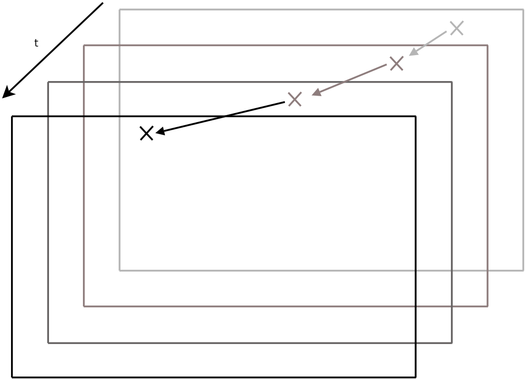
Optical Flow Estimation

Andreas Töscher   Christian Reinbacher

2. Mai 2007

Was ist Bewegung?

- Bewegung auf 3D-Pfad
- Projektion auf zwei Dimensionen
- Optical Flow Estimation = 2D-Bewegungsschätzung



Methoden der Bewegungsschätung

- Feature Tracking
- Block Matching
- gradientenbasierte Verfahren

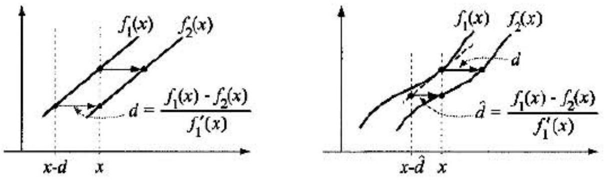
Grundlagen der gradientenbasierten Verfahren

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

Annahmen:

- Helligkeiten sind konstant
- nicht reflektierende Oberflächen
- punktförmige Lichtquelle
- keine Rotationen oder Schatten

Schätzer für 1D-Bewegungen



- $f_2(x) = f_1(x - d)$
- Taylorreihenentwicklung von $f_1(x - d)$:
  $f_1(x - d) = f_1(x) - df_1'(x) + O(d^2 f_1'')$
- Terme höherer Ordnung verwerfen, einsetzen, umformen:
  $\hat{d} = \frac{f_1(x) - f_2(x)}{f_1'(x)}$

Schätzer für 2D-Bewegungen

Analog zum 1D-Fall:

- Taylorreihenentwicklung:
  $I(x+u, t+1) \approx I(x,t) + u \cdot \bigtriangledown I(x,t) + I_t(x,t)$
- Terme höherer Ordnung verwerfen, einsetzen, umformen:
  $\bigtriangledown I(x,t) \cdot u + I_t(x,t) = 0$
- Problem: eine Gleichung in 2 Unbekannten u_1 und u_2
- Lösung: lokale Umgebung betrachten $\rightarrow$ LS-Estimator

LS-Estimator

$E(u) = \sum_X g(x)[u \cdot \bigtriangledown I(x,t) + I_t(x,t)]^2$

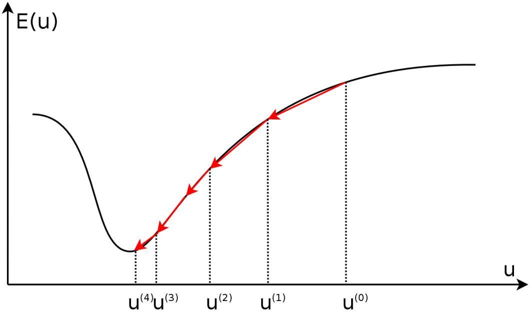
- $E(u)$...Fehlermaß auf lokaler Umgebung
- $g(x)$...2D-Gauss-Kern
- Minimum finden: ableiten nach u_1 und u_2 und Nullsetzen
- in Matrixform: $Mu = b$ wobei

$$M = \begin{bmatrix} \sum g{I_x}^2 & \sum g I_x I_y \\ \sum g I_x I_y & \sum g{I_y}^2 \end{bmatrix}, b = -\begin{pmatrix} \sum g I_x I_t \\ \sum g I_y I_t \end{pmatrix}$$

- Lösung: $\hat{u} = M^{-1} b$

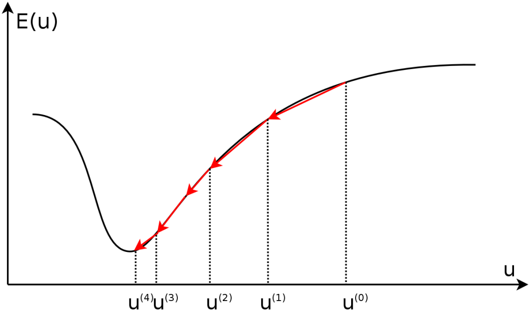
Iterative Methode (I)

