Level-Set-Methoden II

Seminar Bildsegmentierung und Computer Vision

Martin Lemmich

14. November 2005

Inhalt

Kapitel 0: Einführung und Wiederholung

Überblick über den Vortrag und kurze Wiederholung der wichtigsten Grundlagen

Ziele des Vortrags

- **Bewegung in Normalenrichtung**
  Wie kann man mit Level-Set-Methoden die Bewegung eines Objektes in Normalenrichtung darstellen? Wie lässt sich diese Bewegung mit den bisher betrachteten Bewegungen kombinieren?
- **Konstruktion vorzeichenbehafteter Abstandsfunktionen**
  Wie kann man zu einer gegebenen Kontur effektiv eine vorzeichenbehaftete Abstandsfunktion berechnen?
- **Die Partikel-Level-Set Methode**
  Wie kann man Verfälschungen des Objekts im Laufe der Bewegung reparieren?
- **Ausblick**
  Welche weiteren Einsatzmöglichkeiten für Level-Set-Methoden gibt es?

Die Level-Set-Methode

- Zweck der Level-Set-Methode ("Niveaumengenmethode"): Modellierung eines Objekts (z.B. ein Bildsegment) und effektive Berechnung der Bewegung des Objekts
- Dazu implizite Darstellung des Interface (=Kontur) des Objekts als Nullstellenmenge einer Funktion $\phi = \phi(\vec{x})$
- Anforderung an ϕ:
  - $\phi < 0$ gdw. $\vec{x}$ innerhalb des Objekts
  - $\phi = 0$ gdw. $\vec{x}$ auf der Kontur
  - $\phi < 0$ gdw. $\vec{x}$ außerhalb des Objekts
- Spezialfall: ϕ vorzeichenbehaftete Abstandsfunktion, d.h. Eigenschaften wie oben und $|\nabla\phi| = 1$ (wobei ∇ der Gradient-Operator)
- Anmerkung: $\vec{x}$ im folgenden meist aus R^2 oder R^3, Methoden bleiben im wesentlichen die Selben (auch für noch höhere Dimensionen)

Wichtige Gleichungen und Bezeichnungen

- Gradient (im Dreidimensionalen):

$$\nabla\phi(\vec{x}) = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right) \tag{1}$$

  - senkrecht zum Interface
  - zeigt in Richtung des steilsten Anstieges

- Normalenvektor:

$$\vec{N}(\vec{x}) = \frac{\nabla\phi(\vec{x})}{|\nabla\phi(\vec{x})|} \tag{2}$$

- Krümmung:

$$\kappa = \nabla \cdot \vec{N} = \frac{\partial n_1}{\partial x} + \frac{\partial n_2}{\partial y} + \frac{\partial n_3}{\partial z} \tag{3}$$

Bewegung mit der Level-Set-Methode

- Darstellung der Bewegung ebenfalls durch die implizite Funktion $\phi = \phi(\vec{x}, t)$
- Seien t^n und t^{n+1} zwei Zeitpunkte mit $t^{n+1} - t^n = \Delta t$. Schreiben $\phi(\vec{x}, t^n)$, $\phi(\vec{x}, t^{n+1})$ auch als ϕ^n bzw. ϕ^{n+1}
- Level-Set-Gleichung bei gegebenem Geschwindigkeitsfeld $\vec{V} = \vec{V}(t) = (u(t), v(t), w(t))$

$$\phi_t + \vec{V} \cdot \nabla\phi = 0 \tag{4}$$

- Diskretisierung in der Zeit (Eulersche Zeitdiskretisierung):

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \vec{V}^n \cdot \nabla\phi^n = 0 \tag{5}$$

- lässt sich auch schreiben als

$$\phi^{n+1} = \phi^n + \Delta t(-u^n\phi_x^n - v^n\phi_y^n - w^n\phi_y^n) \tag{6}$$

