

Kompaktes Kollisionswarnsystem für Menschen mit Sehproblemen

Praxisphase - Medieninformatik

Eine kurze Geschichte

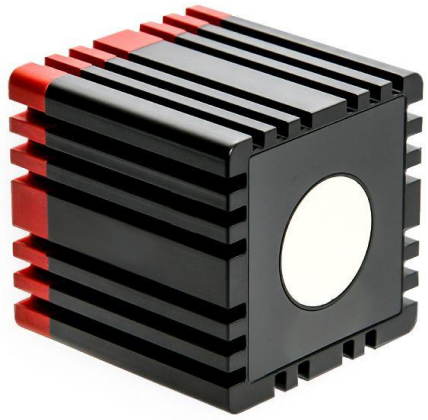
Agenda

- Ziel des Projekts
- Planung
- Implementierung
- Résumé

Ziel des Projekts

Ziel des Projekts

- **Unauffälliges** tragbares System für den In- und Outdoor Einsatz
- Art der Tragevorrichtung: z.B. Gürtel oder Weste
- Das System besteht aus:
 - 1x Sensor-System zum Erkennen von Hindernissen
 - 2x Feedback-System (haptisch oder akustisch)
 - Batterien/Akkus
- Nähert sich die Person einem Hindernis wird das Feedback-System ausgelöst: z.B. *unterschiedlich starke Vibration auf Basis der Entfernung des Hindernisses*



Planung

Wahl des Sensors

Wahl des Sensors (Kriterien)

- Größe
- Stromverbrauch
- Wärmeentwicklung
- In/Outdoor-fähig
- Störanfälligkeit
- Einfluss auf die Umwelt
- Kosten

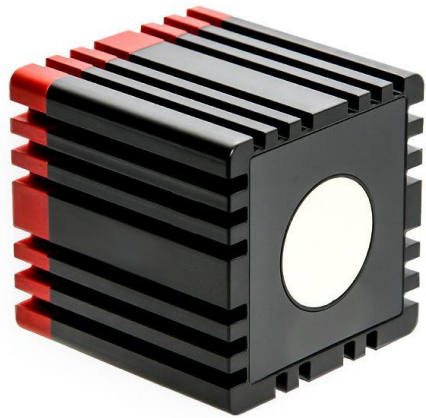
Mögliche Sensortypen

- Ultraschall
- LIDAR / RADAR
- 2D/3D Kamera
- Lichtfeldkamera
- RGB-D Kamera
- Time-of-Flight Kamera

Google Car – LIDAR-System auf dem Dach



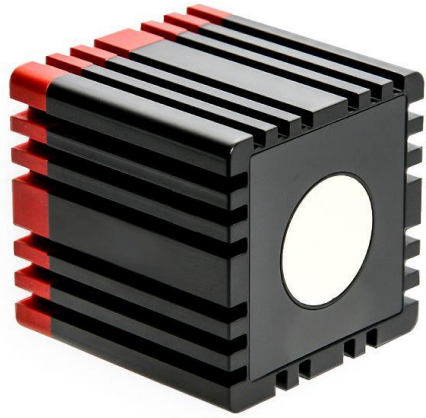
Kinect for Windows v1



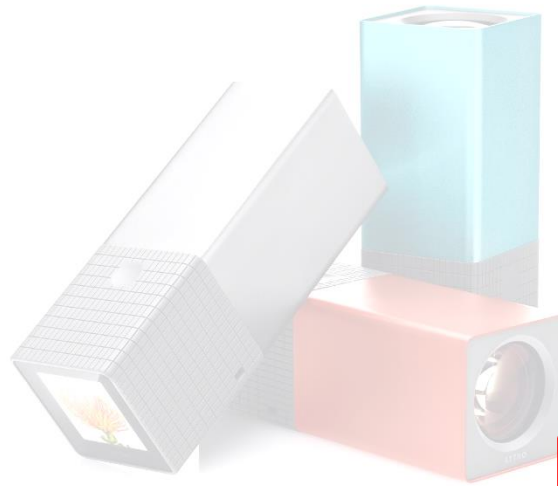
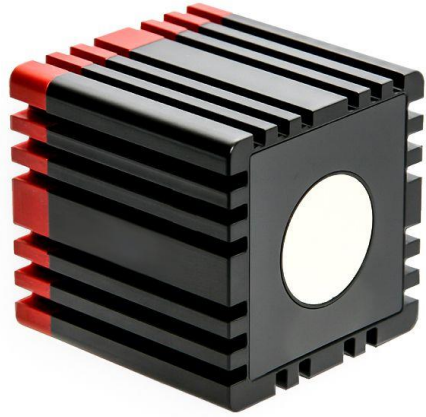
Kinect for Windows v1