- die vorherige Lösung ist optimal, aber nicht für das ursprüngliche Problem
- es treten höhere Ableitungen von $f(x)$ auf, die nicht verschwinden
- Lösung: suche lokales Minimum der Errorfunktion $E(u)$



Iterative Methode (II)

- $u^{(0)} = 0$
- $u^{(n)} = u^{(n-1)} + M^{-1}b$ für $n \geq 1$



Iterative Methode (III)

Vorteil:

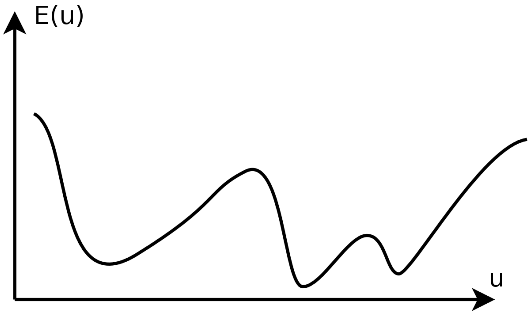
- höhere Ableitungen dürfen ungleich 0 sein

Nachteil:

- für jede Iteration muss die M Matrix invertiert werden

Temporal Aliasing

- die zeitliche Auflösung ist begrenzt
- durch eine zu schnelle Bewegung wird $u^{(0)}$ im falschen lokalen Minimum der Errorfunktion gewählt



Lösung: Coarse-To-Fine Refinement

Algorithmus:

1. Bild mit breitem Gauss glätten
2. Bewegung berechnen
3. Bewegung aus dem Bild herausrechnen
4. beginne bei (1) mit schmälerem Gauss (bis die gewünschte Genauigkeit erreicht ist)

Probleme:

- wird ein Fehler gemacht pflanzt sich dieser fort
- mit vielen kleinen Verfeinerungen ist es sehr Rechenaufwendig

Robust Motion Estimation (I)

- LS-Estimation hat Probleme mit Outlier
- Gründe für Outlier:
  - Helligkeit nicht konstant
  - Schatten
  - Reflexionen
  - mehrere Lichtquellen
  - ...

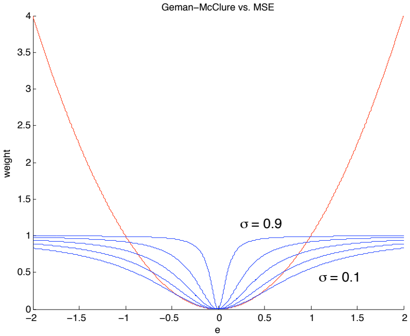
Robust Motion Estimation (II)

Lösung: Verwendung eines Robusten Gewichts

$e(x, u) = u \cdot \bigtriangledown I(x, t) + I_t(x, t)$

$\rho(e, \sigma) = e^2/(e^2 + \sigma^2)$

$E(u) = \sum_X g(x)\rho(e(x, u), \sigma)$



Layered Motion

- Probleme an den Grenzen der Objekte im Bild
- teilen des Bildes in Layer
- jeder Layer kann mit den besprochenen Methoden behandelt werden

Global Smoothing

- Annahme: Bewegung im Bild gleichmäßig, gleichförmig
- Horn-Schunck-Methode:
  $E(u) = \int (\nabla I \cdot u + I_t)^2 + \lambda(||\nabla u_1||^2 + ||\nabla u_2||^2) dxdy$
  - λ...Parameter für Bewegungsgeschwindigkeit
  - Ableitungen diskret approximieren $\rightarrow$ großes lineares Gleichungssystem
  - Lösen mit iterativen Verfahren zB *Gauss-Seidel*

Wahrscheinlichkeiten

- Problem aller Verfahren: Keine Aussage über Güte der Bewegungsschätzung
- benötigt für Fusion von verschiedenen Verfahren
- Lösung: Einführung von Wahrscheinlichkeitsrechnung: $I(x,t) = I(x+u, t+1) + \eta$
  - η... weisses Rauschen mit Mittelwert σ, unkorreliert
- damit ergibt sich: $p(I|u) \propto e^{\frac{1}{2\sigma^2}E(u)}$
  - $E(u)$...Fehlermaß der iterativen Methode
  - Mittelwert $M^{-1}b$
  - Kovarianz M^{-1} definiert Ellipse um jede Schätzung
- Keine Lösung für korreliertes Rauschen

Abschluss

Beispielprogramme zu Global Smoothing und Feature-Based-Tracker (KLT):

- OpenCV
- C

Danke für die Aufmerksamkeit