

Fraunhofer



Institut  
Graphische  
Datenverarbeitung

Entwickelt von Timo Lehr



# Hairdesigner

Haarplugin für Cinema 4D

## Anleitung

## Inhaltsverzeichnis

|  |    |
|--|----|
| Überblick .....  | 3  |
| Die Oberfläche .....   | 4  |
| Parameter-Schieberegler .....  | 5  |
| Selektierung .....   | 6  |
| Beschreibung der Steuerelemente.....   | 7  |
| 1 Plugin an Haargeometrie anheften „Attach To Selected Head-Obj“ .....                   | 7  |
| 2 Aktuelle Kopfgeometrie .....   | 8  |
| 3 Weiteres Haarsets hinzufügen „Add New Hairset“ .....                                   | 8  |
| 4 Zwischen mehreren Haarsets umschalten .....  | 9  |
| 5 Gleichgroße Segmente erzeugen „Auto Subdivide“ .....                                   | 9  |
| 6 Aktuelle Auswahl abschneiden „Cut selected Hairs“ .....                                | 9  |
| 7 Haardichte-Steuerung „Density-Subdivides“ .....  | 10 |
| 8 Zufällige Haardichte „Density Random“ .....  | 10 |
| 9 Anzeige von Strähnen und Polygonen .....   | 11 |
| 10 Haarlänge-Steuerung „Length Scale“ .....  | 11 |
| 11 Zufällige Haarlänge „Length Random“ .....   | 11 |
| 12 Start-Segmentlänge / 13 Segment-Verlängerungsfaktor .....                             | 11 |
| 14 Mittelscheitel-Stärke „Cortex-Scale“ .....  | 12 |
| 15 Haarüberlappung an Scheitel „Cortex Overlap“ .....                                    | 13 |
| 16 Basisfarbe der Haare „Global Color“ .....   | 14 |
| 17 Aufhellung der Haare „Color-Increment“ .....  | 14 |
| 18 Zufällige Haarfarbe „Random Brightness“ .....   | 14 |
| 19 Haardicke-Steuerung „Thickness Scale“ .....   | 14 |
| 20 Zufällige Dicke „Thickness Random“ .....  | 15 |
| 21 Neue Kollisionsobjekte erstellen „Add Collision Sphere“ / „Add Collision Plane“ ..... | 15 |
| 22 Kugelanzahl-Kontrolle .....   | 16 |
| 23 Basiskollisionsgröße der Haarsträhnen „Hair Bounding Size“ .....                      | 16 |
| 24 Erhöhung der Haarkollisionsgröße „Hair Bounding Increment“ .....                      | 16 |
| 25 Zufällige Haarkollisionsgröße „Hair-Bounding-Random“ .....                            | 16 |
| 26 Haarsteifheit „Bending Scale“ .....   | 16 |
| 27 Zufällige Steifheit „Bending Random“ .....  | 17 |
| 28 Schwerkraftstärke „Gravity“ .....   | 17 |
| 29 Manuelle Simulation „Test Simulation with 10 Steps“ .....                             | 17 |
| 30 Automatische Simulation „Auto Simulate“ .....   | 17 |
| 31 Exportmodus wählen „Exportmode“ .....   | 17 |
| 32 Ausgabedatei wählen .....   | 18 |
| 33 Jetzt exportieren „Export to VRML“ .....  | 18 |
| Undo .....   | 19 |
| Der interne Aufbau des Plugins .....   | 20 |
| Hair-Set-Designer.....   | 20 |
| Control-Point-Manager .....  | 22 |
| Export-Templates.....  | 24 |