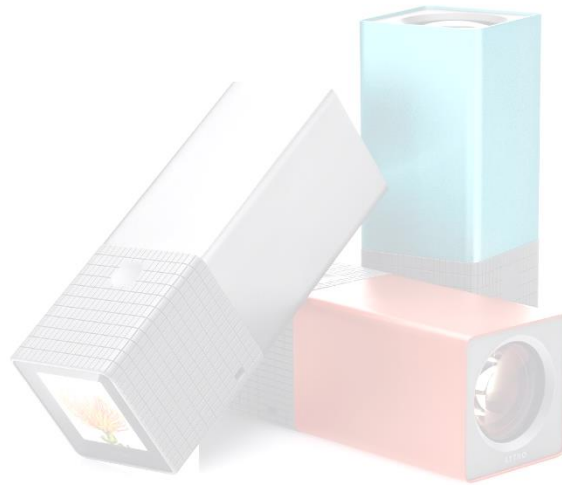
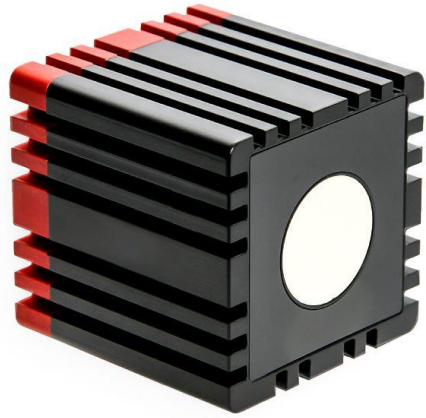
Lytro - Lichtfeldkamera



