

# Kopfsteuerung eines Androiden

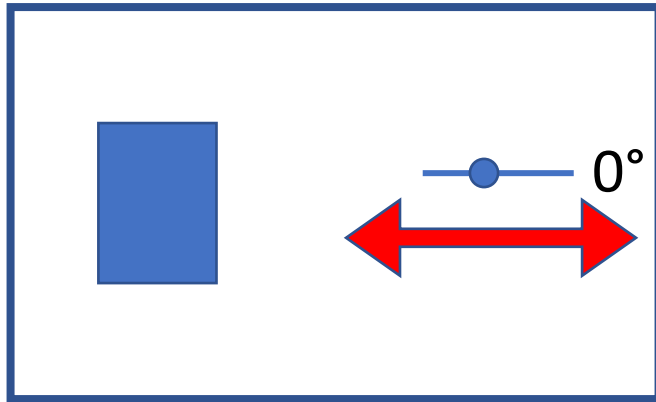
Projektpräsentation im Rahmen des CT-Unterrichtes, Jeremy Sobota

# Gliederung

- Projektbeschreibung, Demonstration
- Ansteuern der Servos
- Bewegungserkennung
- Videostream-Übertragung

Demonstration

# Ansteuern der Servos (Client)



JavaScript-  
Ereignis  
**onchange**



```
function send() {  
  data      =  KopfX:15:-45:45  
  jsForm    =  new FormData();  
  jsForm.append( „Schlüssel123“, data );  
  var xhttp =  new XMLHttpRequest();  
  xhttp.open("POST", "/formloader", true);  
  xhttp.send(jsForm);  
}
```

```
<input type="range" oninput="change()" onchange="send()" id= "Schieberegler" min="-45"  
max="45" step="1" value =„15" >
```

# Ansteuern der Servos (Server)

- Hilfsklasse servo() aus Datei head.py
  - \_\_init\_\_()
  - writeDegFromServer()
- servoHX= head.servo(...)
- dictionary={ „HeadX“ : servoHX, ...}



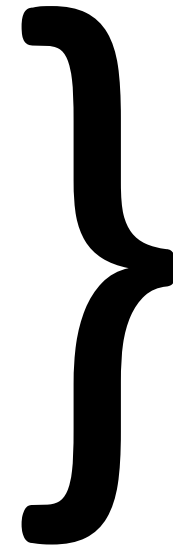
**Initialisierung**

/formloader  
HTTP POST

Clientdaten

**HeadX:15:-45:45**

- formData = request.form['data']
- servo, deg, degmin, degmax = formData.split(":")
- servoToUpdate=dictionary[servo]
- servoToUpdate.writeDegFromServer(  
degmin,  
degmax,  
deg )



**HTTP-POST**

# Bewegunserkennung

- OpenCV

aktuelles Bild

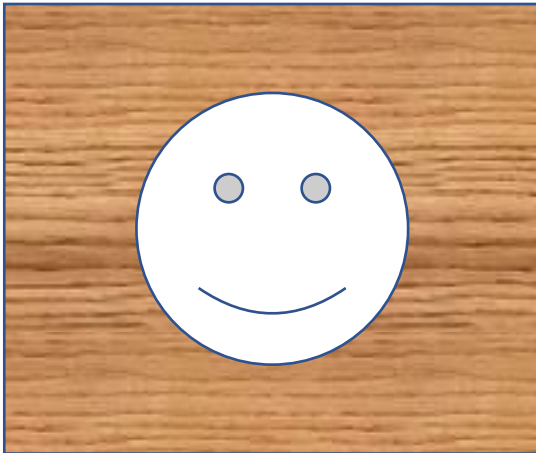


Referenzbild (gewichtet 25%)



# Bewegunserkennung

aktuelles Bild

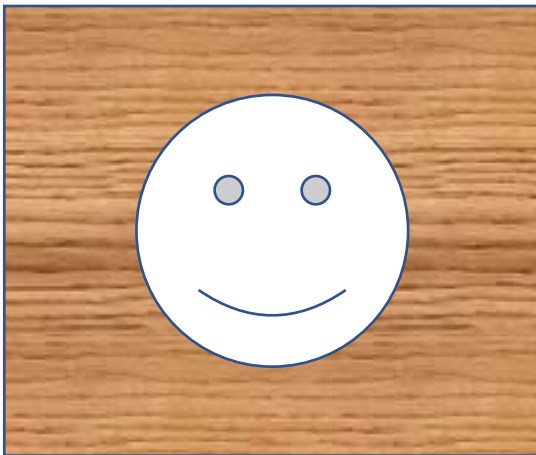


Referenzbild (noch nicht aktualisiert)



# Bewegunserkennung

aktuelles Bild



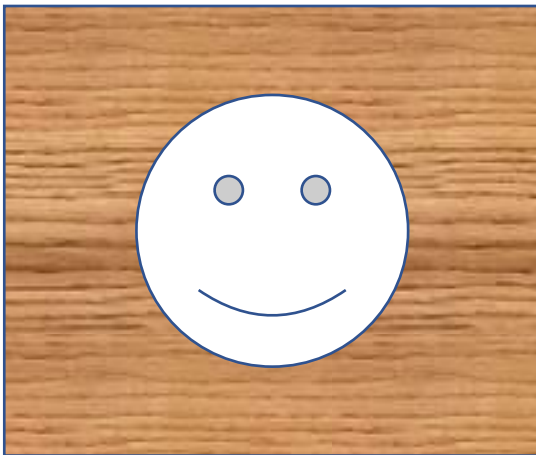
Referenzbild (Gewichtung erster Durchlauf)