**Anleitung zum Stand 06.06.06**

## Überblick

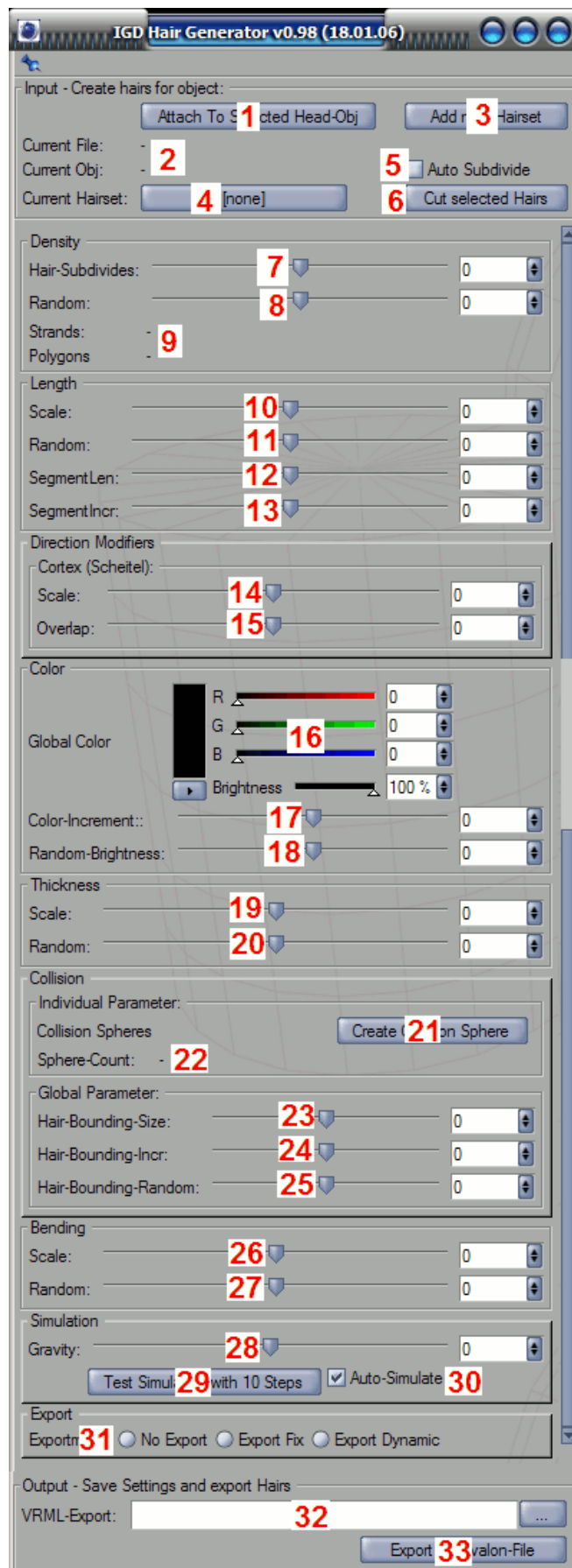
Die Aufgabe des Haarplugins ist es, mittels Cinema4D auf einer bestehender Kopf-Geometrie kinderleicht Haare zu erstellen und individuelle Frisuren zu gestalten. Die erzeugten Haare sind speziell für den Echtzeiteinsatz im Avalon-Rendersystem mit der zugehörigen Haarsimulation zu betreiben.

In dieser Dokumentation werden Stück für Stück alle Steuerelemente beschrieben, so daß der „Einsteiger“ sowohl einen guten Überblick über die Funktionalität bekommt, als auch der „Experte“ eine Gelegenheit hat, nähere Informationen über einen bestimmten Vorgang nachzuschlagen.

Der Aufbau ist so vorgenommen, daß zur leichten Wiederfindung jedes Steuerelemente immer unter der angegebenen Nummer zu finden ist, die gleichzeitig der Unterkapitelnummer entspricht.

Desweiteren finden Sie weiter hinten im Dokument Informationen über die interne Funktionsweise, sowie die Dokumentation über den Aufbau der Template-Dateien für den Export nach Avalon.

## Die Oberfläche



### Kurzübersicht der Steuerelemente

- 1 Plugin an Haargeometrie anheften
- 2 Kontrolle der aktuellen Kopfgeometrie
- 3 Weiteres Haarsets hinzufügen
- 4 Zwischen mehreren Haarsets umschalten
- 5 Gleichgroße Segmente erzeugen
- 6 Aktuelle Auswahl abschneiden

- 7 Haardichte-Steuerung
- 8 Zufällige Haardichte
- 9 Anzeige der Strähnen und Polygone

- 10 Haarlänge-Steuerung
- 11 Zufällige Haarlänge
- 12 Initial Segmentlänge
- 13 Segment-Verlängerungsfaktor

- 14 Mittelscheitel-Stärke
- 15 Haarüberlappung an Scheitel

- 16 Basisfarbe der Haare
- 17 Aufhellung der Haare
- 18 Zufällige Haarfarbe

- 19 Haardicke-Steuerung
- 20 Zufällige Dicke

- 21 Neue Kollisionskugel erstellen
- 22 Kugelanzahl-Kontrolle

- 23 Basisdicke der Haarsträhnen
- 24 Erhöhung der Haardicke
- 25 Zufällige Haardicke

- 26 Haarsteifheit
- 27 Zufällige Steifheit

- 28 Schwerkraftstärke
- 29 Manuelle Simulation
- 30 Automatische Simulation

- 31 Exportmodus wählen

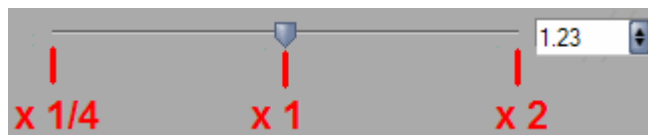
- 32 Ausgabedatei wählen
- 33 Jetzt exportieren