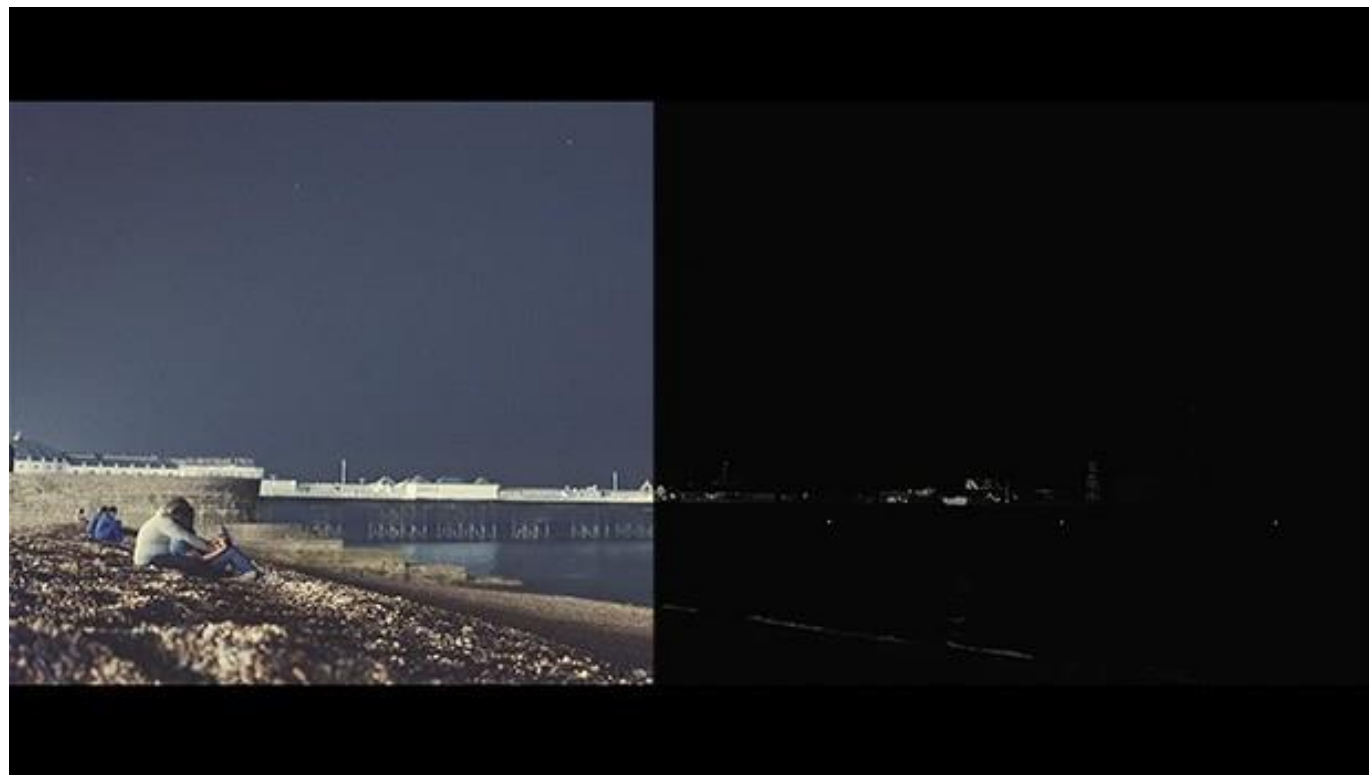
Ultraschallsystem



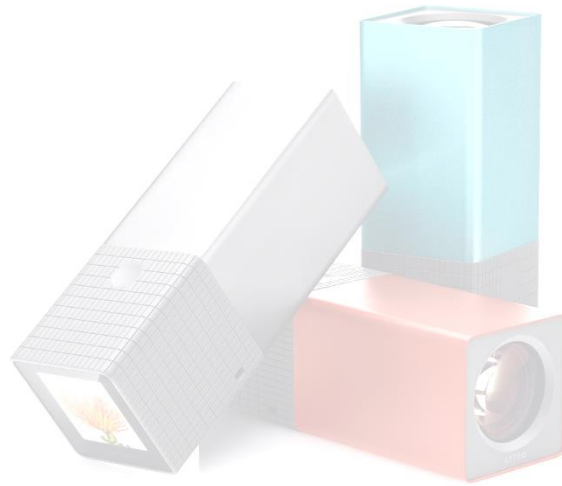
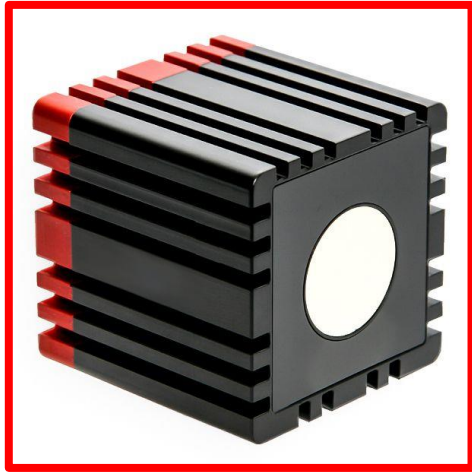
3D CMOS Sensor



