

Hochschule für Technik, Wirtschaft und Kultur Leipzig

Fakultät Informatik, Mathematik und Naturwissenschaften

16IMN

Bewegungserkennung aus Bildfolgen

Digitale Bildverarbeitung

Eingereicht von: Philipp Kleinhenz, Julian Götz

Leipzig 23. Juni 2017

Inhaltsverzeichnis

1	Programmfunktion	1
1.1	Installation	1
1.2	Nutzung	2
1.3	Aufruf über Kommandozeile	2
2	Weitere Informationen	6

1 Programmfunktion

Das vorliegende ImageJ-Plugin Flowig dient zur Erkennung der Bewegung eines markierten Objektes in Bildfolgen. Es wird für jedes Bild die Bewegung des Objekts im Vergleich zum vorherigen Bild erkannt. Zudem wird der optische Fluss berechnet.

Eingabe: Ein Ordner, in dem sich eine Bildfolge befindet. Ist ein Objekt in einem Bild vorhanden, muss dieses durch Bounding-Boxen markiert sein.

Ausgabe: Bewegungsrichtung und Geschwindigkeit des Objekts in jedem Bild

Anzeige: Bewegungsrichtungen auf jedem Bild als Overlay und Darstellung des optischen Flusses

1.1 Installation

Die Datei `Flowig_.java` muss in einem eigenen Unterordner `flowig` in ImageJ's Plugin Ordner liegen.

Flowig nutzt die Bibliothek JavaCV ¹. Damit auf die Bibliothek zugegriffen werden kann, müssen zuerst die Dateien `javacpp.jar`, `javacv.jar`, `opencv.jar` und `opencv-linux-x86_64.jar` in den Ordner `ImageJ/plugins/jars` kopiert werden.

Des Weiteren muss die Version des genutzten Plugin-Compilers auf 1.8 eingestellt sein. Die aktuelle Version lässt sich in ImageJ über folgendes Menü einsehen:

Edit → Options → Compiler...

¹<https://github.com/bytedeco/javacv>

1.2 Nutzung

Nachdem das Plugin gestartet wurde, muss ein Ordner gewählt werden, der eine Bildfolge enthält.

Nach Anwenden des Plugins auf die geöffnete Bildfolge, wird in der Konsole die Bewegungsrichtung, sowie die Geschwindigkeit des Objekts in jedem Bild ausgegeben.

Für jedes Bild wird zudem ein Overlay erstellt, das die Änderung der Position im Vergleich zum vorherigen Bild in Form von Pfeilen darstellt. Abbildung 1.2 zeigt ein beispielhaftes Overlay. Dort wird die Bewegung des Objekts von Abbildung 1.1 zu Abbildung 1.2 dargestellt.

Der beispielhafte optische Fluss zu Bild 1.2 wird in Abbildung 1.3 dargestellt. Jeder Pixel bekommt einen Farbwert zugewiesen. Das Zuweisen geschieht über die Auswahl einer Farbe in einem Farbrad. Die Richtung des Vektors des Pixels im optischen Fluss bestimmt den Farbton, der Betrag des Vektors bestimmt die Sättigung. Es wird zudem eine Legende angezeigt, ein beispielhaftes Farbrad wird in Bild 1.4 gezeigt.

1.3 Aufruf über Kommandozeile

Damit das Plugin über die Kommandozeile aufgerufen werden kann, muss man es mindestens einmal über **Compile and Run...** ausführen. Daraufhin kann man es über folgenden Befehl aufrufen:

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

Mögliche Parameter:

`path=dir` ist der Ordner, der die Bildfolge enthält.

