

GEORG-AUGUST-UNIVERSITÄT-GÖTTINGEN

10. Mai 2013

TreeSketch:  
Interactive Procedural Modeling of Trees on a Tablet

von

Steffen Zenker

Seminar Computergrafik WS 12/13  
Prof. Dr. W. Kurth

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Methoden zur Baummodellierung . . . . .	1
1.2.1	Rekonstruktion . . . . .	2
1.2.2	Prozedurale Modelle . . . . .	2
1.3	TreeSketch . . . . .	2
<b>2</b>	<b>Implementierung</b>	<b>4</b>
2.1	Idee . . . . .	4
2.2	Generierung von Baummodellen . . . . .	5
2.3	Manipulation von Baummodellen . . . . .	9
2.4	Benutzeroberfläche . . . . .	10
<b>3</b>	<b>Zusammenfassung</b>	<b>12</b>

# 1 Einleitung

## 1.1 Motivation

In vielen 3D-Anwendungen ist es erforderlich komplexe Bäume darzustellen, die völlig natürlich und folglich für die betrachtende Person „schön“ aussehen. Dabei treffen zwei für sich schon schwierige Probleme aufeinander: die kreative Gestaltung von Bäumen, sowie die Einbettung dieser in die Gesamtszene und das Erstellen realistischer Bäume. So existiert beispielsweise für eine Filmszene eine grobe Skizze, die künstlerische Anforderungen an einen oder mehrere Bäume beinhaltet. Bei der Erstellung der Baummodelle müssen einerseits diese Anforderungen eingehalten als auch ein natürliches Baumwachstum simuliert werden. Abbildung 1 zeigt beispielhaft eine Filmszene, die von Bäumen geprägt ist. Insbesondere ist hier zu erkennen, dass es schwierig ist, Bäume in einem Modellkatalog zu finden, die die Anforderungen erfüllen und es wichtig ist bei mehreren Bäumen diese unterschiedlich zu gestalten, da die Szene ansonsten sehr künstlich und wenig ansprechend aussehen würde.



Abbildung 1: Aus Avatar – Aufbruch nach Pandora, James Cameron, 2009

Um den immensen Aufwand zu reduzieren und die beiden oben genannten Probleme zu lösen, wurde das Tool TreeSketch [1] entwickelt. Es verbindet prozedurale Baummodelle mit einem multi-touch Tabletinterface, also gleichsam die biologische Plausibilität der entstehenden Bäume mit einer einfachen Handhabung, die künstlerische Anforderungen erfüllt. Der Vision von R. Cook

*„Our [modeling] tools need to feel more like brushes.“*

Rob Cook, 2009 Steven A. Coons Award speech

folgend, kann der Wachstumsprozess durch Pinselwerkzeuge gelenkt werden. So ist es möglich komplexe Bäume einfach per Hand zu skizzieren ohne dabei komplizierte Algorithmen Schritt für Schritt anpassen zu müssen.

## 1.2 Methoden zur Baummodellierung

Es existieren verschiedene Ansätze zur Baummodellierung. Bäume können entweder von biologischen Vorbildern rekonstruiert, oder das Baummodell komplett künstlich generiert werden. Dabei spielt die Computergrafik in beiden Fällen eine komplett unterschiedliche Rolle. Im ersten Fall besteht die Hauptaufgabe in der Datenverarbeitung und der Darstellung der Daten, wohingegen bei der letzteren Vorgehensweise eine komplette Wachstumssimulation durchgeführt wird.

### 1.2.1 Rekonstruktion

Rekonstruktionsmethoden basieren auf der Vermessung und dem Sammeln von Daten über reale Bäume mit Hilfe von z.B. Laserscans, oder Fotos. Ohne große Benutzerinteraktion können so sehr genaue Baummodelle erzeugt werden. Allerdings muss zunächst ein Baum gefunden werden, der die künstlerischen Voraussetzungen erfüllt, was bei vielen Bäumen einen großen Aufwand nach sich zieht. Ein einziges Baummodell würde zu eintönig aussehen, falls man dies in einer Ansammlung von Bäumen einfach wiederholen würde. Desweiteren muss der gewählte Baum aus technischen Gründen ausreichend isoliert stehen, da andere Objekte auch in das Modell mit einfließen würden. Problematisch wird dies bei Bäumen, die z.B. nur in Wäldern vorkommen.

