

Synthetische Baummodelle aus iterierten diskreten Graphen

Sabine Wolff

04.12.2012

Motivation

- Bäume tauchen häufig in virtuellen Welten auf
 - Computerspiele
 - Filme
- Eine möglichst einfache Baummodellierung ist gesucht und gleichzeitig soll Kontrolle der Form ermöglicht werden.
- realistisches Aussehen
 - komplizierte Astverzweigungen
 - individuelle Unregelmäßigkeiten

Gliederung

- Verwandte Ansätze
- Der Algorithmus von Xu und Mould
 - Prinzip
 - Pseudocode
 - Variation der Parameter
- Ergebnisse
 - Vergleich mit Photos
 - Vergleich mit Modellen anderer Methoden
 - Zusammenfassung und Ausblick

Methodenübersicht

- L-Systems: Lindenmeyer-Systeme
 - Modellierung von Pflanzen und Ökosystemen [2]
 - Grammatik mit paralleler, gleichzeitiger Ersetzung [1]
- Ausnutzen des zur Verfügung stehenden Raumes [3]
- Kontrolle der Form durch Skizzieren [4, 5]
- Beispiele aus einer Bibliothek [5]
- Geometrische Methoden: viele Parameter [6]
- Botanische Ansätze
 - Berücksichtigung von Umweltbedingungen [7]
- 3D Scans
- Orientierung an Photos [8]

Selbstorganisierende Baummodelle: Palubicki et al. (1)

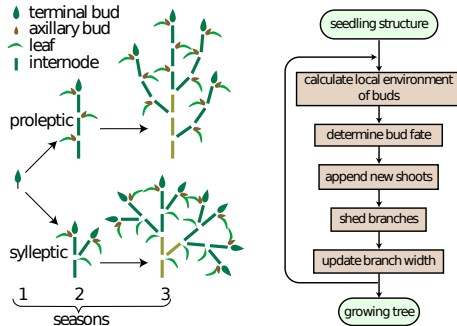
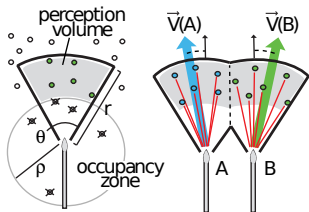


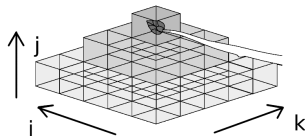
Abbildung : Simulation von Pflanzenwachstum von Palubicki et al. [7]

- Biologischer Ansatz
- Simulation der Baumentwicklung
- Interaktionen zwischen Baum und Umgebung
- Wettbewerb der Knospen und Äste nach den Ressourcen Raum und Licht
- Raumkolonisierung
- Ausbreitung von Schatten

Selbstorganisierende Baummodelle: Palubicki et al. (2)



(a) Platzkonkurrenz [7]



(b) Schattenpixel [7]



(c) Fertige Modelle [7]

Photobasierte Baummodelle: Neubert et al. (1)

1. Vorbereitung
 - 2D Attraktorgraph wird für die Bilder erstellt (Äste)
2. Erzeugen des Pixelmodells
 - Aus dem Bild werden Dichtewerte für das Pixelraster ermittelt
3. Berechnung des Richtungsfeldes
 - Richtungsfelder für jede Ebene
 - Ausgerichtet am Attraktorgraph
 - Felder der Ebenen ergeben Vektorfeld in 3D

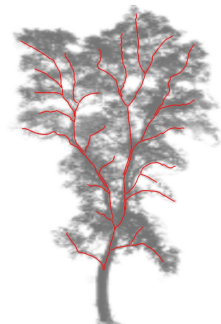


Abbildung :
Attraktorgraph [8]

Photobasierte Baummodelle: Neubert et al. (2)

4. Partikelsimulation

- Anhand der Dichtewerte werden die Ausgangspositionen von Teilchen bestimmt.
- Das 3D-Gerüst des Baumes wird mit Partikelflussimulation erzeugt.

5. Erzeugen der Geometrie

- Allometrische Proportionsberechnungen werden verwendet.
- Kleine Äste und Blätter werden hinzugefügt.



Abbildung :
Generiertes Modell
von Neubert et al. [8]

Der Algorithmus: Prinzip

- unregelmäßiges Gitter
- ausgewähltes Volumen
- Graph mit zufällig gewichteten Kanten
- ein Wurzelknoten, mehrere Endknoten
- Suche Pfad mit den geringsten Kosten
- Subgraph an jedem Endknoten
- mehrere Iterationen
- Äste werden durch dünner werdende Zylinder erzeugt

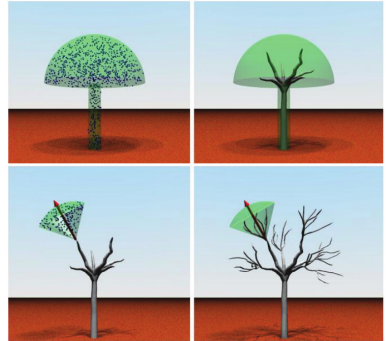


Abbildung : Illustration des Algorithmus von Xu et al. [9]