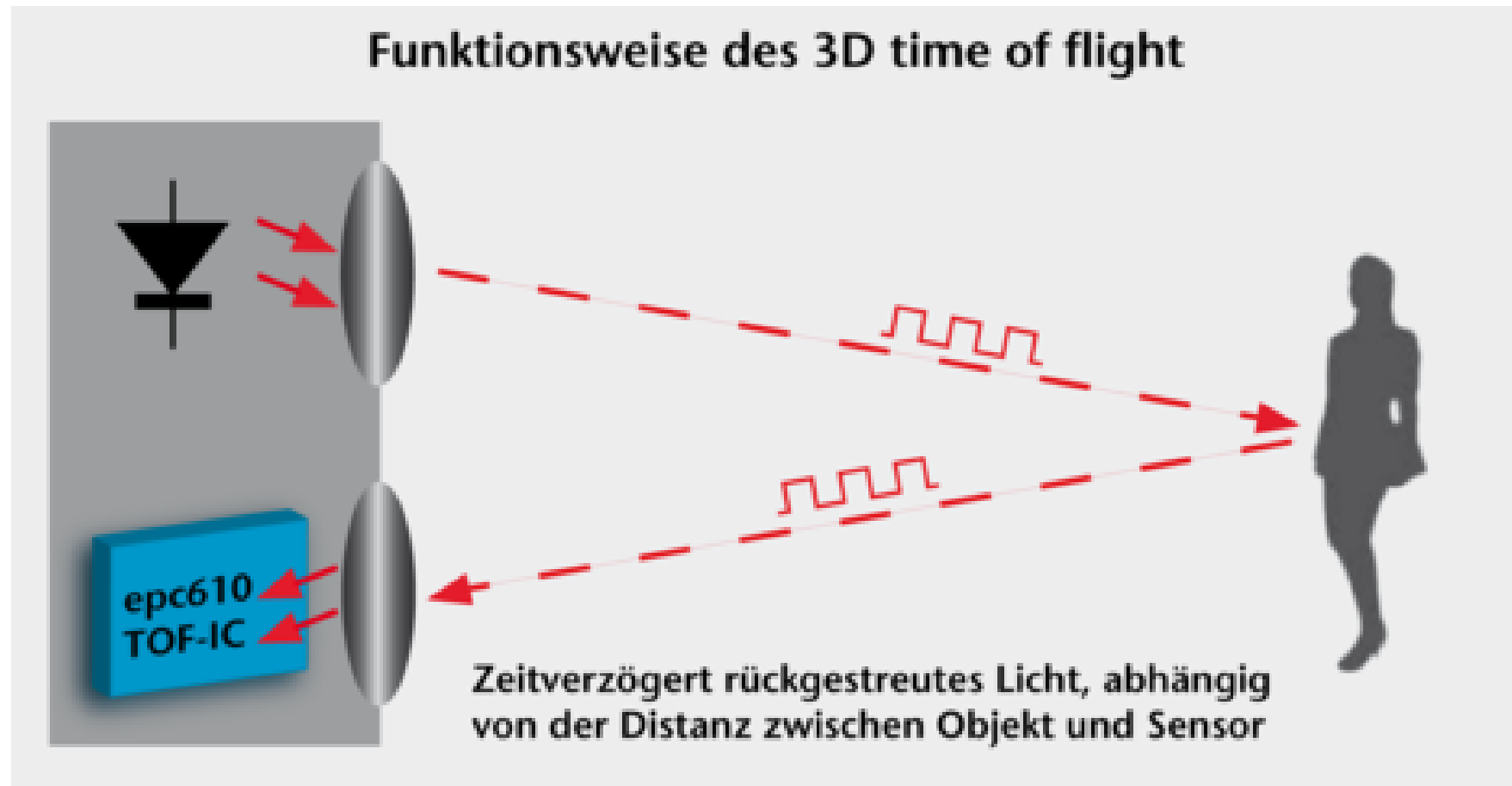
Sony a7S – Low-light



Time-of-Flight System



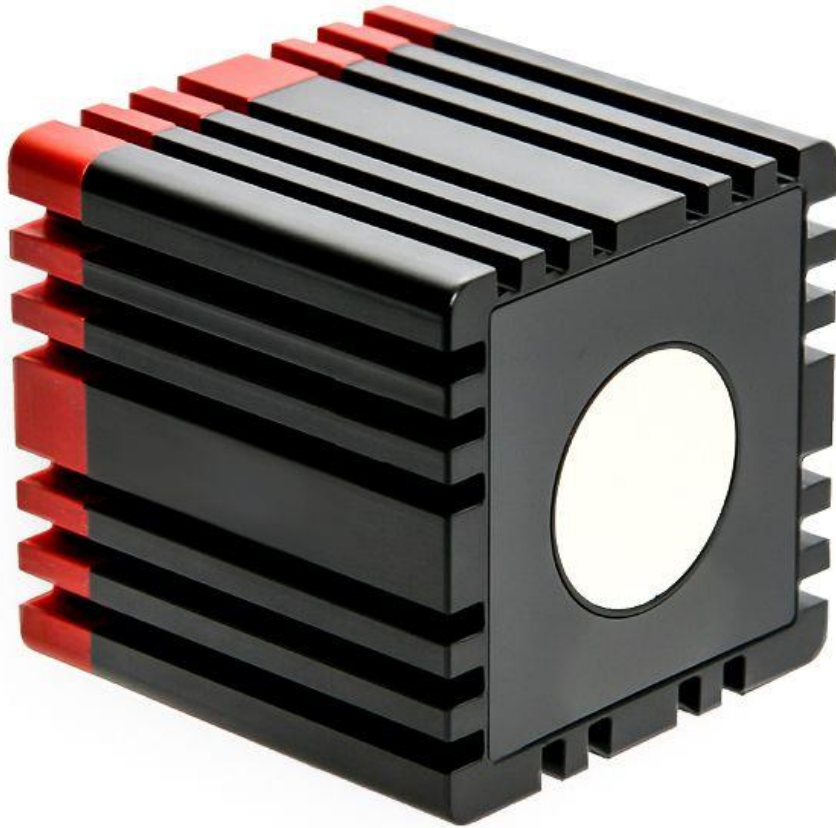
Time-of-Flight - SwissRanger SR4000



Time-of-Flight System

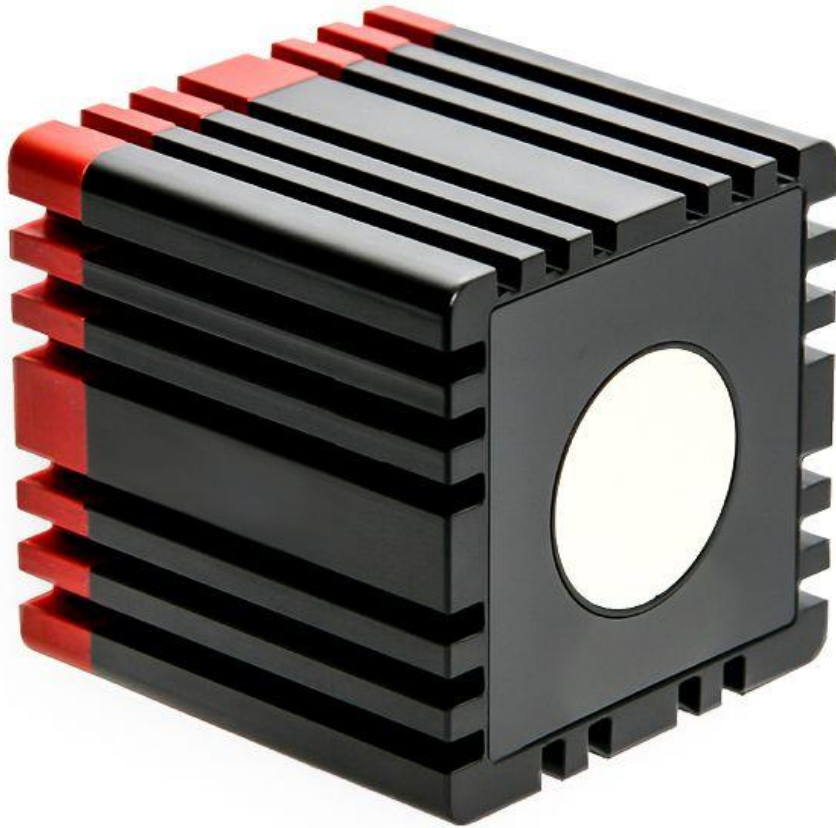
- Eigene aktive Lichtquelle
- Jeder Bildpunkt hat einen Tiefenwert
- Aufwändige Technik
- Je länger der Lichtimpuls, desto größer ist die möglich Messreichweite
- 50 ns ca. 7,5m Reichweite
- **Vorteil:** Geschwindigkeit und Genauigkeit
- **Nachteil:** Externe Lichtquellen