### 1.2.2 Prozedurale Modelle

Um diese Nachteile auszugleichen können Bäume komplett computergestützt generiert werden. Dazu existieren zwei Extreme: Baumgeneration aufgrund rekursiver oder hierarchischer Verzweigungsmuster und Baumsimulation, bei der sich die Verzweigungsstruktur durch die Konkurrenz um Raum und Licht der Blätter, bzw. Äste entwickelt. Bei allen Ansätzen ist jedoch das gemeinsame Ziel Bäume mit realistischem Aussehen zu erzeugen, wobei die Möglichkeit besteht auf die genaue Form einzuwirken.

Rekursive und hierarchische Modelle wie [2, 3, 4, 5, 6] funktionieren in einer bottom-up Arbeitsweise. Sie wiederholen lokal definierte Verzweigungsgeometrien in Form von Regeln über die aufeinander folgenden Verzweigungsordnungen. Somit kann eine große Anzahl von verschiedenen Baumformen erzeugt werden. Die Kontrolle über die genaue Form benötigt allerdings ein genaues Verständnis des Algorithmus und dessen Implementierung seitens des Modellierers um die Parameter anpassen zu können. In den meisten Fällen entstehen die globalen Eigenschaften des Baumes wie die Silhouette, oder die Dichte der Zweige durch die explizite Ausführung des Algorithmus und können nicht direkt eingestellt werden. Eine Vereinfachung die Baumform besser zu spezifizieren wurde durch die Möglichkeit parametrische Kurven [7] anzugeben hinzugefügt.

Selbstorganisierte Baumgenerierung, die ihre Wurzeln bezüglich Computergrafik in [8, 9] hat, beseitigt einige Probleme der rekursiven und hierarchischen Modelle. So ist deren größte Limitierung, dass die harmonische Verteilung der Astdichten innerhalb der Krone nicht automatisch entsteht, sondern durch den Anwender sichergestellt werden muss. Durch die Simulation von Konkurrenz um Licht und Raum entstehen natürliche und biologisch plausible Bäume. Desweiteren wurden in [10] interne Signale eingeführt, die die eingesammelte Lichtenergie der Blätter unverteilen und so entscheiden, ob ein Ast profitabel für die Pflanze ist, oder nicht. Die konkrete Baumform kann durch die Manipulation der Umgebung bestimmt werden.

## 1.3 TreeSketch

Die prozeduralen Komponenten von TreeSketch basieren auf Algorithmen für die Generierung von selbstorganisierten Bäumen [9]. Sie ermöglichen eine interaktive Benutzung und können daher von der eingesetzten Tablet-Plattform (Apple iPad 2) optimal profitieren. Bäume können so, wie in Abbildung 2 gezeigt, mit der multitouch Oberfläche per Finger und den bereitgestellten Pinselwerkzeugen skizziert werden. Um dieses Grundgerüst wird dann der Baum generiert und kann anschließend editiert werden. Dafür stehen eine große Auswahl von Fingergesten zur Verfügung, die z.B. den Baum einfach vergrößern, oder rotieren. Die Benutzeroberfläche bleibt dabei intuitiv und beschränkt die ganze Aufmerksamkeit auf die Modellierung des Baumes und versteckt also die internen Algorithmen.



Abbildung 2: Bäume können durch einfache Züge des Pinselwerkzeugs modelliert werden (aus [1]).

Neben dem Zeichenbereich gibt es noch interaktive Elemente um eine Vielzahl an Parametern (siehe Abbildung 3) zu definieren, mit denen experimentiert werden kann um die Form des Baumes zu beeinflussen. Dazu zählen äußerliche Faktoren wie z.B. die Auswahl von Blättern, aber auch algorithmische Parameter die z.B. über die Position und Wachstumsrichtung von neuen Ästen entscheiden.