Der Algorithmus: Parameter

Globale Parameter:

- Subgraphauflösung k
(Anzahl der Knoten)
- Verknüpfungsabstand der Knoten d
- Schrumpffparameter des Subgraphs α

Knoten N , räumliche Position
 $N.\vec{x}$

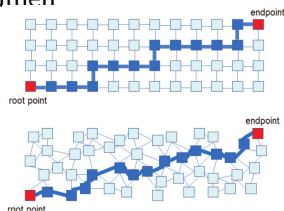
Argumente:

- Graphvolumen V
- Wurzelknoten R
- Verzweigungsfaktor b
- Anzahl der Graphknoten K
- Subgraphgröße r
- Lebensdauer des Astes L

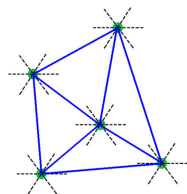
Der Algorithmus: Pseudocode(1)

1. Erzeuge den Graphen G für das aktuelle Volumen

- A. $G \leftarrow \emptyset$
- B. Finde zufällige Position \vec{p}
- C. Falls \vec{p} außerhalb von V
verwerfe \vec{p}
- D. Für alle Knoten $N \in G$:
Falls $|\vec{p} - N.\vec{x}| < d$
verwerfe \vec{p}
- E. Sonst
erzeuge Knoten m
 $m.\vec{x} = \vec{p}$
 $G \leftarrow \emptyset$
- F. Solange $|G| < K$ wiederhole B bis E
- G. Für alle Paare von Knoten $n, m \in G$:
Falls $|n.\vec{x} - m.\vec{x}| < d$
erzeuge eine Kante
- H. Für alle Kanten
setze zufällige Kosten



(a) Vergleich der Gitter [9]



(b) Yaograph [10]

Der Algorithmus: Pseudocode(2)

2. Wiederhole für b Endpunkte e :
 - A. Setze $e.\vec{x} \leftarrow$ zufällige Position in V
 - B. Für alle Knoten $n \in G$
 - Falls $|e.\vec{x} - n.\vec{x}| < d$
 - Erzeuge eine Kante von n zu e

$T \leftarrow \emptyset$
3. Erzeuge Pfade zu allen Endpunkten e
 - A. Finde die Sequenz P von Kanten mit den geringsten Kosten von e zu R
 - B. Für alle Kanten $E \in P \setminus T$:
 - $T \leftarrow \{T, E\}$

Der Algorithmus: Pseudocode(3)

4. Rekursion für alle Endpunkte:
 - A. Für alle Endpunkte e :
 - B. $\vec{v}_g = (e.\vec{x} - R.\vec{x})/|e.\vec{x} - R.\vec{x}|$
 - C. Definiere V Kegel mit Spitze bei $e.\vec{x}$, mit Radius r , Mittelpunktswinkel α und Mittelachse in Richtung \vec{v}_g
 - D. Falls $L > 0$
 $T \leftarrow \{T, \text{maketree}(V, e, b, k, a \cdot r, L - 1)\}$
5. Gib T zurück

Form des Graphen

Das Graphvolumen der ersten Iteration beeinflusst am stärksten die generelle Form des Baumes.

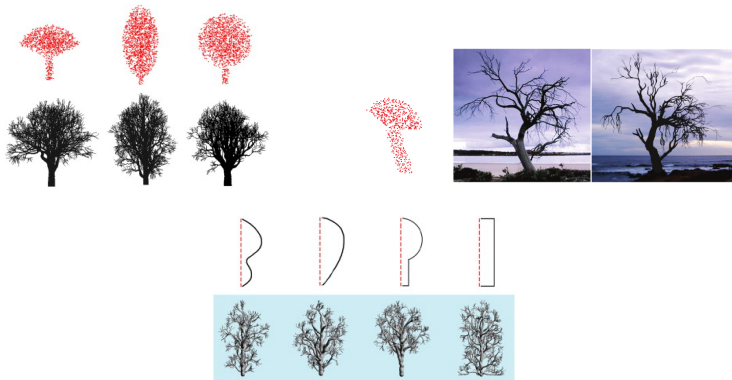


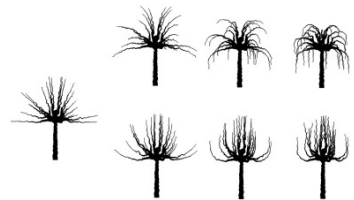
Abbildung : Einfluss des Graphvolumens [9]

Erzeugen des Subgraphs

- Subgraphvolumen wird generiert, um nötige Benutzerinteraktion zu minimieren
- Kegel ergibt gute Resultate
- Ausrichtung des Kegels \vec{v}_g :
 - Vektor von Wurzel zu Endknoten
 - Rotation
 - zufällige oder feste Richtung
 - umweltbeeinflusst
- Variation des Winkels α und des Schrumpfpparameters a .