Time-of-Flight - SwissRanger SR4000



- Auflösung: 176 x 144 (HxB)
- Maße: 65 x 65 x 76 mm (W x H x D)

Time-of-Flight - SwissRanger SR4000



- Auflösung: 176 x 144 (HxB)
- Maße: 65 x 65 x 76 mm (W x H x D)

4,295 USD

Kinect for Windows v2



- HD-Sensor (1920 x 1080 Pixel) bis 30fps
- IR-Sensor (512 x 424 Pixel) @ 30fps
- Depth-Sensor (512 x 424 Pixel) @ 30fps
- Tiefen-Reichweite: 0,8 m bis 7,0/8,0 m
- **Systemanforderungen:**
 - Windows 8 (64 bit)
 - 2 GB RAM,
 - Physikalische 3,1 GHz Dual-Core CPU
 - USB 3.0
 - DirectX 11 (keine VM-Unterstützung)

Planung

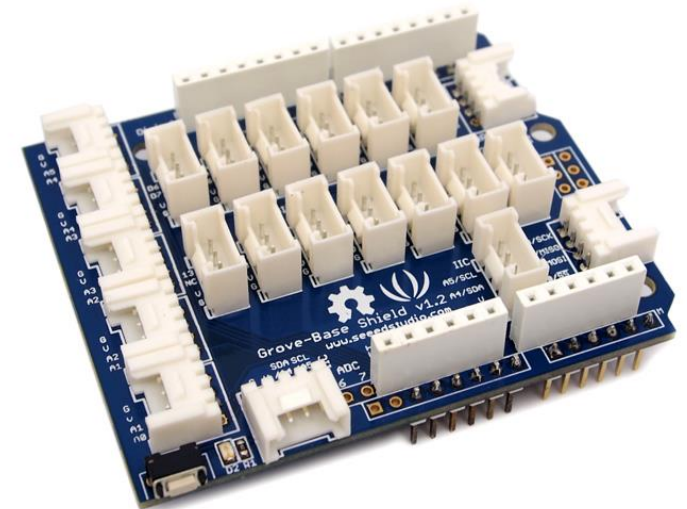
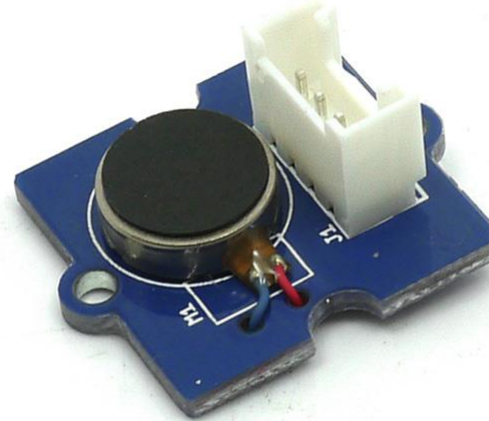
Wahl der Feedback - Methode

Wahl der Feedback - Methode

Arduino UNO Rev3



SeeedStudio Grove:



Implementierung

Feedback-System

Feedback-System

- Regulation der Vibrationsstärke via:
 - Spannung
 - Pulsweitenmodulation (PWM)
- PC kommuniziert via USB (SerialPort) mit Ardunio
- Zukünftig: Kommunikation mittels Bluetooth