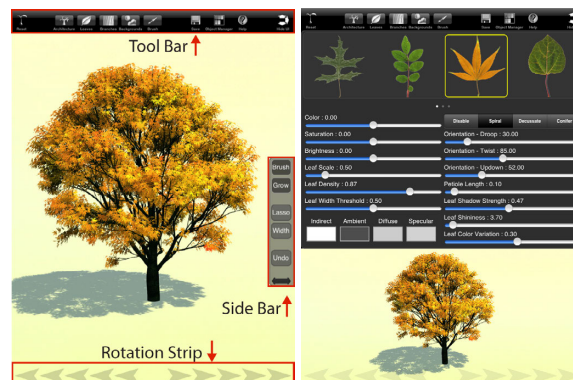


Abbildung 3: Treesketch Benutzeroberfläche, Zeichenbereich links und Auswahl der Darstellungsparameter rechts (aus [1])

## 2 Implementierung

### 2.1 Idee

In TreeSketch sollen höchste biologische Plausibilität mit der interaktiven Gestaltung von Bäumen kombiniert werden. Dazu werden zum einen Mechanismen der Konkurrenz um Licht und Raum (Natürlichkeit) und zum anderen interne Signale (Kontrolle der Form) benutzt. Durch die Nutzung von selbstorganisierten Modellen wird die Breite an möglichen Formen zwar etwas eingeschränkt, aber diese ermöglicht die Skizzierung von Grundformen durch ein Pinselwerkzeug.

Die grundsätzliche Idee ist in Abbildung 4 dargestellt. Beginnend mit einem Samen, wird das Wachstum durch das Pinselwerkzeug gesteuert (brauner Kreis). Wachstum ist nur zu einem der innerhalb des Pinselbereichs zufällig verteilten Wachstumspunkte (gelb) möglich. Ob ein neuer Zweig zu einem freien Wachstumspunkt entsteht, wird durch die vorhandene Energie gesteuert. Lichtenergie wird unter Beachtung von Verschattungseffekten von allen Knospen aufgenommen und an die Wurzel geschickt (blaue Pfeile). Von dort wird die eingesammelte Energie wieder abhängig von der Relevanz für die Pflanze an die Knospen verteilt (rote Pfeile). Reicht die Energie aus, so wird eine neue Verzweigung erzeugt und das Verfahren fortgesetzt.

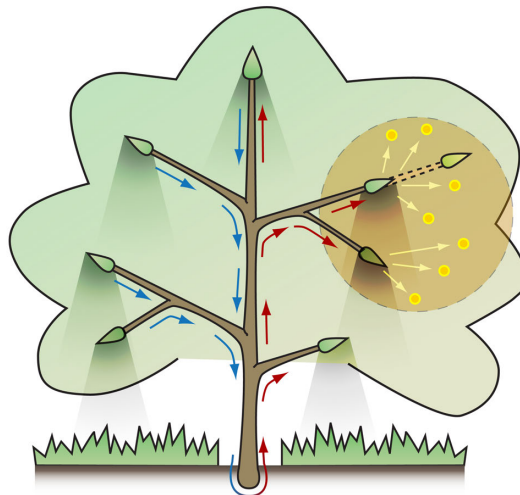


Abbildung 4: Licht wird von Knospen abhängig von Schatten aufgenommen und nach unten geschickt (blaue Pfeile). Energie wird zurück an die Knospen verteilt (rote Pfeile). Knospen mit ausreichend Energie und freiem Wachstumspunkt (gelb) sprießen. Wachstumspunkte werden mit dem Pinselwerkzeug (braun) verteilt (aus [1]).

Das Pinselwerkzeug verteilt in seinem Einflussbereich die Wachstumspunkte innerhalb einer Sphäre anhand einer Gleichverteilung. Dies kann durch Fingerbewegungen des Benutzers gesteuert werden. Bei Bedarf kann der Radius auch während eines Pinselzuges modifiziert werden (siehe Abbildung 5 rechts).