Die Oberfläche teilt sich zunächst in 3 Teile:

| Abschnitt         | Beschreibung  |
|-------------------|---|
| <b>Input:</b>     | Hier erfolgt die Zuordnung des Plugins zur gewünschten Eingangsgeometrie aus Cinema4D.  |
| <b>Parameter:</b> | Die Parameter dienen der Gestaltung der Haare und sind als vertikal scrollbare Liste angeordnet, um die große Anzahl möglichst übersichtlich zu halten. |
| <b>Output:</b>    | Hier wird der Export der Haare in ein definierbares VRML-Format vorgenommen.  |

## Parameter-Schieberegler

Die meisten Parameter des Haarplugins werden zwecks einfacher Bedienbarkeit als Schieberegler angeboten, für die genaue Positionierung allerdings auch als Text-Eingabefeld. Die Logik ist für alle Schieberegler die selbe:



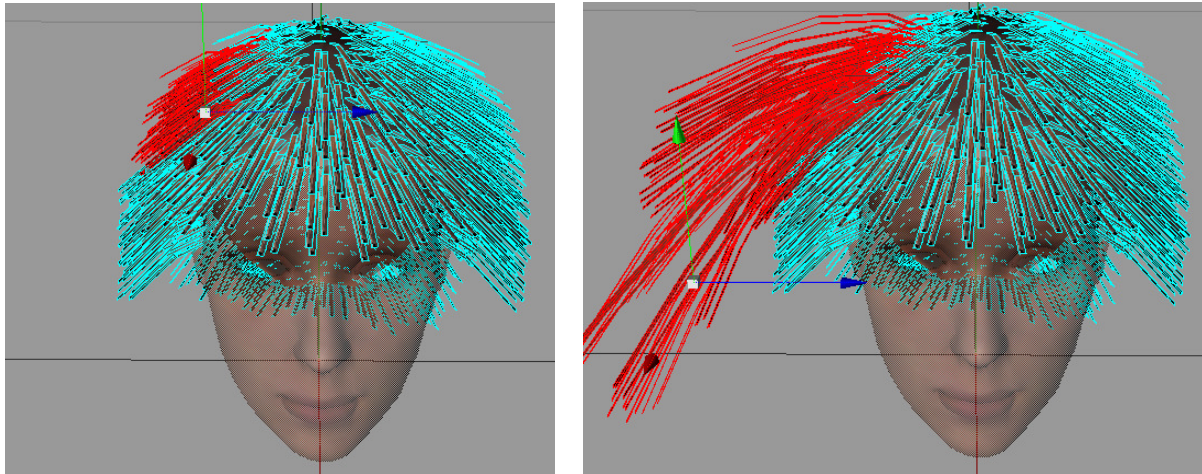
**Der linke Anschlag entspricht dem Viertel des aktuellen Wertes, der Rechte entspricht dem Zweifachen des Wertes.**

Alle Regler wirken rein multiplikativ und haben in der Mitte ihren Neutralzustand. Durch ziehen in eine Richtung wird der aktuelle Wert entsprechend obiger Angaben multipliziert. Nach dem Loslassen der Maustaste stellt sich der Regler stets wieder in der Mitte ein.

Die multiplikativen Regler haben den Vorteil, daß man damit mehr als einen Wert auf einmal verändern kann. Wenn man beispielsweise die Haarlänge verändert, so werden beim rechten Anschlag alle Haare doppelt so lang wie vorher sein, egal wie unterschiedlich lang sie sind.

## Selektierung

Für alle Regler gilt zusätzlich, daß ihre Veränderung immer stets nur auf die ausgewählten Haarpolygone greifen, sofern diese existiert. Ebenfalls ist es immer möglich, statt der Haare auch entsprechende Polygone auf der Kopfgeometrie zu markieren – die Funktionsweise ist die selbe. Dies ist für Kurzhaarfrisuren besonders interessant, da dadurch eine genauere Markierung möglich ist.



**Wenn eine Markierung vorhanden ist, wird nur auf den markierten Bereiche die einzelnen Transformationen angewandt.**

Zusätzlich gewährleistet das Textfeld rechts neben dem Regler immer eine manuelle, absolute Positionierung. Dabei erhalten alle ausgewählten Haare den selben Wert. Der Initialwert des Textfeldes wird beim ersten Klick auf den Regler der aktuellen Menge angepaßt und gibt deren Durchschnitt an.

Noch ein paar Dinge zur Funktionsweise der Markierungen: Intern arbeitet das Markierungsmodul des Plugins stets auf Basis von Kontrollpunkten, wie sie auf Seite 22 beschrieben werden. Eine Markierung über die Haare selbst ist nur möglich, da intern die zugehörigen Kontrollpunkte in der Nähe der Haarwurzeln ermittelt und selektiert werden. Die Selektion des Umfeldes ist erforderlich, da es sonst nicht möglich wäre, die Haardichte im Nachhinein zu erhöhen und die neuen Haare mit ähnlichen Parametern wie die bereits existierenden Haare auszustatten. Wie groß das Umfeld um den Anker herum ist, hängt von der Haardichte ab, so daß in etwa 1-2 Haarabstandsbreiten mitselektiert werden.

