

Computer Vision: Optische Flüsse

D. Schlesinger – TUD/INF/KI/IS

- Bewegungsanalyse $\leftrightarrow$ Optischer Fluss
- Lokale Verfahren (Lukas-Kanade)
- Globale Verfahren (Horn-Schunck)
(+ kontinuierliche Ansätze: mathematische Grundlagen)
- Erweiterungen, Probleme ...

Einsatzbereiche:

- Detektion bewegter Objekte (insbesondere unbekannter Objekte), z.B. bei Überwachung
- Detektion sich unabhängig bewegter Objekte (Fahrerassistenzsysteme)
- Verfolgung
- Registrierung (z.B. medizinischer Bilder)
- Flussdynamik, Physik
- Depth from Motion, Lokalisierung und 3D-Rekonstruktion, Robotik, SLAM
- Superresolution, Deblurring
- Videokompression, Videobearbeitung (Segmentierung, Effekte ...)

Methoden:

- Dicht, Undicht (besondere Punkte + Bildmerkmale + Matching)
- Dicht $\rightarrow$ Diskret, Kontinuierlich
- Dicht, Kontinuierlich $\rightarrow$ Optische Flüsse (heute)

Es ist im allgemeinen Fall nicht möglich,
die „echte Bewegung“ aus den Bildern zu ermitteln
– ein **schlecht-gestelltes** (ill-posed) Problem.

Aperturproblem:



Die Bewegungskomponente entlang der Kanten kann beliebig sein
 $\Rightarrow$ Kanten unterschiedlicher Richtungen müssen anwesend sein.

Sei $I : \mathbb{R}^2 \times \mathbb{Z} \rightarrow \mathbb{R}$ die Bildfunktion,

$I(x, y, t)$ ist der Wert an der Stelle (x, y) (kontinuierlich) zum Zeitpunkt t (diskret)

(u, v) ist das Verschiebungsfeld, d.h.

$u : \mathbb{R}^2 \times \mathbb{Z} \rightarrow \mathbb{R}$ und $v : \mathbb{R}^2 \times \mathbb{Z} \rightarrow \mathbb{R}$ sind die Verschiebungen in x - bzw. y -Richtungen.

Optic Flow Constraint (die korrespondierenden Punkte sind gleich):

$$I(x + u(x, y, t), y + v(x, y, t), t + 1) = I(x, y, t)$$

Kürzel: $I(x + u, y + v, t + 1) = I(x, y, t)$.

Problem: $I(x + u, y + v, t + 1)$ ist nicht linear in $(u, v) \Rightarrow$ linearisieren:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + I_x u + I_y v + I_t$$

mit räumlichen (I_x und I_y) und zeitlicher (I_t) Ableitungen der Bildfunktion.

Daraus folgt:

$$I_x u + I_y v + I_t = 0$$

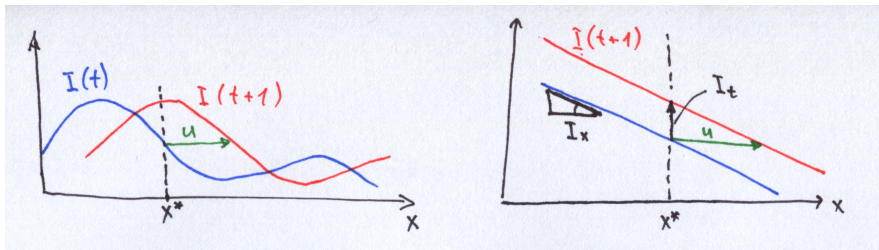
Eine Gleichung, zwei Unbekannte u und v – keine eindeutige Lösung!!!

(erinnere an das Aperturproblem)

Optic Flow Constraint

$$I_x u + I_y v + I_t = 0$$

Beispiel für 1D:



Das funktioniert nur, wenn die Verschiebung kleiner ist, als eine „charakteristische Größe“ – die dominante Frequenz der Bildfunktion.
zum Beispiel wenn Bilder relativ glatt sind und die Verschiebung relativ klein.

Interessant: Im 1D gibt es kein Aperturproblem
– eine Gleichung und eine unbekannte Verschiebung.