Abbildung 5: links: Einsatz des Pinselwerkzeugs unter Rotation und sichtbaren Wachstumspunkten, rechts: Modifikation des Pinselradius während eines Zuges (aus [1])

Zusätzlich zum Pinselwerkzeug gibt es noch das Lassowerzeug. Dies kann dazu genutzt werden um vielmehr die Silhouette des Baumes zu bestimmen, als dessen Hauptachsen zu skizzieren. Um die benötigte 3D Hülle für das Baumwachstum zu erhalten wird die 2D Skizze zunächst mit Bildverarbeitung behandelt. Zuerst wird die Skizze skelettiert um die Hauptachsen zu erhalten. Dies wird durch bildbasierte Erosion umgesetzt. Um diese Achsen wird dann im nächsten Schritt eine 3D Hülle aufgeblasen. Zum Schluss werden die Wachstumspunkte in einem einschließenden Quader gleichmäßig verteilt und wieder gelöscht, falls sie nicht in dieser Hülle liegen.



Abbildung 6: Abstrakte Baumformen erzeugt mit dem Lassowerzeug (aus [1]).

## 2.2 Generierung von Baummodellen

Wie in Abbildung 7 (links) dargestellt, sind Knospen, aus denen eine Verzweigung resultieren kann, mit einem Abstand  $l$  an den Ästen verteilt. Sie sind in einem Winkel  $\phi$  (Phyllotaxis Winkel) verdreht, was zu einer spezifischen Blattanordnung führt, und die Verzweigungen sprießen in einem Winkel  $\psi$  vom Hauptast. Sprießt eine Knospe, ist die Richtung grundsätzlich durch den zugeordneten Wachstumspunkt bestimmt. Allerdings wird diese Richtung noch von der Erdanziehungskraft beeinflusst und ist ggf. durch die Begrenzung des Raumes beschränkt.

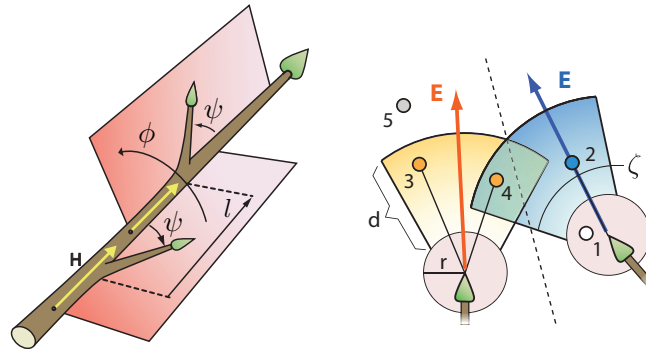


Abbildung 7: links: Anordnung der Knospen an den Ästen, rechts: Konkurrenz der Knospen um Wachstumspunkte (aus [1])

Die Knospen sind von einer Kugel mit Radius  $r$  umgeben (siehe Abbildung 7 (rechts)). Weiterhin hat jede Knospe einen Interaktionsbereich definiert durch einen Ausschnitt einer Kugel mit Durchmesser  $d + r$  begrenzt auf den Winkel  $\zeta$  in Knospenrichtung. Ein Wachstumspunkt wird entfernt, falls

- er in Kugeln einer, oder mehrerer Knospen liegt (Bsp.: Punkt 1)
- er durch Pinsel erzeugt wurde und das Maximalalter überschreitet (direkterer Einfluss)
- er durch Lasso oder autonom erzeugt und das Wachstum abgeschlossen wurde (nicht durch Knospen erreichbar)

Die Knospen konkurrieren nun um die verbleibenden Wachstumspunkte anhand der folgenden Regeln:

- Punkte, die nur in einem einzigen Interaktionsbereich liegen, werden der zugehörigen Knospe zugeordnet (Bsp.: Punkt 3 oder 2)
- Punkte, die in zwei Bereichen liegen, werden der Knospe zugeordnet, die am nächsten ist
- Alle anderen Punkte bleiben in diesem Schritt unzugeordnet

Die endgültige Wachstumsrichtung ist dann der Durchschnitt der Richtungen der einer Knospe zugeordneten Punkte.