Implementierung

Einfach Erkennung vom nächst möglichen Hindernis

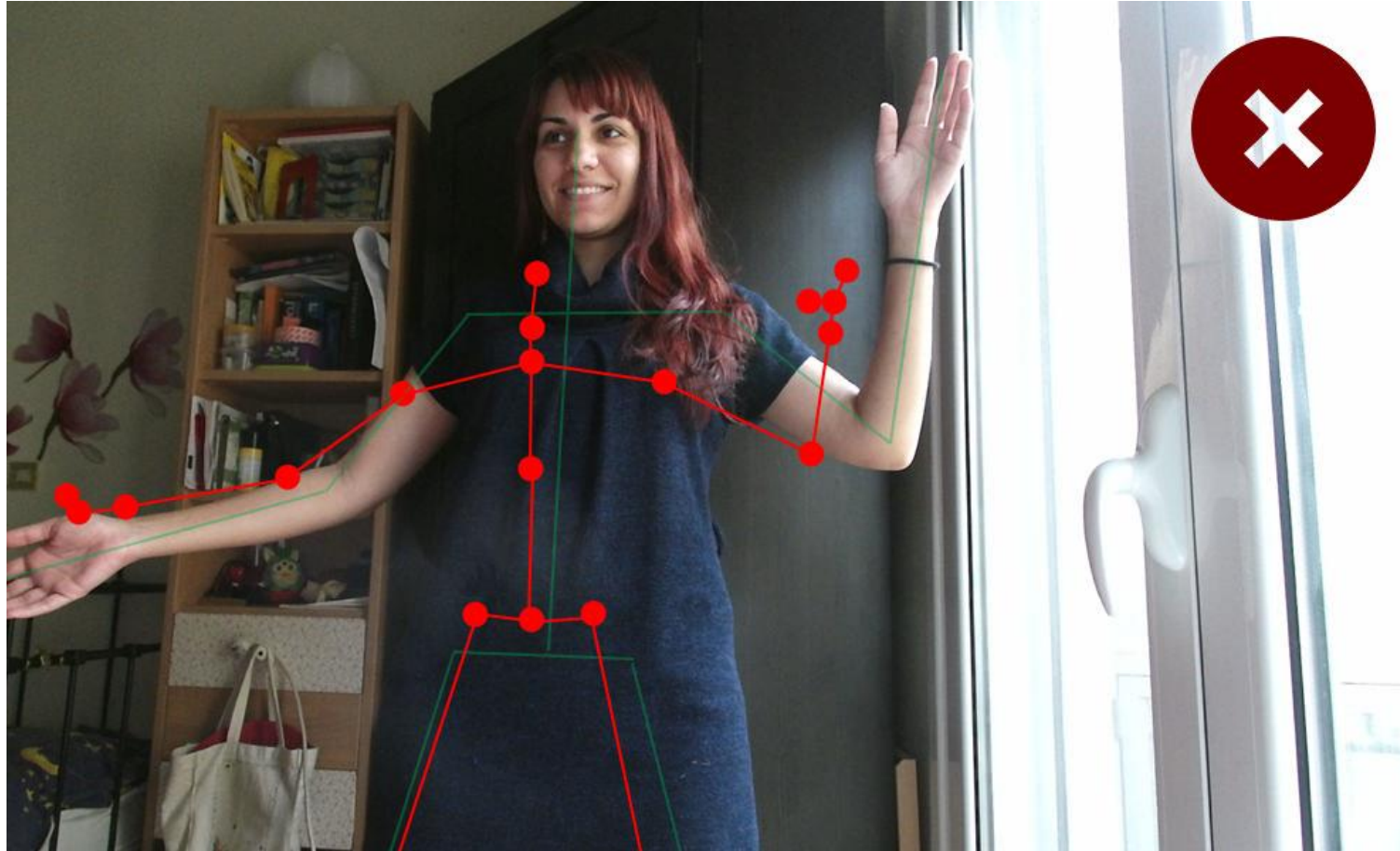
Einfache Erkennung vom nächst möglichen Hindernis

- Prüfen des Tiefenwerts jedes Pixels
- Tiefenwert soll nicht kleiner als 1m und nicht größer als 1,5 m sein
- $512 \times 424 = 217088$ Iterationen
- Hindernis entdeckt = Alle 3 Motoren vibrieren mit voller Leistung
- **Problem:**
 - Fehlerhafte oder fehlende Tiefenwerte (max. fehlende Werte 15%)
- **Lösung:**
 - Eine gewisse Anzahl von Bildern abwarten und vergleichen