Beispiel

- Gegeben:
  - Interface 2-dimensionaler Kreis mit Mittelpunkt (0,0) und Radius r
  - ϕ vorzeichenbehaftete Abstandsfunktion
  - $\vec{x_1} = (r, 0)$ und $(\vec{x_1} = \frac{1}{2}\sqrt{2}(r, r))$
  - $\vec{V} = (1, 0)$ (Bewegung nach rechts)
- $t^0 = 0$: $\phi^0(\vec{x_1}) = \phi^0(\vec{x_2}) = 0$ (Punkte liegen auf dem Interface)
- $t^1 = 1$:
  - $\phi^1(\vec{x_1}) = -1$
  - $\phi^1(\vec{x_2}) = -\frac{1}{2}\sqrt{2}$
  - Punkte nun innerhalb des Interface mit neuem Abstand
- Qualität des neuen Wertes vom Zeitabstand abhängig: z.B. für $r = 1/2$ liegt $\vec{x_1}$ bei $t = 1$ wieder auf dem Interface aber: $\phi^1(\vec{x_1}) = -1 \neq 0$

Upwind-Differencing

- Untersuchen jetzt nur x-Komponente in Eulerscher Zeitdiskretisierung (5):

$$\phi^{n+1} = \phi^n + \Delta t(-u^n \phi_x^n) \tag{7}$$

- Sinnvolle Approximation für ϕ_x^n an Gitterpunkt x_i?
- Upwind-Ansatz (dt.: "windwärts"):
  - Wähle $\phi_x^+ := \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x}$, wenn $u_i < 0$
  - Wähle $\phi_x^- := \frac{\phi(x_i) - \phi(x_{i-1})}{\Delta x}$, wenn $u_i > 0$
  - Dabei ist Δx der jeweilige Abstand der Gitterpunkte

Kapitel 1: Bewegung in Normalenrichtung

- Wie kann man mit Level-Set-Methoden die Bewegung eines Objektes in Normalenrichtung darstellen? Wie lässt sich diese Bewegung mit den bisher betrachteten Bewegungen kombinieren?
- Aufstellen der Level-Set-Gleichung für die Bewegung in Normalenrichtung und kombinierte Bewegungen. Bemerkungen zur Diskretisierung dieser Bewegungen

Level-Set-Gleichung

- Bisher: Bewegungen in externem Geschwindigkeitsfeld, krümmungsabhängige Bewegung
- Jetzt: Bewegung in Normalenrichtung
- Geschwindigkeitsfeld gegeben durch: $\vec{V} = a\vec{N}$ (a konstant)
- Dies ergibt mit (4) die Level-Set-Gleichung

$$\phi_t + a|\nabla\phi| = 0 \tag{8}$$

- Anschauung: $a > 0$: Objekt dehnt sich aus, $a < 0$: Objekt zieht sich zusammen, $a = 0$: Objekt steht still

Vereinfachung durch vorzeichenbehaftete Abstandsfunktion

Sei nun ϕ eine vorzeichenbehaftete Abstandsfunktion (d.h. $\Rightarrow |\nabla\phi| = 1$)

- (8) vereinfacht sich zu $\phi_t = -a$
- Eulersche Zeitdiskretisierung: $\phi^{n+1} = \phi^n - a\Delta t$
- Anschauung: $\phi = 0$ Höhenlinie wird nach einem Zeitschritt zur $\phi = -a\Delta t$ Höhenlinie, $\phi = a\Delta t$ Höhenlinie zur $\phi = 0$ Höhenlinie.
- Gradientenversion der Zeitdiskretisierung: $\nabla\phi^{n+1} = \nabla\phi^n$ (da $a\Delta t$ (räumlich) konstant ist).
- Das heißt: Aus ϕ^n vorzeichenbehaftete Abstandsfunktion folgt ϕ^{n+1} vorzeichenbehaftete Abstandsfunktion

Upwind-Differencing bei Bewegung in Normalenrichtung

- Betrachten nun die räumliche Bewegung
- (8) $\Leftrightarrow \phi_t + a\frac{(\nabla\phi)^2}{|\nabla\phi|^2} \cdot |\nabla\phi| = 0 \Leftrightarrow$

$$\phi_t + \left(\frac{a\phi_x}{|\nabla\phi|}, \frac{a\phi_y}{|\nabla\phi|}, \frac{a\phi_z}{|\nabla\phi|}\right)\nabla\phi = 0 \tag{9}$$

