

Cimarron

Stabilization of videos in modern C++

Praktikumsabschlusspräsentation

Marius Herget

Institut für Informatik, LMU München

12. Oktober 2018

Videostabilisierung Methodik um ungewollte Bewegungen (Wackler) der Kamera in einem Video zu reduzieren

C++ eignet sich besonders gut, da Konzepte formuliert werden koennen, auf deren Grundlage eine Anwendung implementiert werden kann und die Speicherverwaltung deterministisch und minimal ist.

Ziel Entwicklung einer Programmierabstraktion zum Kompensieren von ungewollten Bewegungen (Translation und Rotation) der Kamera mit Hilfe von Feature Tracking

DEMO

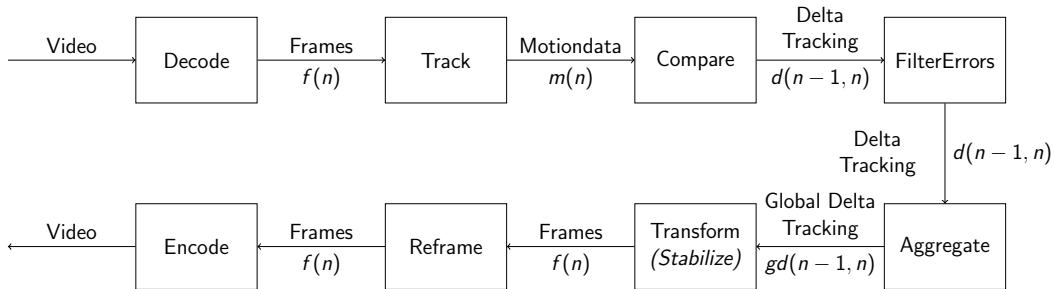


Abbildung: High-level system diagram

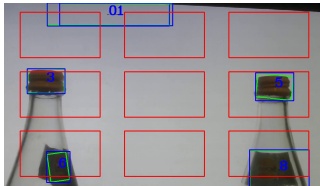
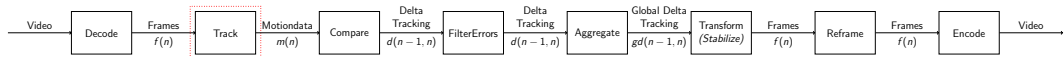
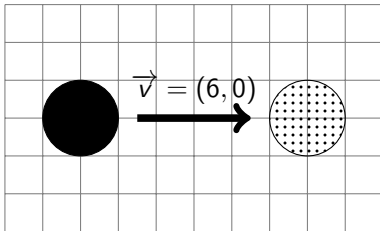


Abbildung: Beispiel

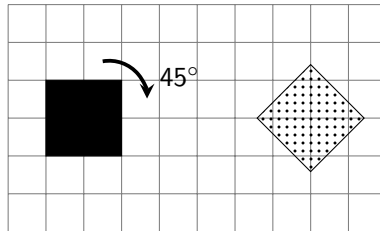
CAMshift Algorithmus von OpenCV [3]:

1. Meanshift
2. Fenstergrösse und Orientierung anpassen
3. Wiederhole 1. bis gewünschte Genauigkeit erreicht ist

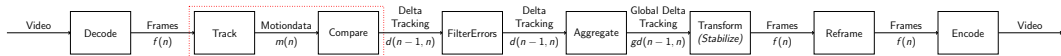




(a) Translation



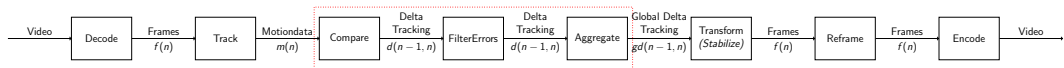
(b) Rotation



Lokal $\Delta(\text{trackingVector}(F(n-1)), \text{trackingVector}F(n))$

Global

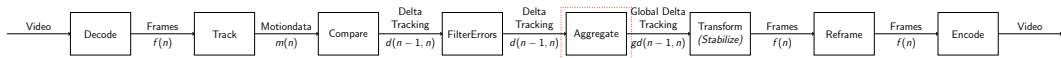
1. Ähnlichkeitmass berechnen
2. Ueberpruefen ob genug Vektoren / Winkel aehnlich genug sind
3. Durchschnittswert der aehnlichen Faktoren berechnen und daraus Inverse bilden

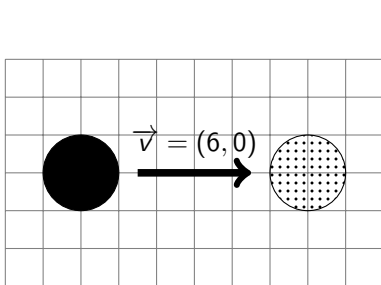


Vektoren Kosinus-Ähnlichkeit

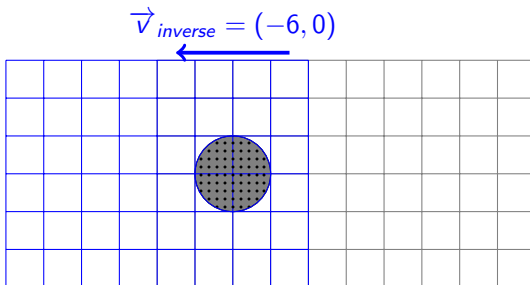
$$\cos_sim = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Winkel Prozentuale Veraenderung