**Tipp:** Um möglichst genaue Parametrisierung zu erreichen, sollte also während der Modellierung mit einer hohen Anzahl dünner Haare gearbeitet werden. Die Dichte im Nachhinein zu verringern ist immer unproblematischer als sie nachträglich zu erhöhen.

## Beschreibung der Steuerelemente

Im folgenden werden die einzelnen Steuerelemente beschrieben. Die Kapitelnummer entspricht immer genau der Nummer im Übersichtsbild.

### ***1 Plugin an Haargeometrie anheften „Attach To Selected Head-Obj“***

Zu Beginn hat der Plugin keine Information darüber, auf welcher Geometrie die Haare erstellt werden sollen. Durch einmaliges Klicken dieses Knopfes zu Beginn verbindet der Plugin mit dem aktuell in Cinema4D ausgewählten Kopf.

Nur durch das Verbinden alleine entstehen aber noch keine Haare. Es ist ein Haarset erforderlich. Besitzt die Geometrie noch kein Haarset, so fragt der Plugin gleich nach dem Verbinden, ob er ein neues Haarset erstellen soll.

**Hinweis:** Wichtig ist, daß die ausgewählte Haargeometrie trianguliert ist. Ansonsten bekommen sie die Fehlermeldung „`WARNING: Your Mesh contains no polygons - maybe you forgot to make editable.`“. Sie können jedes Mesh triangulieren, in dem Sie in der Knotenansicht auf der ausgewählten Geometrie das Kontextmenü öffnen und auf „Make editable“ drücken.

Die Anfangszustände vieler Parameter werden in dem Moment initialisiert, in dem der Attach-Knopf gedrückt wird, da zu diesem Zeitpunkt die Größe der Kopfgeometrie vorliegt. Dies wird gemacht, da erfahrungsgemäß sehr unterschiedlich große Geometrien verwendet werden, die Haare aber möglichst immer ungefähr die richtige Größenordnung haben sollen.

**Tipp:** Falls weitere Teile des Avatars in ein und derselben Geometrie liegen, so extrahieren Sie mittels der Cinema4D Funktionen nur die für Haare relevanten Bereiche, sonst sind die Haare evtl. zu klein.

## **2 Aktuelle Kopfgeometrie**

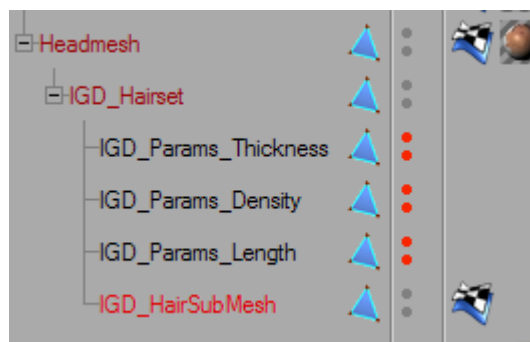
Sobald Sie sich mittels „Attach To Selected Head-Obj“ einer Geometrie angeheftet haben, wird diese zur Kontrolle in diesem Feld angezeigt. Angezeigt wird dabei die aktuelle Datei, sowie der Name des Kopf-Knotens.

## **3 Weiteres Haarsets hinzufügen „Add New Hairset“**

Mit diesem Knopf können Sie zum bestehenden Haarset ein weiteres hinzufügen. Es ist vorgesehen, auf die selbe Geometrie mehrere verschiedene Haarsets zu legen, so daß diese verschieden parametrisiert werden können.

Für jedes Haarset wird in Cinema4D ein Unterknoten „IGD\_Hairset“ angelegt, der lediglich ein leerer Sammelbehälter für weitere Unterknoten darstellt. In diesem Knoten werden sofort mehrere Unterknoten angelegt:

Der Unterknoten „IGD\_HairSubMesh“, der direkt nach dem Erstellen fokussiert wird, stellt die eigentliche Haargeometrie dar. Desweiteren werden noch einige Parametergeometrien angelegt, welche für lediglich der internen Speicherung der Haarinformationen dienen.



**Für jedes Haarset wird der Knoten „IGD\_Hairset“ samt Unterknoten angelegt.**



#### **4 Zwischen mehreren Haarsets umschalten**

Mit dem Knopf (4) können Sie zwischen allen Haarsets wechseln. Jedesmal, wenn der Knopf gedrückt wird, wird das nächste Haarset im Wechsel markiert. Der Knopf bekommt jeweils den Namen des aktuellen Haarsets. Mit jedem Umschalten wird auch automatisch der zugehörige Parametersatz geladen.