- x-Komponente des Vektors: $a\phi_x|\nabla\phi|^{-1}$ (= "Geschwindigkeit in x-Richtung", vgl. "u"im externen Geschwindigkeitsfeld).
- Fallunterscheidung beim Upwind Differencing: Vorzeichen entscheidet ob ϕ_x^- oder ϕ_x^+ als Approximation für ϕ_x verwendet wird
- Problem: Verwendete Approximation beeinflusst Vorzeichen

Upwind-Differencing - Vorzeichenunterscheidung

Sei im Folgenden $a > 0$

Mögliche Fälle:

- $\phi_x^- > 0$ und $\phi_x^+ > 0$: $a\phi_x|\nabla\phi|^{-1}$ positiv, verwenden ϕ_x^- für die Approximation.
- $\phi_x^- < 0$ und $\phi_x^+ < 0$: $a\phi_x|\nabla\phi|^{-1}$ negativ, verwenden ϕ_x^+.
- ϕ_x^+ und ϕ_x^- haben unterschiedliches Vorzeichen: Benötigte Approximation im Vorhinein unklar
- Lösung: z.B. mit Godunov-Methode
  - Gleiches Vorzeichen: Konsistent mit Upwind Differencing
  - $\phi_x^+ > 0$ und $\phi_x^- < 0$: $\phi_x := 0$
  - $\phi_x^+ < 0$ und $\phi_x^- > 0$: Wähle betragsmäßig größere Approximation

Bewegung in Normalenrichtung kombiniert mit krümmungsabhängiger Bewegung...

- Level-Set-Gleichung unter Einbeziehung der Krümmung

$$\phi_t + a|\nabla\phi| = b\kappa|\nabla\phi| \tag{10}$$

- Unabhängige Diskretisierung des hyperbolischen Terms $a|\nabla\phi|$ (wie vorher beschrieben) und des parabolischen Terms $b\kappa|\nabla\phi|$ möglich (z.B. über zentrale Differenzen)
- Anschließend Entwicklung in der Zeit z.B. über Euler-Diskretisierung

...und einem externen Geschwindigkeitsfeld

- Hinzunahme eines externen Geschwindigkeitsfeldes zu (10):

$$\phi_t + \vec{V} \cdot \nabla\phi + a|\nabla\phi| = b\kappa|\nabla\phi| \tag{11}$$

- Erneut unabhängige Diskretisierung des parabolischen Terms $b\kappa|\nabla\phi|$
- Allerdings hat man nun zwei hyperbolische Terme $\vec{V} \cdot \nabla\phi$ und $a|\nabla\phi|$
- Beide Effekte bewegen den Punkt in einer Komponente in die gleiche Richtung: Upwind-Ansatz siehe oben
- Ansonsten muss der dominierende Effekt für den entsprechenden Punkt bestimmt werden

Kapitel 2: Konstruktion vorzeichenbehafteter Abstandsfunktionen

- Wie kann man zu einer gegebenen Kontur effektiv eine vorzeichenbehaftete Abstandsfunktion berechnen?
- Bedeutung der vorzeichenbehafteten Abstandsfunktion, Entwicklung eines konkreten Algorithmus zur Berechnung

Konstruktion einer vorzeichenbehafteten Abstandsfunktion

- viele Vereinfachungen durch vorzeichenbehaftete Abstandsfunktionen als Repräsentant des Interfaces
- Durch Entwicklung in der Zeit entfernt sich implizite Funktion von Ausganswerten d.h. steile/flache Gradienten und andere (numerische) "Unschönheiten" können entstehen
- Wünschenswert:
  - Zu Beginn der Bewegung Initialisierung des Interface durch vorzeichenbehaftete Abstandsfunktion ϕ
  - Während der Bewegung regelmäßige Reinitialisierung von ϕ auf vorzeichenbehaftete Abstandsfunktion
- Unkritisch, da Werte auf dem Interface nicht verändert werden ($\phi = 0$) und abseits des Interface keine relevanten Informationen enthalten