Nach der Bestimmung der Wachstumsrichtung muss nun noch die Stärke des Wachstums ermittelt werden. Dies wird mit einem erweiterten Borchert-Honda Algorithmus [11] umgesetzt. Dafür hängt die Stärke des Wachstums aktiver Knospen

$$n = \left\lfloor \frac{v}{v_{max}} n_{max} \right\rfloor$$

von der zur Verfügung stehenden Energie  $v$  ab. Dabei ist  $v_{max}$  die höchste Energie an den Knospen und  $n_{max}$  ein Parameter, der die maximale Länge bestimmt. Diese Energie wird durch das Sammeln der Lichtenergie

$$Q = bL^k, \text{ mit } (k > 0)$$

die durch die Lichtmenge  $L$  erzeugt wurde, ermittelt.



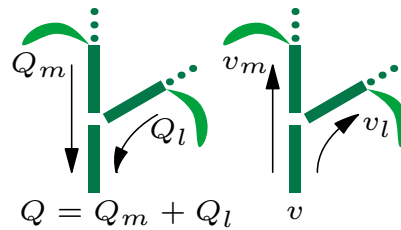


Abbildung 8: Energieumverteilung im Borchert-Honda Algorithmus (aus [11]).

Wie in Abbildung 8 dargestellt, wird die Energie zunächst eingesammelt und dann anhand von

$$v_m = v \frac{\lambda Q_m}{\lambda Q_m + (1 - \lambda) Q_l}, v_l = v \frac{(1 - \lambda) Q_l}{\lambda Q_m + (1 - \lambda) Q_l}$$

an die Hauptzweige ( $v_m$ ) und lateralen Nebenzweige ( $v_l$ ) verteilt. Dabei steuert der Parameter  $\lambda$  die Stärke der Bevorzugung von Haupt-, bzw. den Nebenzweigen.

Der Koeffizient  $b$  in der Lichtsammelgleichung  $Q = bL^k$  beinhaltet die Ausprägung von horizontalen, bzw. vertikal angeordneten Knospen. Es werden drei verschiedene Arten von Knospen unterschieden:

- Epitonie: Knospen auf der oberen Seite von Zweigen (Abbildung 9 rechts)
- Amphitonie: Knospen in horizontaler Position
- Hypotonie: Knospen auf der Unterseite (Abbildung 9 links)



Abbildung 9: Darstellung des Einflusses von verschiedenen Einstellungen bzgl. des Gravimorphismus (aus [1]).

Um diesen Effekt zu implementieren wird, wie in Abbildung 10 gezeigt, auf die rechtwinklige Ebene zum Zweig eine Ellipse gelegt. Die Neigung  $b$  einer Knospe zu Wachsen hängt von der

Distanz  $d$  von Ast zu Ellipse im Winkel  $\chi$  ab. Mit  $\tau$  als Winkel zwischen  $H$  und der Vertikalen  $Y$  erhält man

$$b = \cos^2 \tau + d \sin^2 \tau.$$

Dies maximiert den Effekt für horizontale Äste. Soll der Effekt nun manipuliert werden, so wird einfach die Ellipse passend verschoben, sodass nun andere Äste bevorzugt werden.

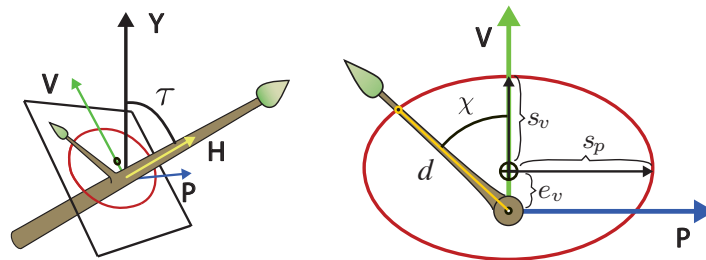


Abbildung 10: Umsetzung des Gravimorphismus durch eine Ellipse auf der Ebene orthogonal zur Richtung  $H$  (aus [1]).