Annahme: die Verschiebungen innerhalb eines kleinen Fensters (lokal) sind konstant:

$$I_x(x', y', t)u + I_y(x', y', t)v + I_t(x', y', t) = 0 \quad \forall x', y' \in F(x, y)$$

– ein überbestimmtes Gleichungssystem $\Rightarrow$ summarische Abweichungen minimieren:

$$\sum_{x', y' \in F(x, y)} \left(I_x(x', y', t)u + I_y(x', y', t)v + I_t(x', y', t) \right)^2 \rightarrow \min_{u, v}$$

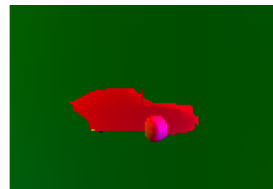
Ableiten, null setzen, auflösen $\Rightarrow$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

Etwas dazu – die Summanden können mit einem Gausschen Kern gewichtet werden.

Das System ist nur dann eindeutig lösbar, wenn es nicht entartet ist (linear unabhängige Gleichungen) – wieder das Aperturproblem.

Beispiel:



Nachteile:

- Zu restriktive Annahmen (lokal konstant) insbesondere für die **Diskontinuitäten**
- Keine transparente Modellierung des Flussfeldes
- Unabhängige Berechnungen für jeden Ort (x, y)

Vorteile:

- Schnell und einfach

Die erste Annahme ist dieselbe – Optic Flow Constraint:

$$I_x u + I_y v + I_t = 0$$

Die zweite Annahme – das Flussfeld soll möglichst glatt sein, d.h.

$$\int_{\Omega} |\nabla u|^2 + |\nabla v|^2 dx dy$$

soll möglichst klein sein (Ω ist der Definitionsbereich, ∇ ist Gradient).

Alles zusammen:

$$E(u, v) = \int_{\Omega} \left[(I_x u + I_y v + I_t)^2 + \alpha (|\nabla u|^2 + |\nabla v|^2) \right] dx dy \rightarrow \min_{u, v}$$

Die Optimierung wird im **Funktionsraum** durchgeführt (alle mögliche Funktionen (u, v) werden betrachtet).

$E(u, v)$ ist ein Maß für die Bewertung der Funktionen – **Energiefunktional**.

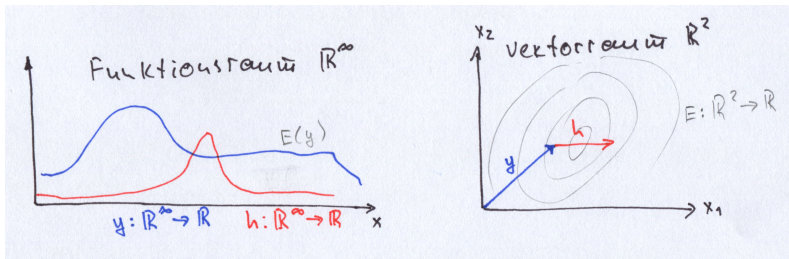
Eine solche Optimierungsaufgabe nennt man oft **Kontinuierliche Energieminimierung**.

Calculus of Variations, Variationelle Ansätze

Sei $y : \Omega \rightarrow \mathbb{R}$ eine Funktion – ein „Vektor“ im unendlichdimensionalen Raum. Energiefunktional $E(y)$ ordnet jeder Funktion y eine Zahl zu, d.h. $E : \mathbb{R}^\infty \rightarrow \mathbb{R}$.

Gâteaux Ableitung entlang einer „Richtung“ $h : \Omega \rightarrow \mathbb{R}$:

$$\left. \frac{\partial E(y)}{\partial y} \right|_h = \lim_{\varepsilon \rightarrow 0} \frac{E(y + \varepsilon h) - E(y)}{\varepsilon} = \left. \frac{dE(y + \varepsilon h)}{d\varepsilon} \right|_{\varepsilon=0}$$



Euler-Lagrange Gleichungen: alle Ableitungen = 0 (für alle Richtungen h)
– notwendige Bedingungen für das Optimum.

Energiefunktional:

$$E(u, v) = \int_{\Omega} \left[(I_x u + I_y v + I_t)^2 + \alpha (|\nabla u|^2 + |\nabla v|^2) \right] dx dy \rightarrow \min_{u, v}$$

Gâteaux Ableitungen schreiben, null setzen, auflösen:

$$(I_x u + I_y v + I_t) I_x - \alpha \Delta u = 0$$

$$(I_x u + I_y v + I_t) I_y - \alpha \Delta v = 0$$

mit dem **Laplacian**: $\Delta u(x, y) = \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2}$ ($\Delta v(x, y)$ analog)

– System aus unendlich vielen Gleichungen (weil für alle $(x, y) \in \Omega$).

