

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

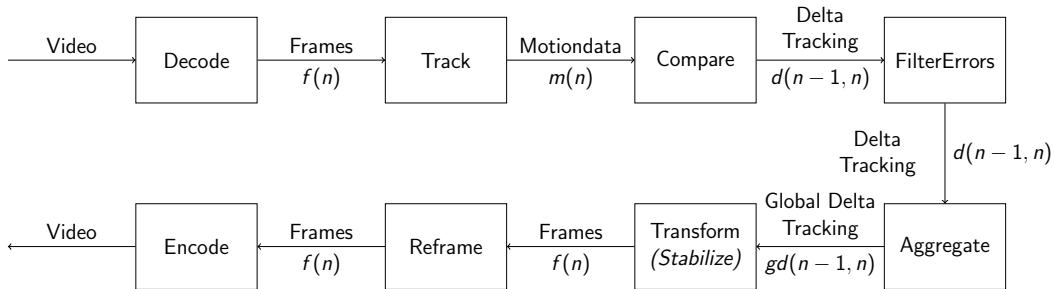


Abbildung: High-level system diagram

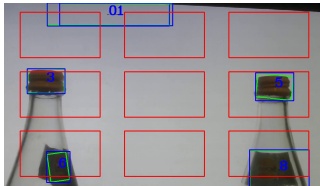
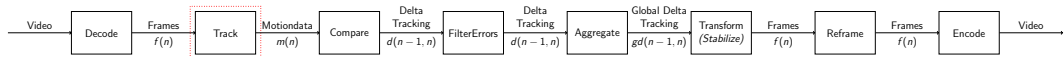
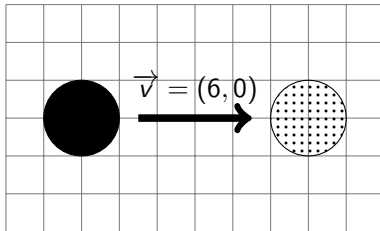


Abbildung: Beispiel

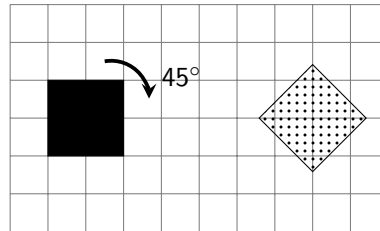
CAMshift Algorithmus von OpenCV [1]:

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

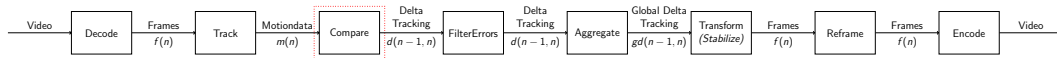




(a) Translation

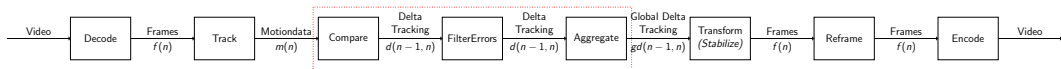


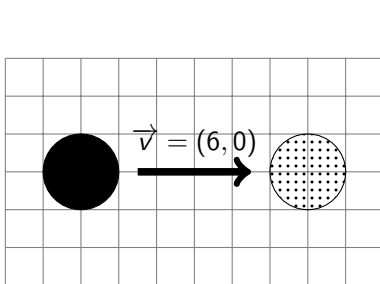
(b) Rotation



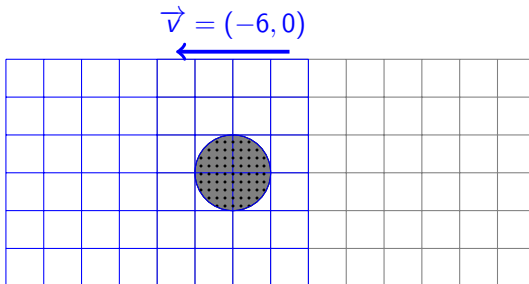
Lokal $\Delta(\text{trackingVector}(F(n-1)), \text{trackingVector}(F(n)))$

Global



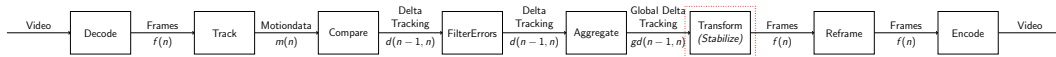


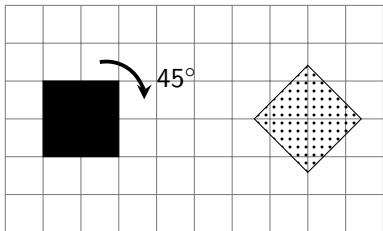
(a) Erkennung



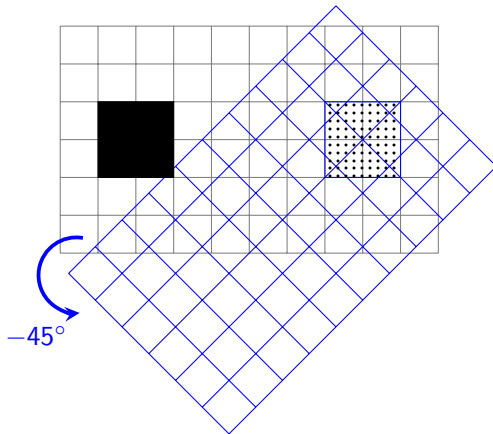
(b) Inverse

Abbildung: Translationsstabilisierung

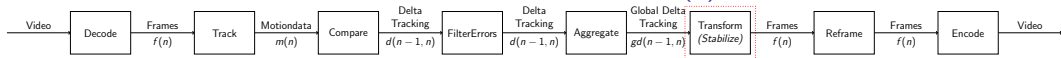




(a) Erkennung



(b) Inverse



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

Vektoren: Berechne die Cosine similarity von jedem Tracking window mit jedem anderen TrackingVector:

$$cos_sim(\forall transVector(delta(f(n), f(n + 1))))$$

BACKUP

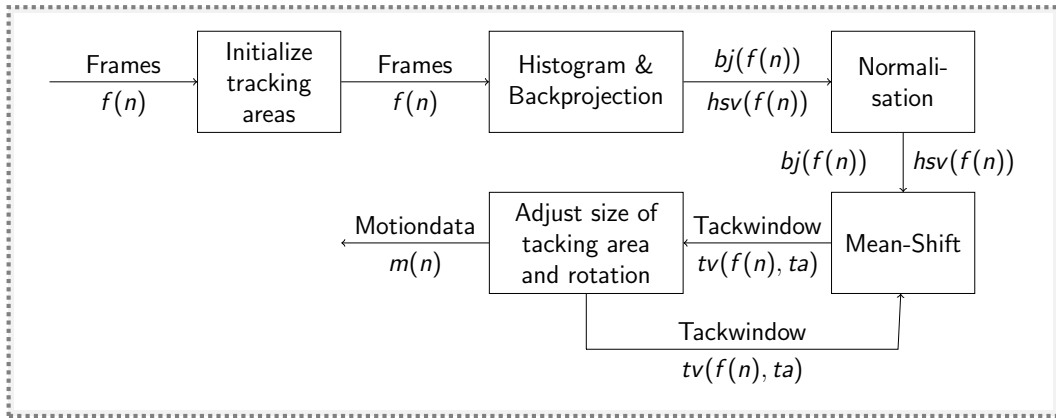


Abbildung: Detailed system diagram of *Track*