# Bewegunserkennung

aktuelles Bild

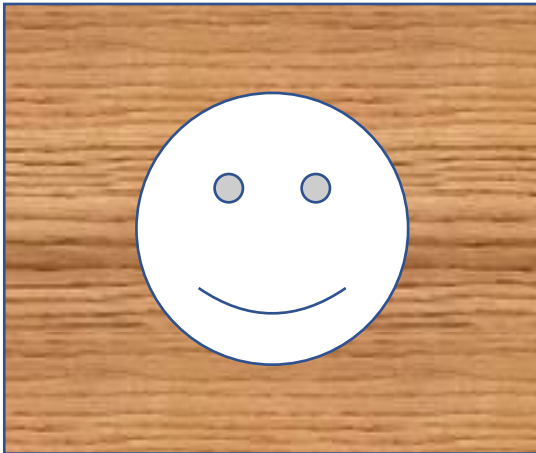


Referenzbild (Gewichtung zweiter Durchlauf)



# Bewegunserkennung

aktuelles Bild

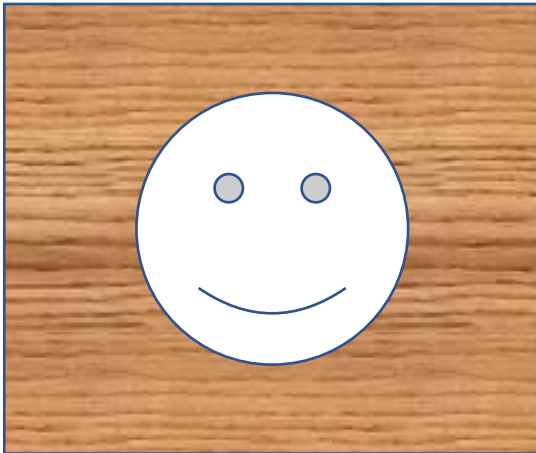


Referenzbild (Gewichtung dritter Durchlauf)



# Bewegunserkennung

aktuelles Bild

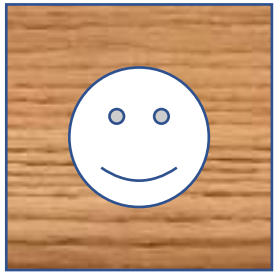


Referenzbild (Gewichtung vierter Durchlauf)



# Bewegunserkennung

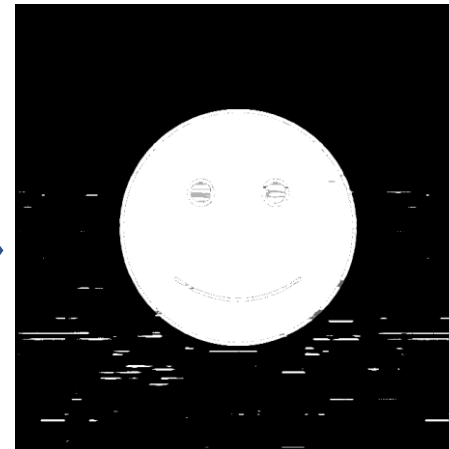
aktuelles Bild



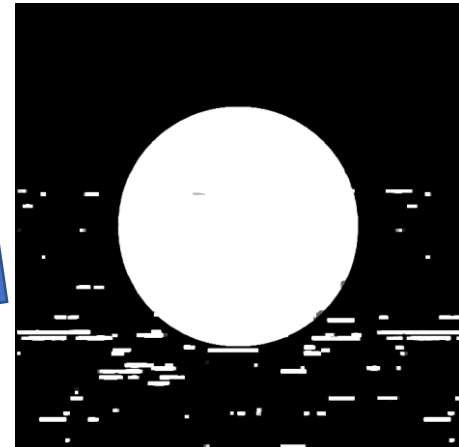
Referenzbild



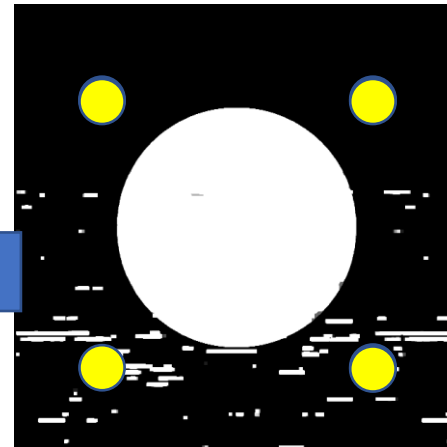
Threshold



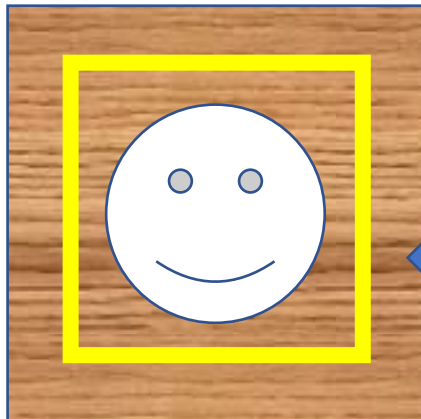
dilate



findContours



Resultat



# Videostream-Übertragung

createDefaultPicture()

stream()

MIME-Type

→ multipart/x-mixed-replace;

boundary=grenze

---

```
def stream():
```

```
    while True:
```

```
        frame = getFrame()
```

```
        yield (b'--grenze\r\n' b'Content-Type:
```

```
            image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```

HTTP/1.1 200 OK

Content-Type: multipart/x-  
mixed-replace;  
boundary=grenze

--grenze

Content-Type: image/jpeg  
JPEG-BYTES

--grenze

Content-Type: image/jpeg  
JPEG-BYTES