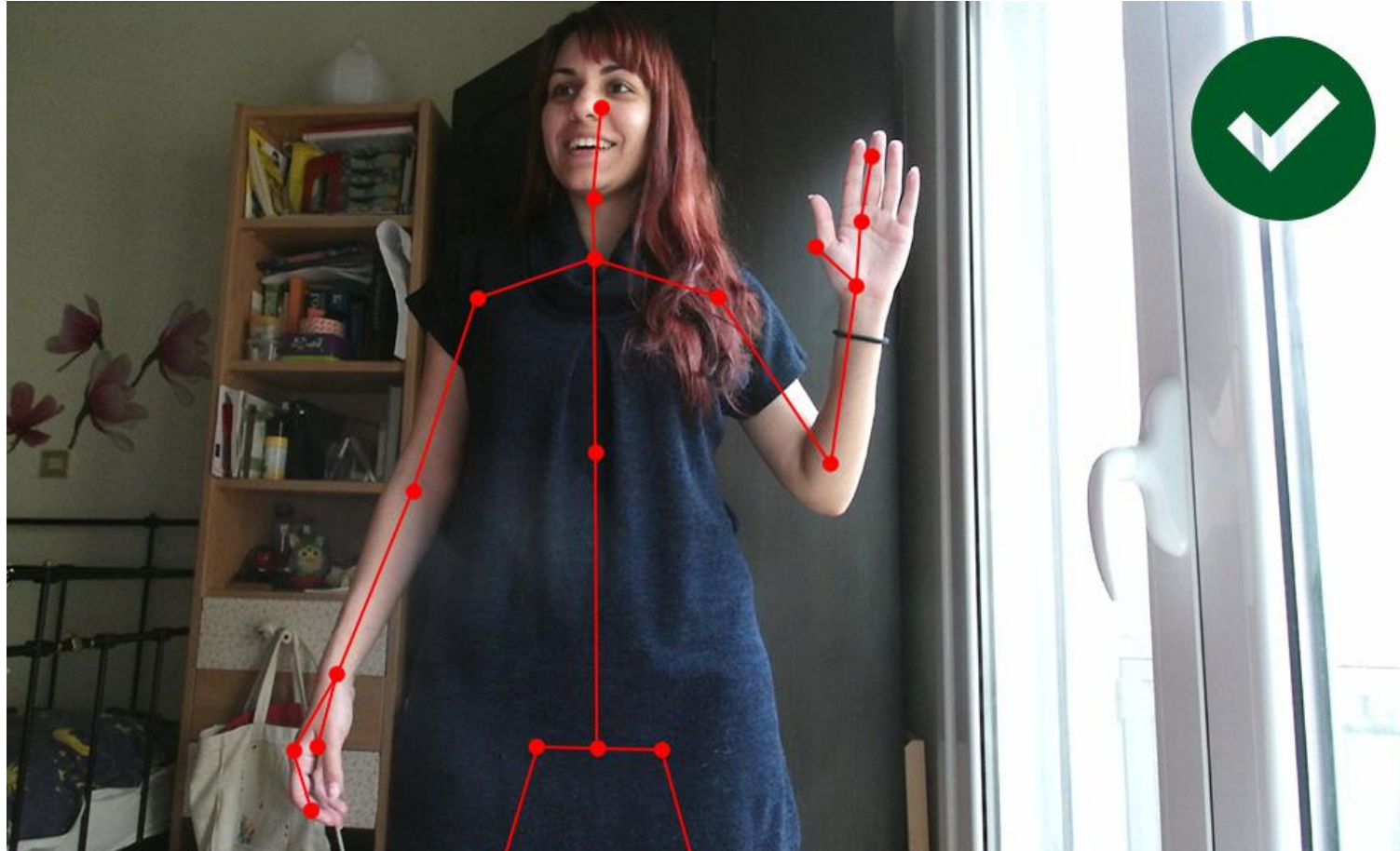
Unterschiedliche Koordinatensystemeinteilungen

- CameraSprace Koordinaten (X, Y, Z) -> Unbekannte Einteilung
- DepthSpace Koordinaten (X, Y) -> 512 x 424
- ColorSpace Koordinaten (X, Y) -> 1920 x 1080
- **Problem:**
 - CameraSpace X und CameraSpace Y entsprechen keiner Sensorauflösung
 - Koordinaten müssen in die jeweilige Koordinatensystemeinteilung umgerechnet werden

