Fakultät für Informatik
Technische Universität Chemnitz

Objektverfolgung durch Sensorfusion in Multisensorumgebungen

Personenzählung mit mehreren Kameras

Falk Schmidsberger und Danny Kowerko
{Falk.Schmidsberger, Danny.Kowerko}@informatik.tu-chemnitz.de

16. März 2019

Inhalt

- Personenzählung in einem geschlossenen Raum
  (z.B. öffentliches Verkehrsmittel)
- Videokameras und Frameworks zur Objekterkennung (mit Deep-Learning-Einsatz)

- Erkennung von Personen mit Deep Learning-basierten Algorithmen
- Erschwert in Szenarien mit teilweisem oder hohem Verdeckungsgrad
- Verbesserungsansatz: Fusion mehrerer Kamerasensoren mit überlappenden Sichtbereichen
- Regelbasierte und maschinell erlernte Ansätze

- Erstellen von Videoaufnahmen mit Personen in einem realitätsnahen Szenario mit mehreren Kameraperspektiven
- Personendetektion in den Videoaufnahmen mit verschieden Frameworks zur Objekterkennung
- Auswertung und Kombination der Personendetektion der verschiedenen Frameworks

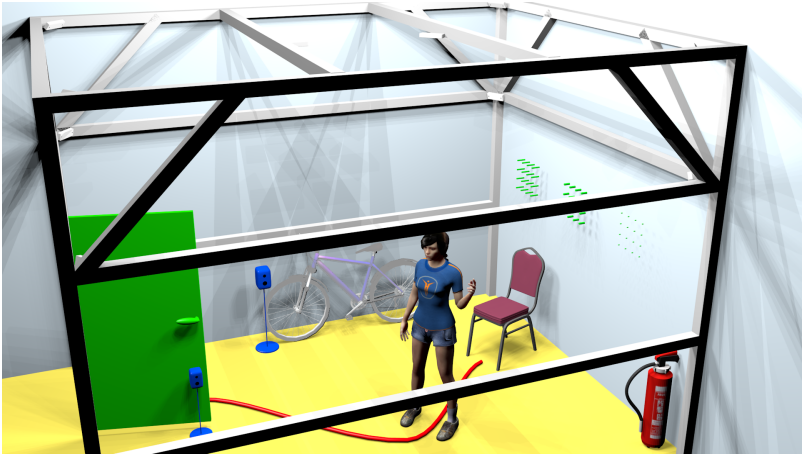


Abbildung: Visualisierung des Laboraufbaus [1]

Videoaufnahme
Laboraufbau schematisch

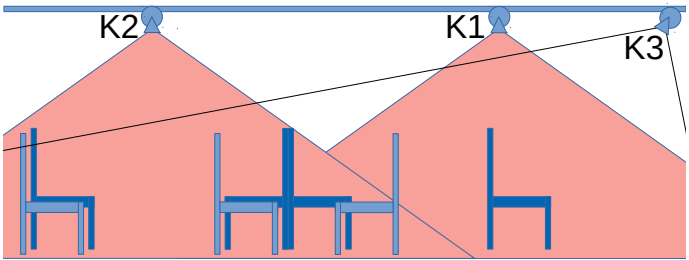


Abbildung: Laboraufbau schematisch mit 3 Kameras

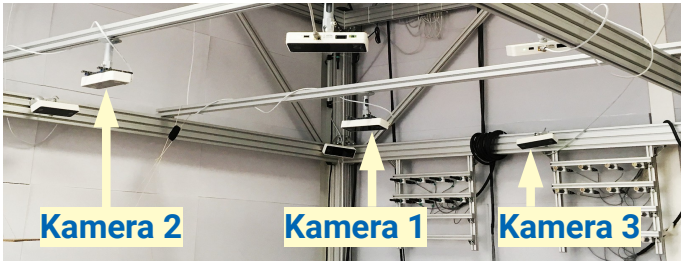


Abbildung: Laboraufbau mit 3 Kameras

- Aufnahme 1:
  - Eine Kamera (Kamera 3)
  - Gesamtansicht des Businneren
  - 3557 Frames
  - max. 6 Personen im Bus
- Aufnahme 2:
  - synchron mit allen 3 Kameras, bei geringer Überlappung der Top-Kameras im mittleren Bereich
  - Kamera 1: Ansicht des Busvorraums von oben
  - Kamera 2: Ansicht des hinteren Bussabschnitts von oben
  - Kamera 3: Gesamtansicht des Businneren
  - 1685, 1647 bzw. 1550 Frames
  - max. 4 Personen im Bus