Crossing Times - Schnittzeitpunkte mit dem Interface

- Betrachten Punkt $\vec{x}$ außerhalb des Interface. Abstand zum Interface?
- Konzept: Entwickle Interface mit Geschwindigkeit a=1 in Normalenrichtung (Gemäß (8))
- Ermittle durch Interpolation den Zeitpunkt t_0 , an dem der Punkt das Interface schneidet (d.h. von positivem zu negativem Wert wechselt)
- t_0 ist gleich dem Abstand des ursprünglichen Punktes zum Interface
- Analog für $\vec{x}$ innerhalb des Interface mit a=-1

Die Fast-Marching-Methode

- Algorithmus zur Berechnung einer vorzeichenbehafteten Abstandsfunktion
- Angelehnt an obiges Konzept zur Berechnung der Schnittzeitpunkte
- Schritte:
  1. Initialisiere ein Punkteband entlang der Kontur
  2. Berechne vorläufigen Wert für alle an das Band angrenzenden Punkte
  3. Füge Punkt mit niedrigstem Wert zum Band hinzu
  4. Aktualisiere vorläufige Werte der Nachbarn, springe zu 3., falls noch nicht alle Punkte im Band

Verwaltung der Punkte

- Problem: Speicherung der vorläufigen Werte, schneller Zugriff auf Punkt mit niedrigstem Wert.
- Lösung: Speicherung der Punkte in einer Heap-Struktur
- Eigenschaften des Heap:
  - Darstellung der Punkte als Knoten eines (möglichst vollständigen) Binärbaums
  - vollständig: In jeder Ebene ist der Baum vollständig besetzt, nur in der untersten Ebene Lücken möglich
  - Wert des Elternknoten stets kleiner als Wert der Kindknoten

Arbeiten mit dem Heap

- Schritt 2:
  - Starte mit leerem Heap
  - Für jeden berechneten Punkt:
    - Hänge Punkt an der ersten freien Position an
    - Stelle Heap-Eigenschaft durch (evtl. mehrmaliges) Vertauschen mit jeweiligem Elternknoten her.
- Schritt 3:
  - Füge Wurzel dem Punkteband hinzu
  - Setze kleineren Kindknoten an freie Position
  - Verfahre so, bis der Baum wieder vollständig ist
- Schritt 4:
  - Existiert noch kein vorläufiger Wert, hänge den Punkt unten an den Heap, und stelle Heap-Eigenschaft wieder her
  - Ist schon ein vorläufiger Wert berechnet, aktualisiere diesen, und vertausche mit Elternknoten bzw. kleinerem Kindknoten bis Heap-Eigenschaft wieder hergestellt

Fast-Marching-Methode: Initialisierung des Bandes

- Betrachte alle $\phi_{i,j} = \phi(x_i, y_j)$, $\phi_{i+1,j}$ mit unterschiedlichen Vorzeichen
- Berechne Kandidaten für Abstand durch Interpolation: $\phi_{i,j}^{neu} = \phi_{i,j}^{alt} \cdot \frac{\Delta x}{|\phi_{i,j}^{alt} - \phi_{i+1,j}^{alt}|}$, analog $\phi_{i+1,j}^{neu}$
- Verfahre so in allen Koordinatenrichtungen
- Neuer Wert ist (betragsmäßiges) Minimum aller Kandidaten

Fast-Marching-Methode: Berechnung der vorläufigen Werte

- Folgendes Verfahren für Punkte außerhalb des Objekts
- Für innere Punkte analog durch "Vertauschen "von innen und außen
- Vorläufiger Wert nur von endgültig berechneten Nachbarn abhängig
- Gesuchter Wert: $\phi_{i,j}$
- Mit x_i, y_j und Nachbarpunkten $x_{i\pm1}, y_j$ und $x_i, y_{j\pm1}$ ergeben sich vier Quadranten
- Berechne Wert für jeden Quadranten wie folgt und nehme Minimum als vorläufigen Wert für $\phi_{i,j}$

Berechnung der Werte in einem Quadranten

- Ist nur ein Nachbarwert ($:= \phi_1$)endgültig berechnet, setze $\phi_{i,j} = \phi_1 + \Delta x$
- Idee für zwei berechnete Nachbarwerte ϕ_1 und ϕ_2: Interpoliere Abstand auf der Verbindungsstrecke $\overline{\phi_1\phi_2}$ und berechne $\phi_{i,j}$ als $\min\limits_{\vec{h}\in\overline{\phi_1\phi_2}} \left\{\phi(\vec{h}) + |(x_i, y_j) - \vec{h}|\right\}$
- Dies ergibt:

$$\phi_{i,j} = \min_{\theta_1+\theta_2=\Delta x, \theta_i \geq 0} \{\theta_1\phi_1 + \theta_2\phi_2 + |(\theta_1, \theta_2)|\} \tag{12}$$