#### **5 Gleichgroße Segmente erzeugen „Auto Subdivide“**

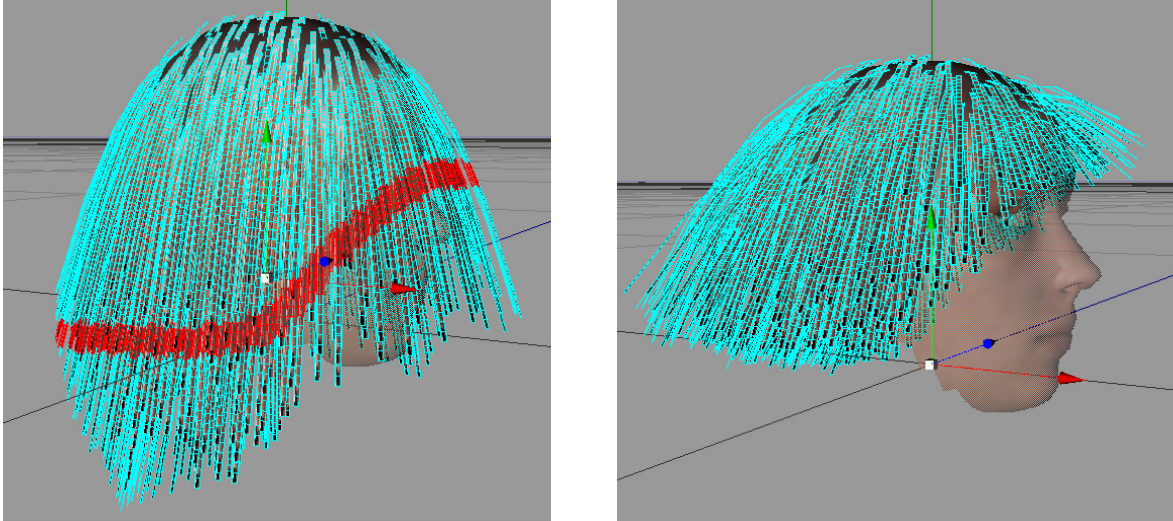
Dieser Schalter steuert die interne Geometrieerzeugung der Haare. Wird der Schalter aktiviert, so werden immer möglichst gleichlange, kleine Segmente generiert. Dies hat den Vorteil, daß beim Schneiden wesentlich präziser gearbeitet werden kann.

Der Subdivide bezieht sich nur auf die Darstellung, die interne Segmentgröße bleibt unberührt.

**Tipp:** Beachten Sie allerdings, daß der Plugin bei aktiviertem Subdivide wesentlich langsamer arbeitet und auch wesentlich mehr Speicher verbraucht. Es wird empfohlen, den Schalter nur für die Schneidefunktion zu aktivieren.

#### **6 Aktuelle Auswahl abschneiden „Cut selected Hairs“**

Ein wesentlicher Bestandteil des Frisierens ist das Schneiden. Man selektiert in Cinema4D einfach mit dem Polygonauswahl-Tool alle Polygone des Haarmeshs, die abgeschnitten werden sollen. Mit dem Knopf „Cut selected Hairs“ hat der Benutzer dann die Schere, von der jeder Frisör träumen würde. Mit einem Klick verschwinden alle selektierten Haare. Gleichzeitig werden alle freischwebenden Haarüberreste entfernt.



Nach dem Markieren erfolgt das Schneiden mit einem Knopfdruck.

**Tipp:** Es ist zu empfehlen, für das Schneiden den „Auto-Subdivide-Modus“ anzuschalten, da hier die Genauigkeit wesentlich höher ist. **Hinweis:** Beachten Sie, daß die „Entfernen“-Taste keine Schneideoperation durchführt.

### **7 Haardichte-Steuerung „Density-Subdivides“**

Mit dem Regler Density-Subdivides wird festgelegt, wie hoch die Haardichte für den angegebenen Haarbereich sein soll. Aus Geschwindigkeitsgründen ist der Plugin künstlich auf 100.000 Haare limitiert. Nebenbei sei erwähnt, daß dies in etwa der echten Haaranzahl eines Menschen in der Realität entspricht, diese Grenze sollte also für den Echtzeit-Renderbereich nicht zu niedrig liegen.

**Hinweis:** Die Haardichte sollte möglichst nur zu Beginn des Frisierens festgelegt werden, sonst ist der Erhalt der alten Frisur nicht gewährleistet. Dies liegt daran, daß die Haare über eine Zufallsverteilung erstellt werden und für ggf. neue Haarpositionen noch keine eindeutigen Parameter vorhanden sind.

### **8 Zufällige Haardichte „Density Random“**

Desweiteren ist es möglich, einen Zufallswert zu verwenden, um die Haardichte nicht überall gleich zu verteilen.

## **9 Anzeige von Strähnen und Polygonen**

Da der Plugin letztendlich die Aufgabe hat, eine echtzeitfähige Haargeometrie zu erzeugen, sollte der Benutzer immer die Anzahl Strähnen und Polygone während des Modellierens im Auge behalten.