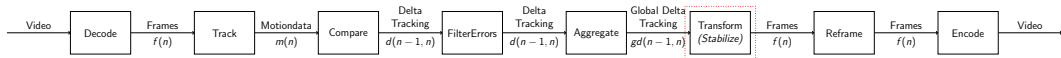


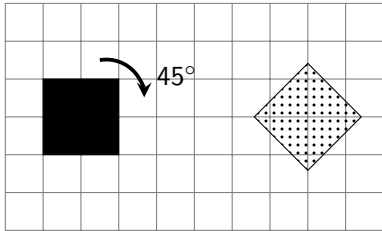
(a) Erkennung



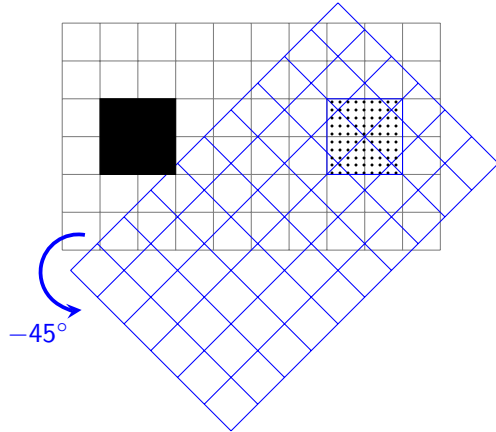
(b) Inverse

Abbildung: Translationsstabilisierung

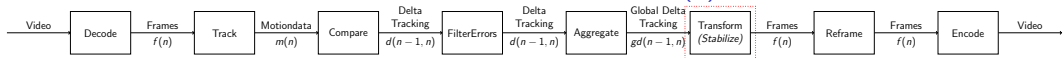




(a) Erkennung



(b) Inverse



Frame 2-Dimensionale Matrix zur Speicherung von Pixeldaten [1]

TrackingWindow Rotierendes Rechteck, welches die *Features* verfolgt [2]

Concept	Valid Expressions	Semantics	Dependent Types
Frame			
TrackingWindow			

Smoothing Die Stabilisierung ist aktuell noch sehr harsch. Dies kann durch ein *Smoothing* der globalen Delta Vektoren optimiert werden.

Reframing Es entstehen aktuell noch schwarze Ränder, die bei der Stabilisierung genutzt werden. Durch eine Analyse welche Pixel immer belegt sind, kann das Video beschnitten werden, um eine bessere Nutzererfahrung zu bieten. Möglich ist es auch, die entstehenden Ränder automatisch mit Pixeln zu ergänzen, um ein stimmiges Gesamtbild ohne *Crop* zu erzeugen [4].

- [1] [OpenCV](https://docs.opencv.org/3.1.0/d3/d63/classcv_1_1Mat.html). *cv::Mat Class Reference*. https://docs.opencv.org/3.1.0/d3/d63/classcv_1_1Mat.html. (Accessed on 10/12/2018).
- [2] [OpenCV](https://docs.opencv.org/3.4/db/dd6/classcv_1_1RotatedRect.html). *cv::RotatedRect Class Reference*. https://docs.opencv.org/3.4/db/dd6/classcv_1_1RotatedRect.html. (Accessed on 10/12/2018).
- [3] [OpenCV](https://docs.opencv.org/3.4/db/df8/tutorial_py_meanshift.html). *Meanshift and Camshift*. https://docs.opencv.org/3.4/db/df8/tutorial_py_meanshift.html. (Accessed on 10/12/2018).
- [4] [Chongwu Tang u. a.](#) *A fast video stabilization algorithm based on block matching and edge completion*. Okt. 2011.

BACKUP

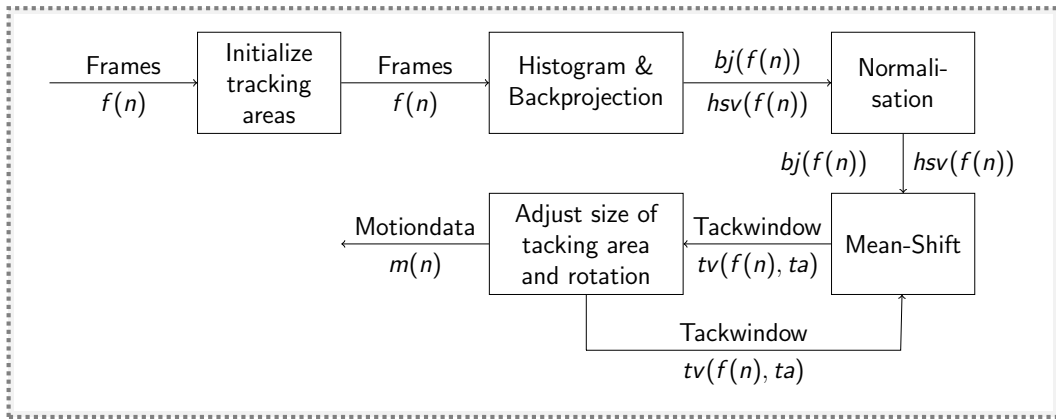


Abbildung: Detailed system diagram of *Track*