- Mittels der Lagrangeschen Mutliplikatorenregel und einer kurzen Rechnung erhält man schließlich

$$(\phi_{i,j} - \phi_1)^2 + (\phi_{i,j} - \phi_2)^2 = (\Delta x)^2 \tag{13}$$

- Wähle größere Lösung der Quadratischen Gleichung als vorläufigen Wert

Kapitel 3: Die Partikel-Level-Set-Methode

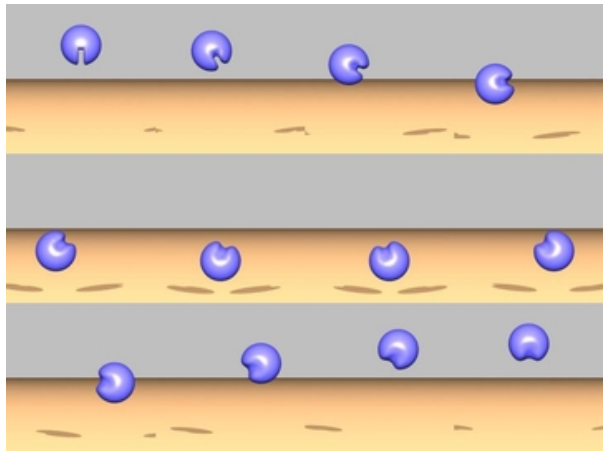
- Wie kann man Verfälschungen des Objekts im Laufe der Bewegung reparieren?
- Idee der Partikel-Level-Set-Methode zur Reparatur des Objekts

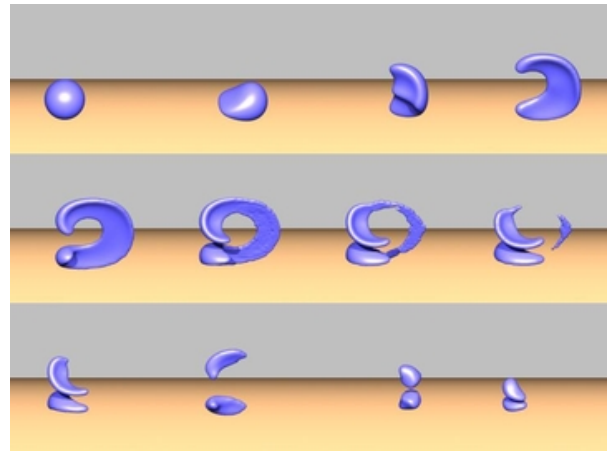
Partikelmethode zur Erhaltung der Charakteristik

- Problem: Durch zu grobe Annäherungen kann ein Objekt bei Bewegung verfälscht werden
- Lösung mit Partikelmethode:
  - Auswahl einiger Punkte in der Nähe des Interface
  - Zuweisung eines Einflußradius gleich dem Abstand
  - Kreuzt ein Punkt das Interface: Fehlverhalten der Level-Set-Methode
  - Korrektur:
    - Punkt fälschlicherweise innen: Ziehe Einflußradius vom Interface ab
    - Punkt fälschlicherweise außen: Füge Einflußradius zum Interface hinzu