Auswertung der Videodaten

Frameworks zur Objekterkennung (mit Deep-Learning-Einsatz)

- Nutzung von drei verschiedenen Objekterkennungsframeworks zur Detektion von Personen im Videoframe (keine Wiedererkennung):
  - OpenPose
  - Detectron
  - Yolo V3

OpenPose [2]

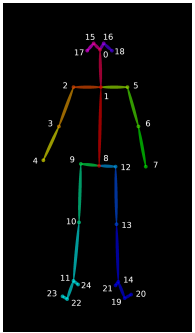


Abbildung: Ausgabeformat der Skelettstruktur (Body 25)

- Erkennung von mehreren Personen im Bild
- u.a. Skelettstruktur (25 Keypoints + jeweiligen Score-Wert)
- https://github.com/CMU-Perceptual-Computing-Lab/openpose
- Ausgabeformate unter: /openpose/blob/master/doc/output.md

Detectron [3]

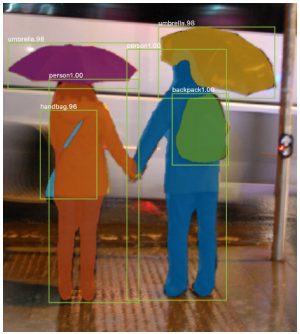


Abbildung: Ausgabe des R-CNN in Detectron [4]

- Erkennung verschiedener Objekte im Bild
- Person, Auto, Fahrrad, Katze, Hund, Pferd, Flasche, Tasse, ...
- 80 Kategorien in MS COCO [5]
- Ausgabe: Bounding Box + Label + Score-Wert
- https://github.com/facebookresearch/Detectron

Yolo V3 [6]

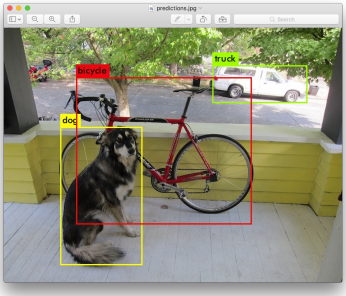


Abbildung: Yolo V3 Beispiel

Quelle: pjreddie.com/darknet/yolo/ (07.03.2019)

- Erkennung verschiedener Objekte im Bild
- Person, Auto, Fahrrad, Katze, Hund, Pferd, Flasche, Tasse, ...
- 80 Kategorien in MS COCO [5]
- Ausgabe: Bounding Box + Label + Score-Wert
- https://pjreddie.com/darknet/yolo/
- https://github.com/pjreddie/
- Implementierung in Python: https://github.com/qqwweee/keras-yolo3

Auswertung der Videodaten
Anwendung der Frameworks

- Anwendung der Frameworks auf die einzelnen Aufnahmen, mit kleinem Mindest-Score-Wert (0.0; 1.0]
- Geschwindigkeit (nur ein Framework): ca. 2 bis 8 Frames pro Sekunde
- Ergebnisse jeweils in einer Json-Datei [7] pro Videoframe
- Unterschiedliche Json-Formate der einzelnen Frameworks

Auswertung der Videodaten

Datenaufbereitung

- Vereinheitlichung der Json-Daten zu:
  Personen mit Bounding Box und Score-Wert
- Festlegung der Fläche des Businneren für jede Perspektive
- Annotation der Videoframes mit der Anzahl der Personen im Bus und der Anzahl der Personen innerhalb der Fläche des Businneren
- Synchronisierung der einzelnen Frames der verschiedenen Videoaufzeichnungen