Eine hohe Anzahl Strähnen beeinflusst z.B. beim momentanen Avalon-Haarshader einen gewaltigen Sortieraufwand, während eine hohe Polygonzahl sich zudem noch wesentlich auf die Simulation niederschlägt, da jedes Polygon 2 Massepunkte besitzt, die sowohl simuliert als auch kollisionsgetestet werden müssen.

Bei dem momentanen Avalon-Haarshader, sowie der aktuellen CantileverBeamSystem Version bewegen sich übliche Größenordnungen der Strähnenzahl bei etwa 2.000 bis 8.000, sowie Polygone zwischen 20.000 und 40.000.

## **10 Haarlänge-Steuerung „Length Scale“**

Mit dem Regler wird die Haarlänge des markierten Bereichs festgelegt.

Die anfängliche Haarlänge wird über ein festgelegtes Verhältnis zur Kopfgröße definiert. Durch die Längenveränderung werden automatisch neue Haarsegmente eingefügt, so daß unabhängig von der Haarlänge die Segmentgröße immer möglichst konstant bleibt.

## **11 Zufällige Haarlänge „Length Random“**

Wie auch bei „Density Random“ kann auf die Haarlänge ein Zufallswert gelegt werden, der die Haare etwas „ausfransen“ läßt.

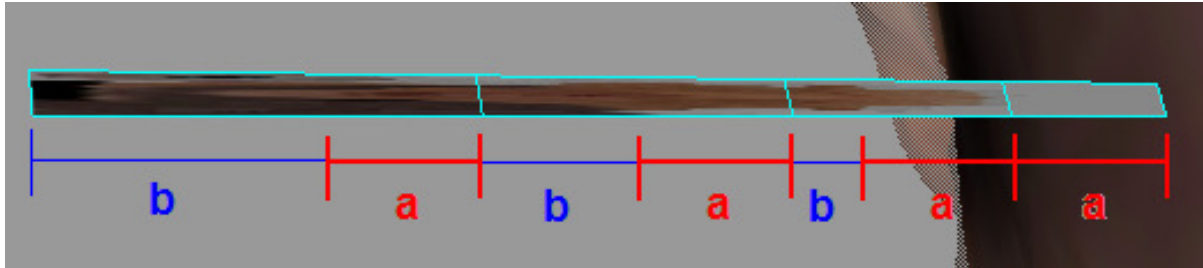
## **12 Start-Segmentlänge / 13 Segment-Verlängerungsfaktor**

Die Haare bestehen aus einzelnen Segmenten, jeweils aus einem Quad-Polygon. Die Segmentgröße ist unabhängig von der Haarlänge. Bei längeren Haaren werden die Segmente nicht größer, dennoch gibt es verschieden lange Segmente:

Sie werden über die Regler „Segment-Len“ („a“ in Bild) und „Segment-Incr“ („b“ in Bild) definiert. Die Segmentlänge des Reglers 12 beschreibt die Länge des 1. Haarsegments. Da für die Simulation möglichst viele Segmente gespart werden sollen, ist es sinnvoll, die

ersten Segmente am detaillierter, also kürzer zu machen als die hinteren, da diese kaum noch eine Biegung erfahren.

Dieser Effekt kann mittels des Schalters 13, „SegmentIncr“ (Segment-Increase, Segmentverlängerung) gesteuert werden. Ein Wert „0“ führt zu gleichgroßen Segmenten; je größer der Wert ist, desto unterschiedlicher werden die Segmentlängen einer Strähne.



Die Segmentgröße definiert sich über die Parameter „Segment-Len“ (a) und „Segment-Incr“ (b), welcher eine Segmentverlängerung bewirkt.

Trotz festgelegter Segmentlängen ist es dennoch möglich, die Haarlänge „millimetergenau“ zu definieren. Dies wird dadurch erreicht, daß bei der Segmentgenerierung abschließend die Segmente so normalisiert werden, daß die exakte Soll-Länge erreicht wird. Dadurch werden die Segmente um bis zu maximal 50% gestaucht.

### **14 Mittelscheitel-Stärke „Cortex-Scale“**

Die Haargenerierung sieht normalerweise vor, daß alle Haare senkrecht zum Kopf wachsen. Um einen Scheitel zu generieren, kann diese Richtung künstlich beeinflusst werden. Über den Regler Cortex-Scale werden die Ausrichtungen der Haare in der Nähe des Mittelscheitels auseinandergedrückt.

**Hinweis:** Für den Scheitel ist es erforderlich, daß er Kopf mit dem Gesicht nach X+ ausgerichtet ist, sonst wird die Haarverschiebung an der falschen Stelle durchgeführt.