1 Programmfunktion



ABBILDUNG 1.1: Ein Eingabebild ohne Overlay

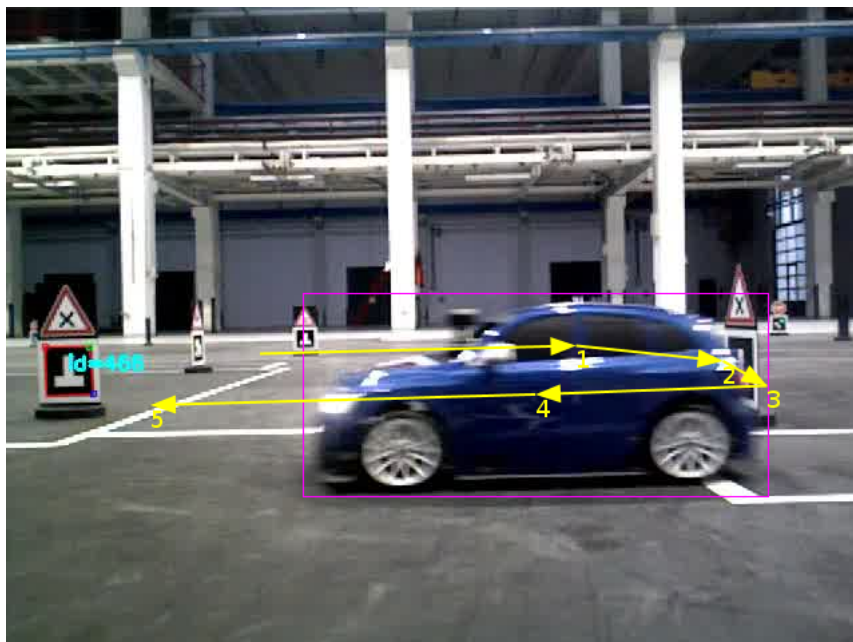


ABBILDUNG 1.2: Bewegung zu Beispiel 1

1 Programmfunktion



ABBILDUNG 1.3: Optischer Fluss zu 1.2

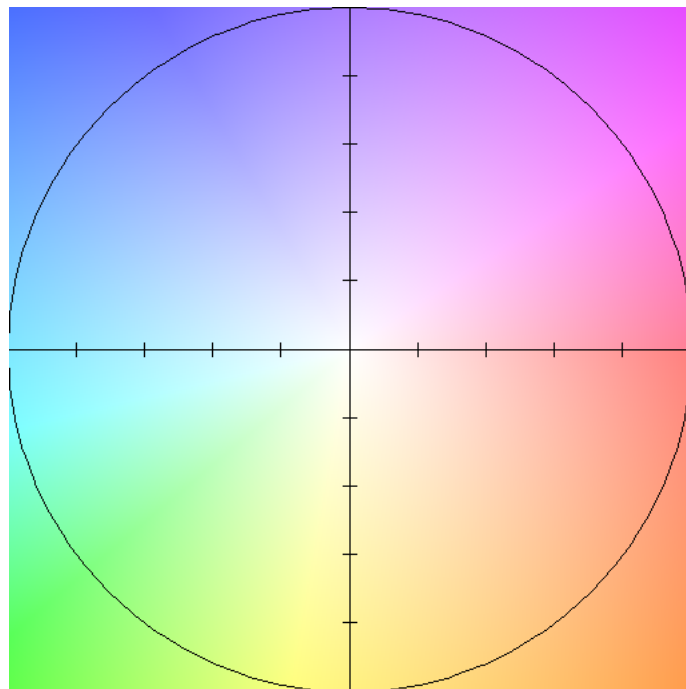


ABBILDUNG 1.4: Beispielhaftes Farbrad

1 Programmfunktion

`flow=flowType` stellt den zu nutzenden Algorithmus zur Berechnung des optischen Flusses dar, es können folgenden Typen angegeben werden:

- `SparseToDense`
- `FarneBack`
- `DeepFlow`
- `DIS`
- `DualTVL1`

Standardwert: `DIS`

`showx=true|false` und `showy=true|false` schalten die Darstellung der x beziehungsweise y Komponenten des optischen Flusses ein und aus, Standardwerte: `true`

`boundscolor` ist die Farbe der Bounding-Boxen im RGB-Format: `<r>+<g>+`, Standardwert: `255+0+255`

`maxmotion` der Maximalwert der Farbintensitätsskalierung, Standardwert: `100`

`scalesize` gibt an, wie oft das Bild herunterskaliert wird für die Flussberechnung, Standardwert: `0`

`scalecolor` gibt den Sättigungsmultiplikator an, Standardwert: `1`

Ist während der Ausführung keine grafische Oberfläche erwünscht, so ist `-macro` durch `-batch` zu ersetzen.

2 Weitere Informationen

- Fluss-Visualisierung (bzw. 2D-Vektor-Feld Visualisierung)
https://github.com/opencv/opencv/blob/master/samples/cpp/tvl1_optical_flow.cpp
- Dichter Optischer Fluss
TVL1: <http://www.ipol.im/pub/art/2013/26/article.pdf>
FarneBack: <http://lmi.bwh.harvard.edu/papers/pdfs/gunnar/farnebackSCIA03.pdf>
- JavaCV - Java OpenCV Bindings
JavaCV: <https://github.com/bytedeco/javacv>
- BlendComposite - Mischmodi für überlagernde Bilder
<http://www.curious-creature.com/2006/09/20/new-blendings-modes-for-java2d/comment-page-1/>