Eine weitere Eigenschaft beim Wachstum von Bäumen ist der Gravitropismus. Dieser spiegelt gleichsam wieder ob Zweige mehr oder weniger ihre Wachstumsrichtung beibehalten, oder sich der Schwerkraft anpassen. Dazu werden drei Arten unterschieden:

- negativer Gravitropismus: Tendenz nach oben zu wachsen
- Plagiotropismus: Tendenz horizontal zu wachsen
- positiver Gravitropismus: Tendenz nach unten zu wachsen (Abbildung 11 in Kombination mit Epitonie)



Abbildung 11: Baum mit Epitonie und positiven Gravitropismus (aus [1]).

Wie auf Abbildung 11 zu sehen, können Gravimorphismus und Gravitropismus in allen möglichen Kombinationen vorkommen und führen somit zu unterschiedlichen Resultaten, die für reale Baumarten spezifisch sind.

Der Gravitropismus wird durch einen Tropismus-Vektor  $T$  durch Drehung von  $Y$  (vertikale Achse) um  $\theta$  bestimmt. Die angepasste Wachstumsrichtung ist die gewichtete Summe aus  $T, H$  (ursprüngliche Wachstumsrichtung) und  $E$  (bevorzugte Wachstumsrichtung durch Umgebung).

$$H' = \alpha H + \beta T + \gamma E$$

Dabei gilt  $\alpha + \beta + \gamma = 1$ . Dies ist zur Veranschaulichung in Abbildung 12 dargestellt.

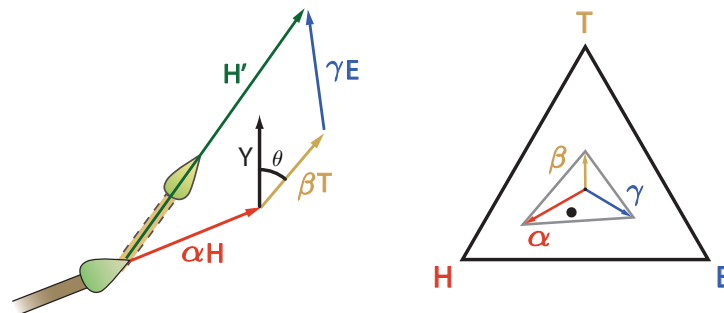


Abbildung 12: Bestimmung der Wachstumsrichtung mit Beachtung des Tropismus-Vektors ([1]).

### 2.3 Manipulation von Baummodellen

Wurde ein Baummodell nach den Vorgaben des Benutzers erzeugt, so kann es nachträglich manipuliert werden. Als Beispiel ist die Verschiebung eines Astes in Abbildung 13 (rechts) angegeben. Der übliche Ansatz führt über inverse Kinematik, die es ermöglicht aus den Zuständen der Gelenke die Position des Endpunktes zu berechnen. Allerdings ist dieses Vorgehen hier problematisch, da lokale Asteigenschaften nicht erhalten werden und diese Methodik nicht reversibel ist. Daher wird eine auf Bäume angepasste PriMo Methode [12] angewandt. Hier wird, wie in Abbildung 13 (links) dargestellt, die Maschengemetrie als Menge von Prismen, verbunden durch elastische Gelenke, interpretiert.



Abbildung 13: links: Prismen verbunden durch elastische Gelenke, rechts: Verschiebung eines Astes mit einer 3-Finger-Geste (aus [1])

Es werden Positionsbedingungen an verschiedene Prismen gestellt und anschließend die Verformung durch Minimierung der elastischen Energie der Gelenke bestimmt. Die Manipulation kann über die touch Oberfläche mit drei Fingern durchgeführt werden.