Die interne Funktionsweise sieht vor, daß für jede Haarsträhne ein Gewichtungswert zwischen 0 und 1 berechnet wird, der angibt, wie stark sich der Scheitel auf diese Strähne auswirken soll. Dieser Wert berechnet sich wie folgt:

*Y=relativen Höhe der Haarwurzel (gemessen an der Boundingbox des Kopfes)*

*Z=Z-Koordinate der Haarwurzel (gemessen an der Boundingbox des Kopfes)*

Beide Koordinaten sind jeweils verteilt auf den Bereich -1... +1. Die Gewichtung für jede Strähne berechnet sich wie folgt:

$$G = \min(0, x^2 * y)$$

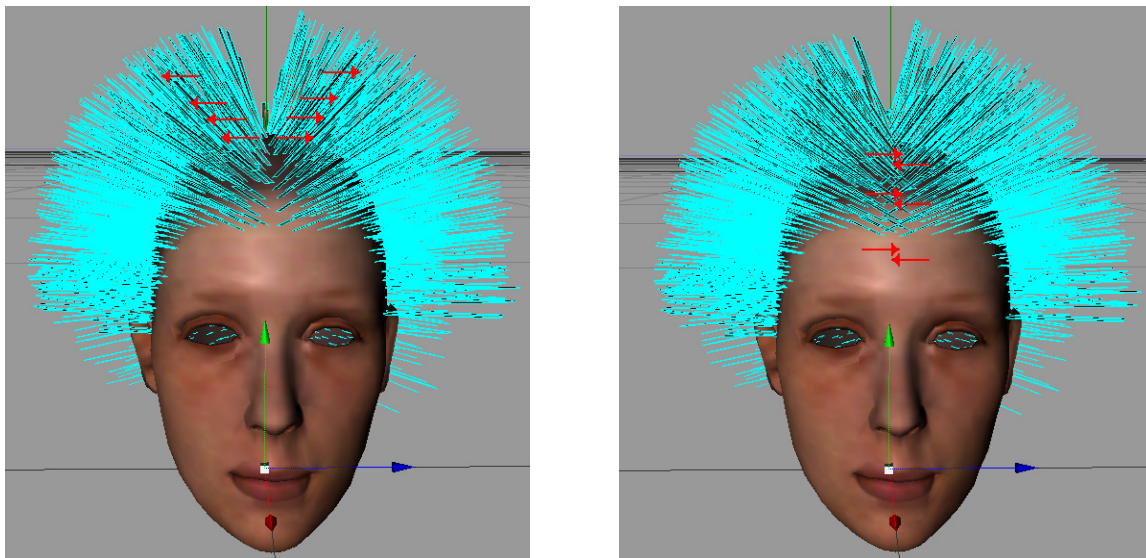
### **15 Haarüberlappung an Scheitel „Cortex Overlap“**

Der Nebeneffekt bei der Erstellung eines Scheitels liegt darin, daß die Scheitelstelle übermäßig „kahl“ erscheint. Dies kann durch eine Erhöhung der Haardichte im Bereich des Scheitels verbessert werden, was jedoch weitere Probleme mit sich bringt.

Eine bessere Möglichkeit liegt darin, mittels eines kleinen „Tricks“ die kahle Stelle aufzufüllen:

Alle Haare im Bereich des Scheitels werden entgegen ihrer Biegung auf die andere Seite geschoben. Dadurch schließt sich die Lücke etwas.

**Hinweis:** Wird der Effekt zu stark benutzt, führt dies zu übermäßigen Durchdringungen der Polygone untereinander. Beim polygonbasierten Echtzeit-Rendern kann dies zu starken Sortierproblemen und dadurch auffälligem „Flackern“ der Strähnen führen.



Links die Verwendung der Scheitelfunktion. Es bildet sich eine kahle Stelle, die mit der Überlappung (rechts) entfernt werden kann.

### **16 Basisfarbe der Haare „Global Color“**

Das Farb-Steuererelement funktioniert genauso wie ein üblicher Cinema4D-Farbregler und erlaubt sowohl die Veränderung der Helligkeit, sowie R,G,B einzeln. Das Resultat ist stets die Farbe, die links angezeigt wird. Diese wird als Basisfarbe für die Haarwurzeln verwendet.

### **17 Aufhellung der Haare „Color-Increment“**

In der Realität hat ein Haar meist keine einheitliche Farbe, da besonders die Spitzen durch äußere Einflüsse ausbleichen, während der Ansatz meist am dunkelsten ist.

Um diesen Effekt nachzubilden, kann neben der Basisfarbe der Haare auch eine „Aufhellungsfaktor“ angegeben werden. Dieser bewirkt, daß die Haare entlang der Strähne linear heller werden.

Die resultierende Farbe wird je nach verwendetem Shader und Exporteinstellungen dann mit der Texturfarbe vermischt.

### **18 Zufällige Haarfarbe „Random Brightness“**

Zusätzlich ist es möglich, eine Zufallskomponente in die Haarfarbe mit einzubeziehen.

### **19 Haardicke-Steuerung „Thickness Scale“**

Um Haare in Echtzeit rendern zu können, müssen Vereinfachungen getroffen werden.

