

# Bewegungserkennung aus Bildfolgen

---

Julian Götz, Philipp Kleinhenz

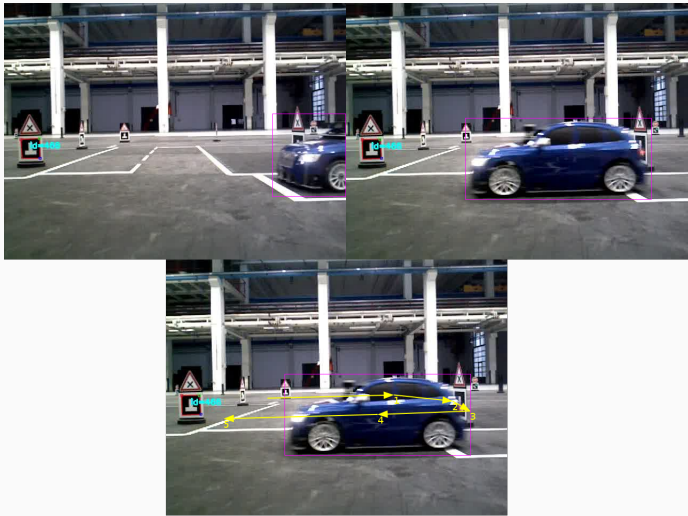
Leipzig, 25. Juni 2017

Hochschule für Technik, Wirtschaft und Kultur Leipzig

**Eingabe:** zwei oder mehrere Bilder mit einem in allen Bildern markierten Objekt

**Ausgabe:** (relative) Bewegungsrichtung und -geschwindigkeit der Objekte

- Trivial
- Vektor zwischen Mittelpunkten der Bounding-Boxen
- Geschwindigkeit = Betrag des Vektors
- Bewegungsrichtung = Richtung des Vektors
- ;) ...im Bildraum :(
- Warum einfach, wenn es auch schwer geht?
- $\Rightarrow$  Objekterkennung durch optischen Fluss (Lucas-Kanade, TV-L1, ...



- Lucas-Kanade
  - kleine Bewegungen nötig
  - verfolgt Einzelpunkte
  - gute Features nötig

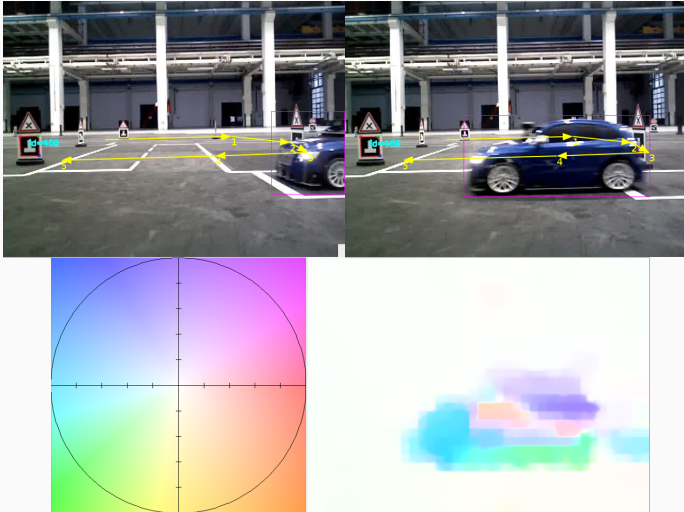
- mittels OpenCV
  - JavaCV - Javabindings für OpenCV
  - eigene Methoden für ImageJ-OpenCV-Interoperabilität
- implementiert verschiedene Optische-Fluss-Algorithmen  
SparseToDense, Farnebäck, DeepFlow, DIS<sup>1</sup>, DualTVL1
- Eingabe: zwei Bilder
- Ausgabe: 2D-2D-Vektorfeld

---

<sup>1</sup>Dense Inverse Search

# Fluss-Visualisierung

- Vektoren bekommen Farbwert zugewiesen
- Wahl der Farbe über Farbrad



```
ImageJ -macro /flowig/macro/runplug.ijm  
Flowig#parameter=<wert>,...
```

- path Pfad zur Bildfolge
- flow Algorithmus zur Flussberechnung
- showx und showy Visualisierung x bzw. y Komponenten
- boundscolor: Farbe der Bounding-Boxen
- maxmotion: Maximalwert der Farbinsensitivitätskalibrierung
- scalesize: Anzahl Skalierungen
- scalecolor: Sättigungsmultiplikator



- Entfernen der Ego-Bewegung
- Clustern anhand ähnlicher Bewegungsvektoren
- um Objekte ohne Bounding-Boxen zu finden