Auch eine Manipulation von Astdicken an verschiedenen Stellen ist möglich. Die Astdicke wird, wie in Abbildung 14 (links) dargestellt, rekursiv für alle Äste bestimmt. Der Stamm eines Baumes mit  $k$  terminierenden Abzweigungen, die die Durchmesser  $d_0, \dots, d_M$  besitzen, benötigt einen Durchmesser  $d$  der, nach der da Vinci Formel,

$$d^n = \sum_{i=0}^M d_i^n$$

erfüllt. Da Vinci besagt, dass die Summe der Schnittflächen der Vateräste der Summe der Querschnitte der Kinderäste entspricht. Somit sollte der Exponent  $n$  eigentlich 2 sein. Für realistischere Ergebnisse wird allerdings die Modifikation

$$n = \frac{\log M}{\log d_b - \log d_e}$$

benutzt, wobei  $d_b$  der Durchmesser des Stammes und  $d_e$  der Durchmesser der terminalen Äste ist. Damit kann durch eine 2-Finger-Geste (siehe Abbildung 14 (rechts)) an bestimmten Stellen im Baum eine Anforderung an Astdicken gestellt werden, ohne die Natürlichkeit des Baumes zu gefährden.



Abbildung 14: links: rekursive Berechnung der Astdicke mit Anforderungen  $d_i$  an verschiedene Äste, rechts: Anforderungen an Astdicken an verschiedenen Stellen im Baum (aus [1])

## 2.4 Benutzeroberfläche

Neben den bereits beschriebenen Komponenten der Benutzeroberfläche, gibt es noch einen Bereich um die algorithmischen Parameter zu definieren (siehe Abbildung 15). Dabei müssen nicht abstrakte Zahlen eingegeben werden, sondern man bekommt über alle Einstellungen ein visuelles Feedback.

1. Gravotropismus: oben links kann über den Pfeil der Winkel  $\theta$  eingestellt werden.
2. Gravimorphismus: oben rechts kann die Ellipse verschoben werden, sodass die Parameter  $s_v$ ,  $s_p$  und  $e_v$  angepasst werden.

3. Verzweigungswinkel: unten links können  $\psi$  und  $\delta$  angepasst werden.
4. Verzweigungsrichtung: unten mitte kann man die Parameter  $\alpha$ ,  $\beta$  und  $\gamma$  aus Abbildung 12 definieren
5. Konkurrenz um Licht und Raum: unten rechts können die Parameter  $r$ ,  $d$  und  $\xi$  eingestellt werden

Alle Änderungen werden direkt auf das Modell angewendet und sind sofort sichtbar.

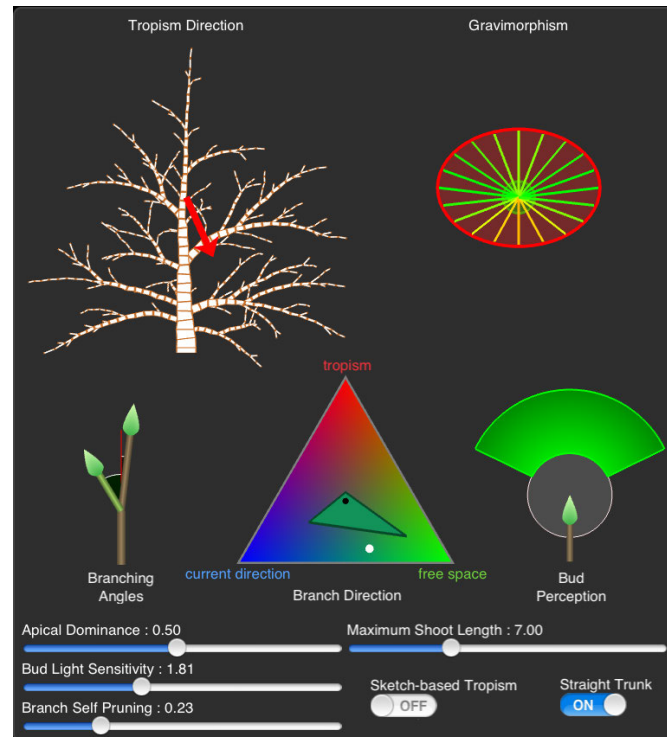


Abbildung 15: Konfigurationsbereich für die algorithmischen Parameter, wie z.B. Gravitropismus oben links, oder Gravimorphismus oben rechts (aus [1]).