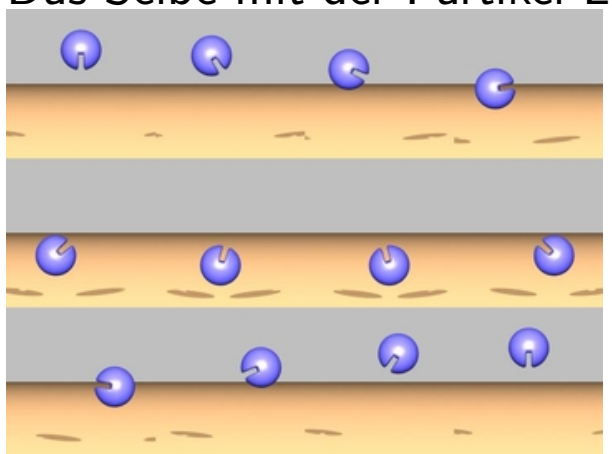
Vorteile der Partikelmethode

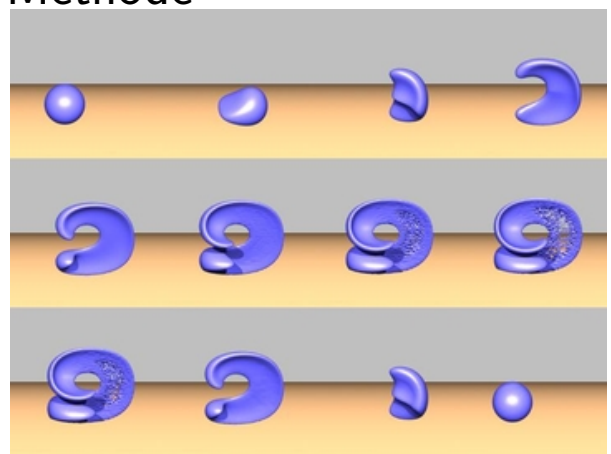
- Rotation eines eingekerbten Körpers und sogenannter "Enright-Test" mit herkömmlicher Level-Set-Methode





- Das Selbe mit der Partikel-Level-Set-Methode





Kapitel 4: Ausblick

- Welche weiteren Einsatzmöglichkeiten für Level-Set-Methoden gibt es?
- Darstellung von Objekten mit Kodimension 2 und offenen Objekten, Rückführung auf bekannte Methoden

Objekte mit Kodimension 2 als Schnitt zweier Level Set - Funktionen

Bisher: Objekte mit Kodimension 1

- Kodimension 1: Das Interface hat eine Dimension die um 1 niedriger ist, als die zugrundeliegende Umgebung
- Interface im R^1 Punkte, im R^2 Kurven und im R^3 Oberflächen
- Interface implizit durch eine Funktion in entsprechend vielen Variablen dargestellt

Jetzt: Objekte mit Kodimension 2

- Level-Set-Methoden auch anwendbar für Objekte mit Kodimension 2 (Punkte im R^2 oder Kurven im R^3)
- Darstellung als Schnittmenge der Nullstellen zweier impliziter Funktionen

Objekte mit Kodimension 2: Kurven im R^3

- Kurve als Schnitt zweier impliziter (3-dimensionaler) Funktionen ϕ_1 und ϕ_2 gegeben
- Tangentenvektor:

$$\vec{T} = \frac{\nabla\phi_1 \times \nabla\phi_2}{|\nabla\phi_1 \times \nabla\phi_2|} \tag{14}$$

- Darüber Definition von Krümmung, Normalenvektor, Binormalenvektor und Torsion
- Diskretisierung mittels obiger Methoden über ϕ_1 und ϕ_2

Offene Kurven und Oberflächen

- Bisher nur Betrachtung geschlossener Oberflächen
- Aber: auch Betrachtung offener Oberflächen mit Rand (Punkte im R^2, Kurven im R^3) innerhalb des Gebiets möglich
- Darstellung des Randes als Schnitt zweier impliziter Funktionen
- Mitbewegung der "fiktiven" Kurve (sog. "ghost curve") zur Berechnung der "realen" Kurve
- z.B: Null-Isokontur von $\phi(x, y) = y$ in dem Bereich in dem $\psi(x, y) = |x| - 1$ kleiner Null ist

Quellenangabe

- **Literatur**
  Der Vortrag basiert auf "Level Set Methods and Dynamic Implicit Surfaces"von Stanley Osher und Ronald Fedkiw, Springer 2003, vorwiegend §6-10.
- **Bilder**
  Die Bilder stammen von Ronald Fedkiws Homepage http://graphics.stanford.edu/∼fedkiw/ (letzter Zugriff am 12.11.2005). Dort findet man auch einige schöne Animationen die unter Verwendung von Level-Set-Methoden erstellt wurden.

Vielen Dank für ihre Aufmerksamkeit!