Jetzt wird **diskretisiert**:

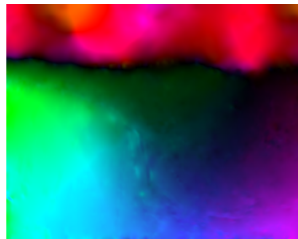
- nur die Werte in ganzzahligen Positionen (Pixel) werden berechnet
- alle Ableitungen (sowie Laplaciane) werden durch **finite Differenzen** ersetzt

Es entsteht ein System linearer Gleichungen $Ax = b$ mit einer dünn besetzten Matrix A

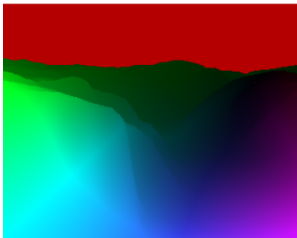
- $2 \times$ (die Anzahl der Pixel) Variablen, genauso viele Gleichungen
- wird meist durch iterative Verfahren gelöst.



Input sequence



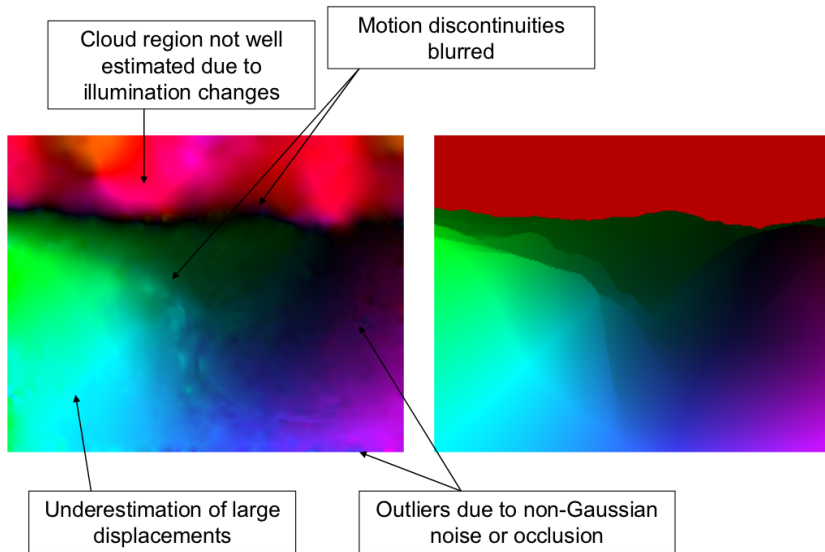
Horn-Schunck (7.17°)



Ground truth



Lucas-Kanade (8.78°)



Verwendung **robuster** Funktionen – Regularizatoren:

$$E(u, v) = \int_{\Omega} \left[(I_x u + I_y v + I_t)^2 + \alpha \Psi(|\nabla u|^2 + |\nabla v|^2) \right] dx dy \rightarrow \min_{u, v}$$

mit einem Regularisator Ψ :

$\Psi(s^2) = s^2$	– Tikhonov
$\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$	– Total Variation (bei $\epsilon = 0$)
$\Psi(s^2) = 1 - \lambda^2 \exp(-\frac{s^2}{\lambda^2})$	– Perona-Malik
$\Psi(s^2) = \begin{cases} 0 & \text{wenn } s^2 = 0 \\ 1 & \text{else} \end{cases}$	– Potts-Modell

Diskontinuitäten werden weniger stark bestraft – bleiben erhalten.

Unabhängigkeit zur Beleuchtungsänderung

– Gradienten der Bildfunktion statt Bildfunktion selbst (oder zusammen).

Große Verschiebungen:

- quadratische Approximation statt linearer – ein anderer Optic Flow Constraint
- Auflösungspyramiden

Glattheit in Zeit berücksichtigen – Ableitungen von u und v nach t bestrafen

Große Verschiebungen:

- Auflösungspyramide:
 - grobe Auflösung – kleine Objekte verschwinden
 - feine Auflösung – das Problem mit der Linearisierung (zu ungenau)
- Quadratische Approximation
 - nicht konvexe Energiefunktionale, nichtlineare Euler-Lagrange Gleichungen ...
 - schwierig zu kämpfen, meist zeitaufwendig

Lernen:

Je komplizierter ist das Modell (der zu optimierende Energiefunktional in diesem Fall), desto mehr freier Parameter hat es – sie müssen gelernt werden.

Bekannte Prinzipien des Machine Learning (Wahrscheinlichkeitstheorie, Maximum Likelihood Prinzip etc.) sind schwierig zu verwenden, denn es gibt kein Wahrscheinlichkeitsmaß für Funktionsräume.

Literatur (Namen):

Joachim Weickert, Daniel Cremers, Tomax Brox, Andres Bruhn ...