### 3 Zusammenfassung

TreeSketch ist ein Tool, das sich zum Ziel setzt die künstlerische und biologisch plausible Generierung von Bäumen zu vereinen. Mit der multitouch Oberfläche bietet es einfache Bedienung durch Finger-Gesten. Diese interaktive Kontrolle wurde mit generativen Algorithmen kombiniert, die eine einfache Handhabung ermöglichen, aber trotzdem alle wichtigen Parameter für einen individuellen Baum bieten. Jede erdenkliche Baumform kann kreiert werden ohne großes Hintergrundwissen über algorithmische, bzw. biologische Details zu kennen. Weiterhin können diese komplexen Modelle auch auf relativ schwacher Hardware erstellt werden. Die erzeugten Modelle können dann in einem finalen Schritt in eine virtuelle Szene (Computerspiel, Film, o.Ä.) eingebettet werden. Abbildung 16 zeigt zwei Bäume, die im Maya gerendert wurden.

Trotzdem gibt es noch einige Ansatzpunkte zur Weiterentwicklung. So haben Biologen festgestellt, dass nicht nur der Winkel der Zweige, sondern auch deren Krümmung eine Rolle beim Gravimorphismus spielt. Desweiteren sollen Modelle von Früchten hinzugefügt werden. Auch diese haben Einflüsse auf die betroffenen Zweige. Schlussendlich sollen noch die Blätter optimiert werden. Die Blattgeometrie soll verfeinert werden um eine noch bessere Darstellung zu erhalten.



Abbildung 16: Zwei in Maya gerenderte und in TreeSketch erzeugte Bäume (aus [1]).

## Literatur

- [1] LONGAY S., RUNIONS A., BOUDON F., PRUSINKIEWICZ P.: TreeSketch: Interactive Procedural Modeling of Trees on a Tablet, EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling, 2012
- [2] HONDA H.: Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology* 31 (1971), 331-338
- [3] AONO M., KUNII T. L.: Botanical tree image generation. *IEEE Computer Graphics and Applications* 4, 5 (1984), 10-34
- [4] BLOOMENTHAL J.: Modeling the Mighty Maple. *Computer Graphics* 19, 3 (1985), 305-311
- [5] OPPENHEIMER P.: Real time design and animation of fractal plants and trees. *Computer Graphics* 20, 4 (1986), 55-64 2
- [6] PRUSINKIEWICZ P., LINDENMAYER A.: *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990
- [7] BOUDON F., PRUSINKIEWICZ P., FEDERL P., GODIN C., KARWOWSKI R.: Interactive design of bonsai tree models. *Computer Graphics Forum* 22, 3 (2003), 591-599
- [8] GREENE N.: Voxel space automata: modeling with stochastic growth processes in voxel space. *Computer Graphics* 23, 4 (1989), 175-184
- [9] RUNIONS A., LANE B., PRUSINKIEWICZ P.: Modeling trees with a space colonization algorithm. In *Eurographics Workshop on Natural Phenomena (2007)*, pp. 63-70.
- [10] MECH R., PRUSINKIEWICZ P.: Visual models of plants interacting with their environment. In *Proceedings of SIGGRAPH (1996)*, pp. 397-410
- [11] PALUBICKI W., HOREL K., LONGAY S., RUNIONS A., LANE B., MECH R., PRUSINKIEWICZ P.: Self-organizing tree models for image synthesis. *ACM Transactions on Graphics* 28, 3 (2009), 58:1-58:10
- [12] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: Primo: Coupled prisms for intuitive surface modeling. In *Proceedings of the fourth Eurographics symposium on Geometry processing (2006)*, pp. 11-20