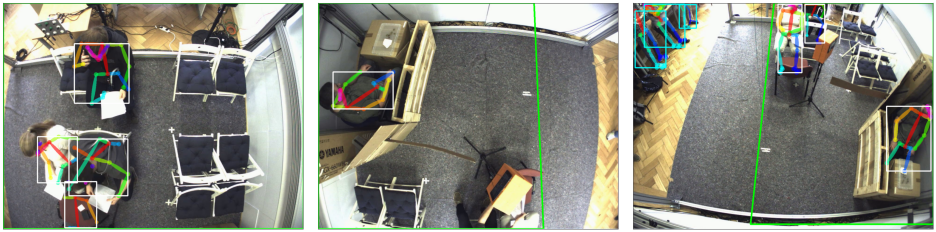


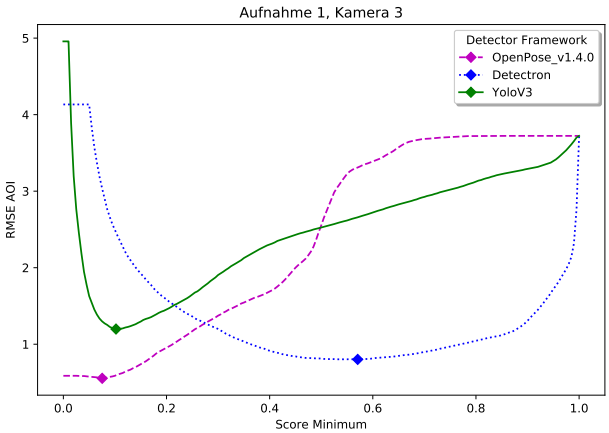
Abbildung: Laboraufbau mit 3 Kameras, das Businnere (*AOI*) ist jeweils grün umrandet

Auswertung der Videodaten
Fehlerminimierung

- Greedy-Algorithmus siehe [8] S. 161 zur Wahl des besten minimalen Score-Wertes je Framework und Kameraperspektive
- Minimierung des Root-Mean-Squared Error (RMSE) der Anzahl $n_d AOI$ der detektierten Personen innerhalb der Busfläche (AOI) im jeweiligen Video

$$RMSE_{AOI} = \sqrt{\frac{\sum_{f=1}^{framecount} (n_{dAOI,f} - n_{gAOI,f})^2}{framecount}} \tag{1}$$

Auswertung der Videodaten
Fehlerminimierung



Detector	$Score_{min}$	$MRMSE_{AOI}$
OpenPose	0.0753	0.5545
Detectron	0.5707	0.8016
Yolo V3	0.1021	1.1984

Abbildung: Minimal Root-Mean-Squared Errors ($MRMSE_{AOI}$) für Aufnahme 1

Auswertung der Videodaten

Fehlerminimierung

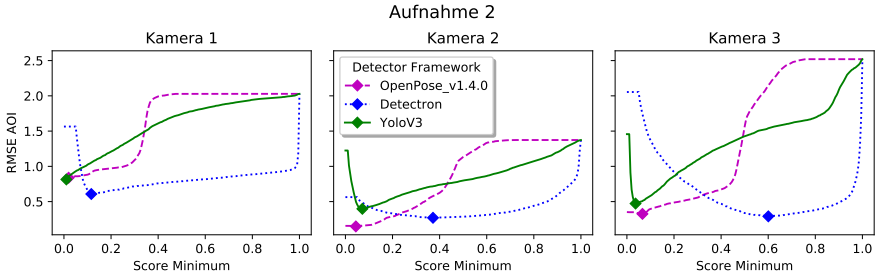


Abbildung: Minimal Root-Mean-Squared Errors ($MRMSE_{AOI}$) für Aufnahme 2

Kamera 1			Kamera 2			Kamera 3		
Detector	$Score_{min}$	$MRMSE_{AOI}$	Detector	$Score_{min}$	$MRMSE_{AOI}$	Detector	$Score_{min}$	$MRMSE_{AOI}$
OpenPose	0.0199	0.8386	OpenPose	0.0442	0.1481	OpenPose	0.066	0.3258
Detectron	0.1161	0.6077	Detectron	0.3719	0.2702	Detectron	0.6	0.291
Yolo V3	0.0103	0.8142	Yolo V3	0.0719	0.3977	Yolo V3	0.0367	0.4723

Auswertung der Videodaten

Sensorfusion

- Sensorfusion durch Trainieren von Entscheidungsbäumen siehe [8] S. 814
- Verwendung der Ausgabedaten von OpenPose, Detectron und Yolo V3 und der Annotationen der Videoframes
- Fusion der 3 Frameworks bei Aufnahme 1

O, D, Y, GT
0, 0, 0, 0
0, 0, 0, 0
0, 0, 0, 0
0, 0, 1, 0
0, 0, 1, 0
0, 0, 1, 1
0, 0, 1, 1
...
...