(a) Baum rechts $a = 0.7$;
links $a = 1$ [9]



(b) Rotierende Subgraphen [9]

Lebensdauer

- Gleichmäßige Verteilung und Abnahme der Lebensdauer führt ähnlich langen Ästen.
- Variation der Lebensdauer führt zu unregelmäßiger Form.
- Zufällige Verteilung: Mit 30% Wahrscheinlichkeit endet der Ast, mit 70% Wahrscheinlichkeit verringert sich die Lebensdauer des Astes um einen zufälligen Wert.
- Umwelteinflüsse können die Lebensdauer beeinflussen.



Abbildung : Variation der Lebensdauer [9]

Vergleich mit Photos



Abbildung : Vergleich von Modellen mit Photos [9]

Vergleich mit Photos

Generierte
Modelle



Photos

Abbildung : Vergleich von Modellen mit
Photos [9]

Mögliche Baumtypen

Durch Variation des

- anfänglichen Graphvolumens,
- des Subgraphvektors,
- des Verzweigungsfaktors
- und des Schrumpfparameter

können viele verschiedene Baumtypen erzeugt werden.

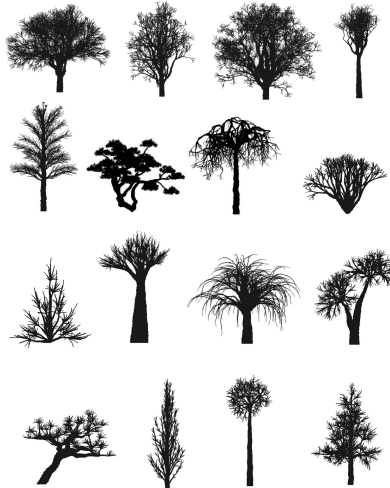


Abbildung : Mögliche Baumtypen [9]

Vergleich mit Photobasierter Methode



Abbildung : links: Neubert et al. [8]; rechts: Xu et al. [9]

- ähnliche und realistische Form
- die Methode von Xu et al. benötigt keine Photos und gibt dem Benutzer so mehr Kontrolle über das Aussehen des Modells

Vergleich mit selbstorganisierendem Baummodell



Abbildung : links: Palubicki et al.[7]; rechts: Xu et al.[9]

- Das Modell von Palubicki et al. sind gleichmäßiger, haben aber auch individuelle Unregelmäßigkeiten.
- Die Äste des Modells von Xu et al. sind stärker gekrümmt und die Lücken zwischen den Ästen sind unregelmäßig verteilt.

Zusammenfassung

- Die vorgestellte Methode basiert auf Pfadsuche in einem Graphen mit zufällig gewichteten Kanten.
- Mit den gleichen Parametern können mehrere individuelle Modelle mit ähnlichem Aussehen erzeugt werden.
- Die Form der Bäume kann kontrolliert werden durch die Veränderung des Graphvolumens, des Subgraphvolumens und die Variation der Lebensdauer eines Astes.
- Der Algorithmus benötigt keine Vorlagen und kann durch die Variation weniger Parameter gesteuert werden.
- Der Algorithmus bietet trotzdem die Möglichkeit Vorlagen zu imitieren.
- Auch Wurzeln können auf die gleiche Weise generiert werden.
- Eine Schwäche des Algorithmus ist der benötigte Speicher.

Ausblick

- Mehr Unregelmäßigkeiten sollen automatisch erzeugt werden.
- Andere Verteilungen für die Lebensdauer testen.
- Benutzerkontrolle durch Skizzieren soll untersucht werden.
- Einflüsse der Umwelt berücksichtigen um individuellere Formen zu erhalten.
- Stärkere Kontrolle der Endpunkteverteilung ermöglicht weitere Benutzerkontrolle des Aussehens.



Jack Scott Hanan.

Parametric L-systems and their application to the modelling and visualisazion of plants.

PhD thesis, University of Regina, 1992.



Przemyslaw Prusinkiewicz, Mark James, and Radomír Měch.

Synthetic topiary.

In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94*, pages 351–358. ACM, 1994.



Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz.

Modeling trees with a space colonization algorithm.

Eurographics Workshop on Natural Phenomena, 2007.



Takashi Ijiri, Shigeru Owada, and Takeo Igarashi.

The sketch lsystem: Global control of tree modeling using free-form strokes.

In *In Smart Graphics*, pages 138–146, 2006.



Makoto Okabe, Shigeru Owada, and Takeo Igarash.

Interactive design of botanical trees using freehand sketches and example-based editing.

Computer Graphics Forum, pages 487–496, 2005.



Jason Weber and Joseph Penn.

Creation and rendering of realistic trees.

In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 119–128. ACM, 1995.



Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz.

Self-organizing tree models for image synthesis.

ACM Trans. Graph., pages 58:1–58:10, 2009.



Boris Neubert, Thomas Franken, and Oliver Deussen.

Approximate image-based tree-modeling using particle flows.

ACM Trans. Graph., 2007.



Ling Xu and David Mould.

Synthetic tree models from iterated discrete graphs.

In *Proceedings of the 2012 Graphics Interface Conference*, GI '12, pages 149–156. Canadian Information Processing Society, 2012.



Ling Xu and David Mould.

A procedural method for irregular tree models.

Computers and Graphics, 2012.



Ling Xu and David Mould.

Constructive Path Planning for Natural Phenomena Modeling, pages 59–69.

Springer, 2008.