Kinect Koordinaten: CameraSpace -> ColorSpace



Kinect Koordinaten: CameraSpace -> ColorSpace



Demo

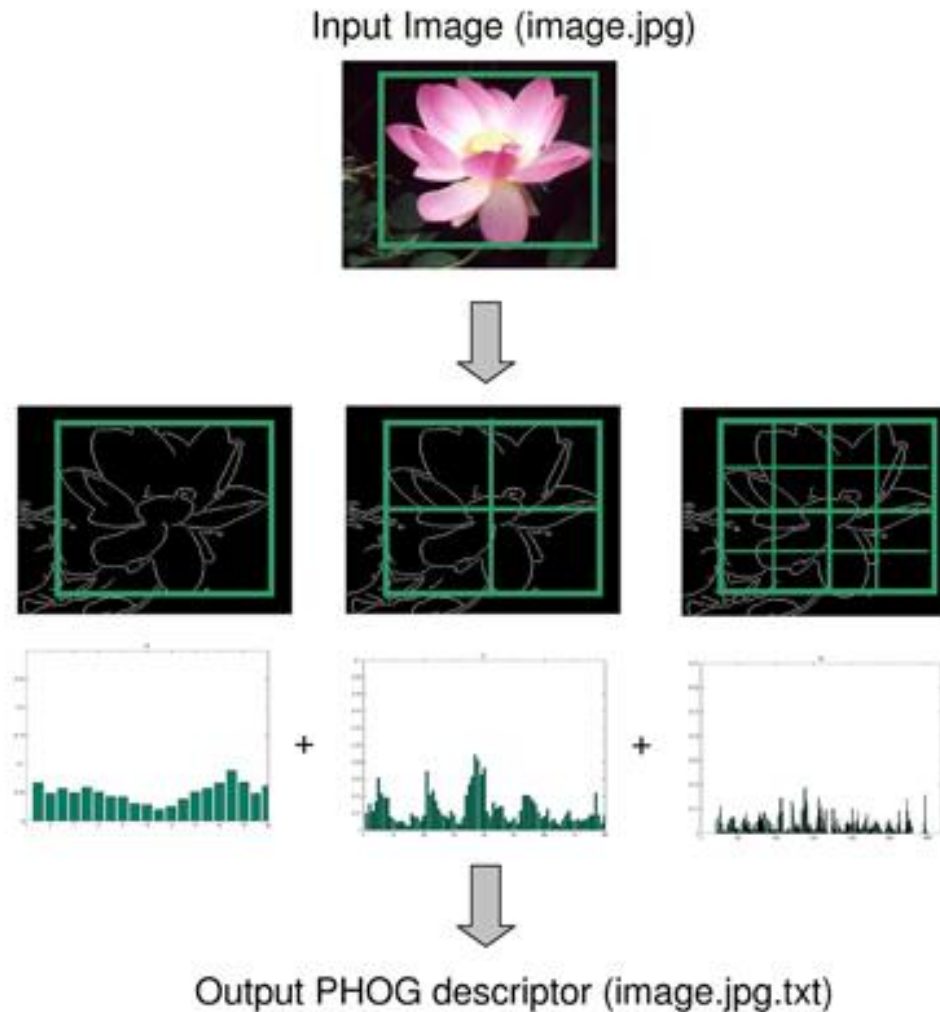
Implementierung

Erkennung von Fußgängern

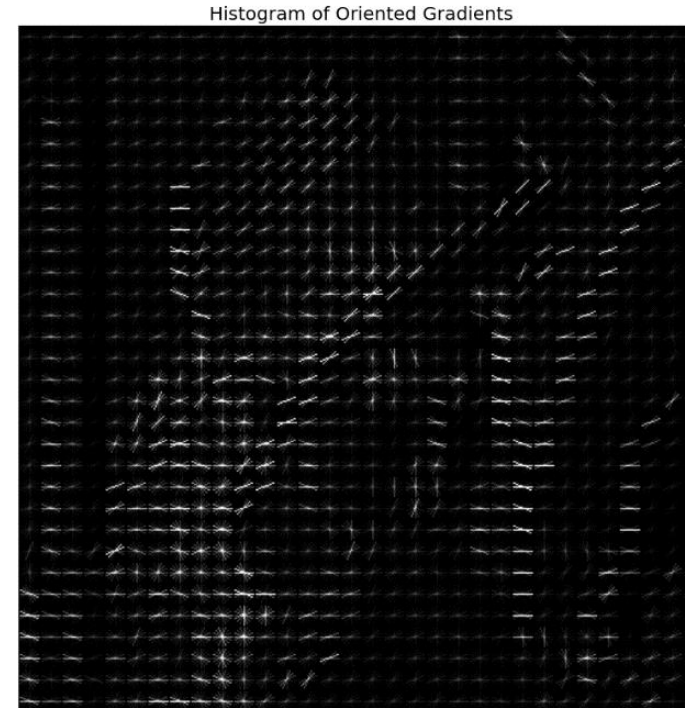
Vorbereitung

- Mögliche Algorithmen:
 - Histogram oriented Gradient (HoG)
 - Support Vector Machine (SVM)
- Klassifizierte Bilderdatenbank der entsprechenden Objekte wird benötigt
- Erkennen und klassifizieren von Objekten recht rechenintensiv
- Portieren von OpenCV 2.4 auf OpenCV 3(Beta)
- **Eventuelles Problem:** Rechenleistung von Einplatinenrechnern

Histogram oriented Gradient



Histogram of Oriented Gradients



Résumé

Persönliches Fazit

Résumé

- Brandaktuelles Forschungsgebiet
- Viele verschiedene Lösungswege
- Viele ungelöste Probleme
- Richtige Wahl der Suchbegriffe:
 - Mensch -> ~~human/people~~ -> pedestrian
 - Objekterkennung -> ~~Object-detection~~ -> obstacle avoidance
- Portierung von Windows auf Linux

Intel Keynote CES 2015 -> Intel RealSense



Ist die Idee umsetzbar?

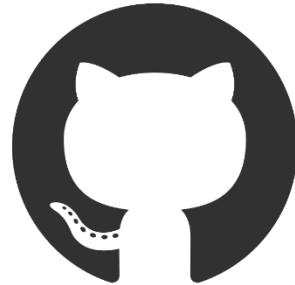
Definitiv!

Aber

Nur mit mehr Manpower



Vielen Dank für Ihre
Aufmerksamkeit



<https://github.com/Norbell/KinectObstacleAvoidance>