$MRMSE_{AOI}$ der einzelnen Framworks:

Detector	$Score_{min}$	$MRMSE_{AOI}$
OpenPose	0.0753	0.5545
Detectron	0.5707	0.8016
Yolo V3	0.1021	1.1984

RMSE durch Nutzung des besten Entscheidungsbaums: 0.3019

Auswertung der Videodaten
Sensorfusion

- Sensorfusion durch Trainieren von Entscheidungsbäumen
- Fusion der 3 Frameworks aller 3 Perspektiven bei Aufnahme 2

...
...

O, D, Y, O, D, Y, O, D, Y, GT
2, 1, 1, 2, 2, 2, 3, 3, 2, 3
1, 1, 1, 2, 2, 2, 3, 3, 2, 3
1, 3, 1, 2, 2, 2, 3, 3, 2, 3
3, 4, 2, 2, 2, 2, 3, 3, 2, 4
3, 4, 1, 2, 2, 2, 3, 3, 2, 4
3, 4, 2, 2, 2, 2, 3, 3, 2, 4
2, 4, 2, 2, 2, 2, 3, 3, 2, 4

...
...

RMSE durch Nutzung des besten Entscheidungsbaums: 0.0154

Weitere Arbeiten

- Bisher nur Annotationen zur Anzahl von Personen verwendet
- Annotationstools zur Korrektur von Bounding Boxen und Posen vorbereitet
- Einbeziehung weiterer Algorithmen zur Objekt / Personenerkennung / Kopferkennung
  $\Rightarrow$ Personen und Objekte können über mehrere Kameras verfolgt werden.
- Einbeziehung weiterer Algorithmen zur Sensorfusion
- Im Zusammenhang mit den vorgenannten Methoden sollen die folgenden Fragen beantwortet werden:
  - Wie viele Kameras/Sensoren sind für das Szenario notwendig?
  - Welche räumliche Anordnung ist optimal (absolute Position und Ausrichtung)?
  - Welche Algorithmen ermöglichen die Verfolgung/Erkennung von Personen in Multikameraaufnahmen durch Fusion der Metadaten aus den Einzelvideos, zur Ermittlung der Anzahl der Personen.
  - Interpretation der aktuellen Fahrgastsituation mit Hilfe der Ortungsdaten und den Gegebenheiten im Verkehrsmittel (stehend/sitzend, Position im Verkehrsmittel).

Zusammenfassung

- Im Labor der Juniorprofessur Mediacomputing der TU Chemnitz wurde eine Multikameraumgebung eingerichtet und Aufnahmen für ein Bussszenario aufgenommen.
- Eine erste Analyse der Daten für die Einzelvideos und eine Sensorfusion war erfolgreich. Die Arbeiten werden entsprechend fortgesetzt.

Referenzen I

[1] Robert Manthey, Rico Thomanek, Christian Roschke, Tony Rolletschke, Benny Platte, Marc Ritter, and Danny Kowerko.
Visual system examination using synthetic scenarios.
In Waldemar Karwowski and Tareq Ahram, editors, *Intelligent Human Systems Integration 2019*, pages 418–422, Cham, 2019. Springer International Publishing.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh.
OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields.
In *arXiv preprint arXiv:1812.08008*, 2018.

[3] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He.
Detectron.
`https://github.com/facebookresearch/detectron`, 2018.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick.
Mask R-CNN.
CoRR, abs/1703.06870, 2017.

Referenzen II

[5] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick.
Microsoft COCO: common objects in context.
CoRR, abs/1405.0312, 2014.

[6] Joseph Redmon and Ali Farhadi.
Yolov3: An incremental improvement.
arXiv, 2018.

[7] The JSON Data Interchange Format.
Technical Report Standard ECMA-404 1st Edition / October 2013, ECMA, October 2013.

[8] Stuart J. Russell and Peter Norvig.
Künstliche Intelligenz: Ein moderner Ansatz.
Pearson, Higher Education, München, 3., aktualisierte auflage edition, 2012.

Vielen Dank! Fragen?