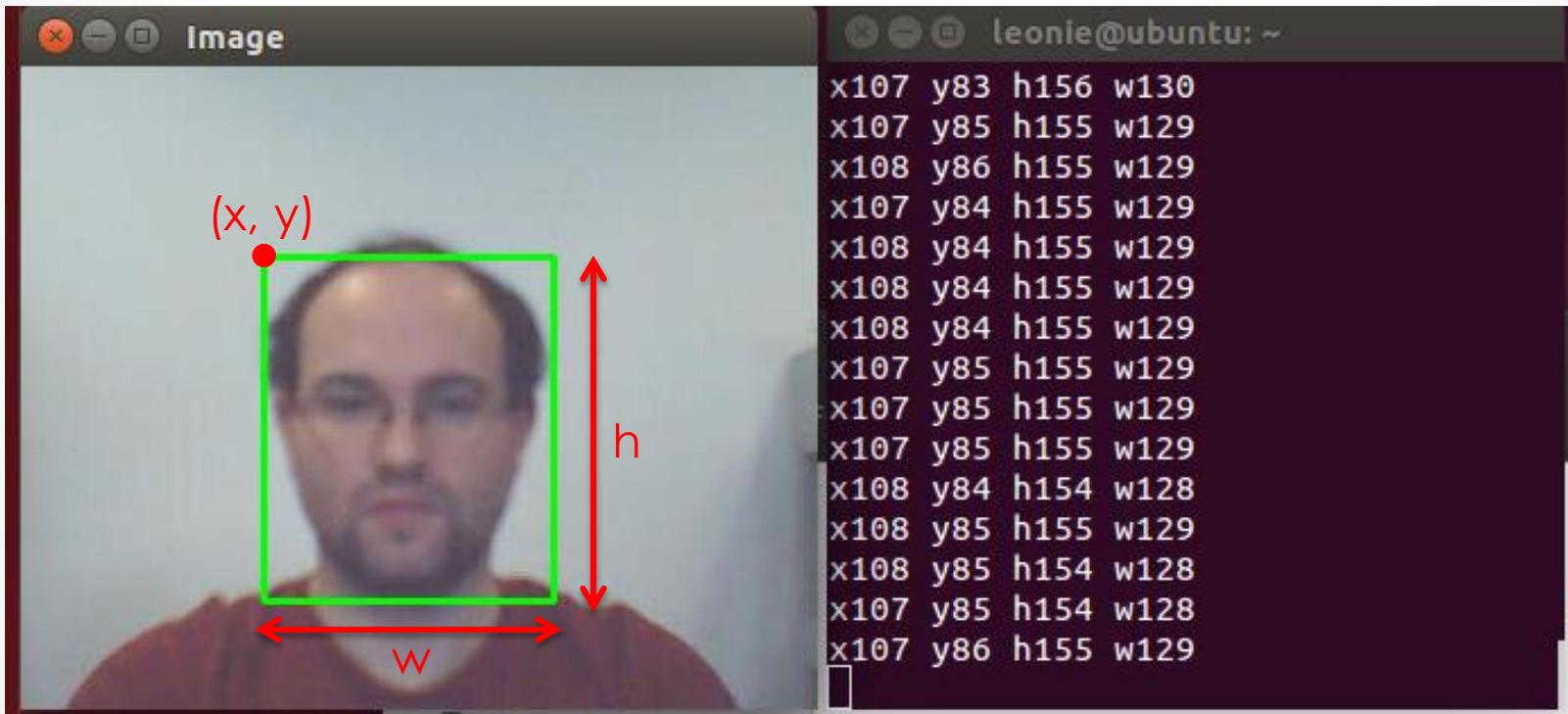


Mira PILOT
auf SCITOS



Lösungsweg

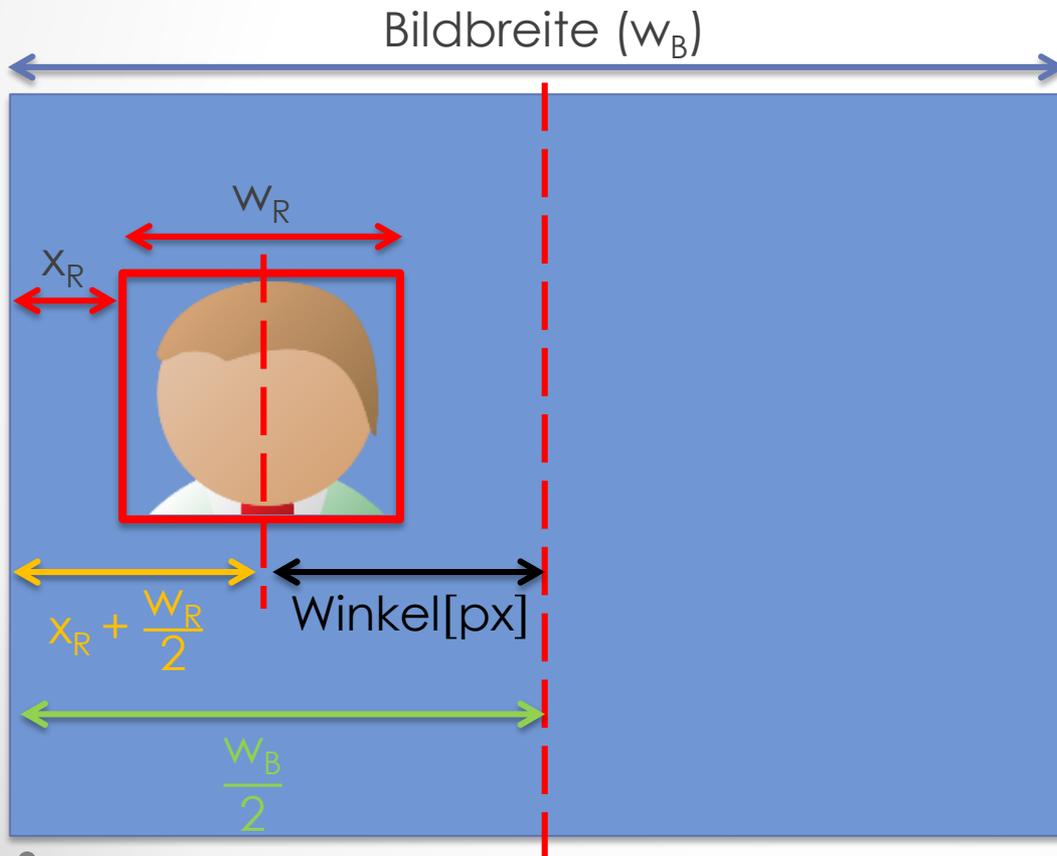
- Ausgabe der benötigten Daten aus dem Tracking“



- Erstellen des Quellcodes zur Auslesung der Daten

Umrechnung in lokale Polarkoordinaten

Winkelberechnung:



$$\text{Winkel}[\text{pixel}] = x_R + \frac{w_R}{2} - \frac{w_B}{2}$$

x_R = Abstand zum Rechteck
 w_R = Rechteckbreite

Umrechnung in lokale Polarkoordinaten

Winkelberechnung:

$$\text{Winkel[pixel]} = x_R + \frac{w_R}{2} - \frac{w_B}{2}$$

Bsp. für Kamera HD Webcam C270 von Logitech:

$$\text{Winkel[pixel]} = x_R + \frac{w_R}{2} - \frac{640px}{2}$$

Mit Kamerablickwinkel von 60° ergibt sich:

$$\text{Winkel}[^{\circ}] = \left(x_R + \frac{w_R}{2} - \frac{640px}{2} \right) \cdot \frac{60^\circ}{640px}$$

Umrechnung in lokale Polarkoordinaten

Abstands-/Tiefenabschätzung:

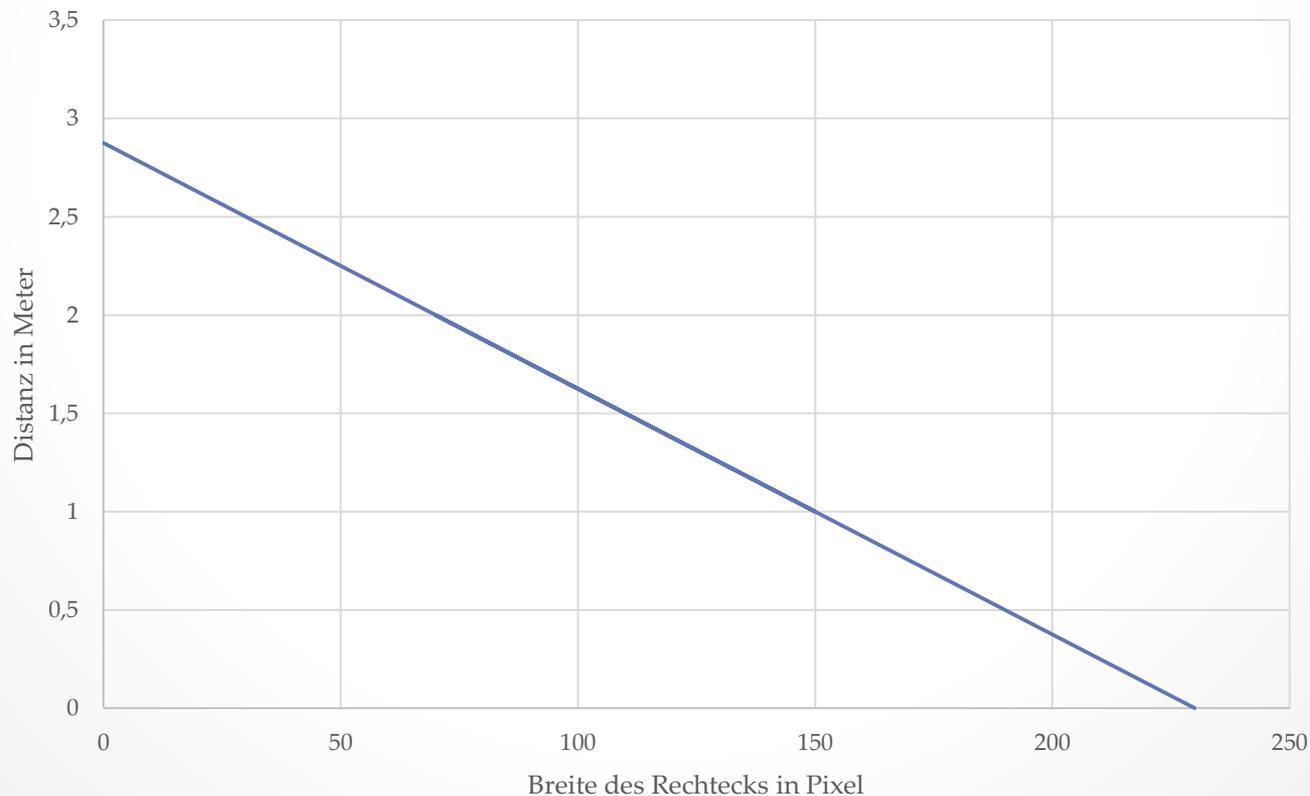
- Messung der Rechteckbreite in Abhängigkeit des Abstandes:

Abstand [m]	Rechteckbreite w_R [pixel]
1	150
1,5	110
2	70

Umrechnung in lokale Polarkoordinaten

Abstands-/Tiefenabschätzung:

Distanz in Abhängigkeit von der Breite des Rechtecks



Umrechnung in lokale Polarkoordinaten

Abstands-/Tiefenabschätzung:

Umrechnung mit ermittelter Gleichung:

$$\text{Abstand[m]} = -\frac{1}{80_{px}} \cdot w_R + 2,875\text{m}$$

→ Ermittlung des Abstands in Abhängigkeit der Rechteckbreite w_R

Ausblick und Erweiterungen

- Übergabe der errechneten Variablen an die Funktion `setGoal()` incl. Toleranzen
- Schwankungen der Abstandsberechnung mittels Toleranzen (Stufen) minimieren
- Sobald möglich: Auf andere Kameras umrechnen