Um die gewaltige Anzahl der Haare zu reduzieren, werden hier mehrere Haare zu einer Strähne zusammengefaßt, welche während der Simulation und des Renderns als ein

Objekt betrachtet wird. Durch eine geeignete Textur mit Alphakanal und einem

leistungsstarken Shader kann dennoch der Eindruck vermittelt werden, die Strähne

bestehe aus einer Vielzahl von Haaren. Der Haardicken-Regler (nicht zu verwechseln mit dem Kollisionsgrößenregler) bestimmt die Breite der einzelnen Strähnen.

**Tipp:** Es müssen nicht alle Strähnen gleich breit sein, die Strähnendicke kann lokal unterschiedlich gestaltet werden, in dem jeweils nur ein Teil der Haare selektiert wird.

Dadurch lassen sich beispielsweise am Hinterkopf wenige, dickere Strähnen anlegen und im Gesicht mehr Dünne, wodurch bei gleicher Strähnenanzahl eine höhere Qualität erreicht werden kann.

**Hinweis:** Zu unterschiedliche Strähndicken führen allerdings unter Umstände zu einer unschön gestreckten Textur, da diese natürlich nur für eine bestimmte Breite vorgesehen ist.

## **20 Zufällige Dicke „Thickness Random“**

Zusätzlich zur obigen Haardicke kann ein Zufallswert beigesteuert werden, so daß die die Haare etwas unterschiedlich gestaltet werden.

## **21 Neue Kollisionsobjekte erstellen „Add Collision Sphere“ / „Add Collision Plane“**

Auch wenn für den Benutzer die Haare immer auf einem bestehenden Kopf modelliert werden, weiß die Haarsimulation im Inneren zunächst nichts davon. Um zu verhindern, daß die Haare in die Kopfgeometrie eindringen, können sogenannte Kollisionsobjekte angelegt werden, die beim Rendern unsichtbar sind und nur der Simulation übergeben werden.

Der Hairdesigner sieht die Verwendung von Kollisionskugeln, Ellipsoiden und Ebenen vor. Durch Drücken des Knopfes „Add Collision Sphere“ erstellt der Plugin eine Cinema4D-Kugel und gliedert diese als Knoten in die Baumstruktur ein. Diese Kugel kann beliebig verschoben und skaliert werden.

Bei der nächsten Simulation werden automatisch die neuen Objektkoordinaten übermittelt, so daß die Haare von allen erstellten oder veränderten Kugeln abprallen. Um eine Ellipsoide zu erzeugen, muß einfach eine bestehende Kugel durch die Cinema4D-Funktion „Make Editable“ in ein Polygon-Mesh umgewandelt werden, welches unterschiedliche Skalierung einzelner Achsen zuläßt.

Zusätzlich besteht auch die Möglichkeit, Ebenen als Kollisionsobjekte anzulegen, um weitere Szenarien nachbilden zu können.

**Hinweis:** Jede Haarmasse wird stets gegen alle Kollisionsobjekte getestet, was die Simulation bei zu großer Anzahl von Kollisionsobjekten in die Knie zwingen kann. Besonders mit der Verwendung von Ellipsoiden sollte nicht zu leichtfertig umgegangen werden, da diese ca. das 3- bis 5-fache der Zeit von Kugeln zur Berechnung benötigen.

## **22 Kugelanzahl-Kontrolle**

Hier wird die Anzahl der Kollisionsobjekte angezeigt, die momentan in der Simulation verwendet werden. Die Aufteilung erfolgt in Kugeln („Spheres“), Ebenen („Planes“) und Ellipsoiden („Ellipsoids“).

## **23 Basiskollisionsgröße der Haarsträhnen „Hair Bounding Size“**

Eine Haarmasse wird in der Simulation nicht als Punkt betrachtet, sondern als Kugel mit einem bestimmten Radius. Dadurch wird erreicht, daß die Haare einen gewissen Mindestabstand vom Kopf einhalten. Dieser Radius ist auf einen üblichen Standardwert eingestellt, kann aber auch mittels des Reglers „Hair-Bounding-Size“ verändert werden.

## **24 Erhöhung der Haarkollisionsgröße „Hair Bounding Increment“**

Normalerweise haben alle Haarmassen entlang eine Strähne den gleichen Radius. Es ist aber unter Umständen sinnvoll, die Radiusgröße entlang der Strähne zu steigern. Dies ist z.B. sinnvoll, da dadurch das Verdeckungsproblem des Echtzeit-Shaders reduziert werden kann, sowie dem Haar mit einem gewissen „Volumen“ versehen wird.

**Hinweis:** Das CantileverBeam-Simulation in Avalon ermittelt die Haardicken-Erhöhung unter Umständen bereits intern, wodurch der Parameter keine Wirkung hat.

**Hinweis:** Bei der Verwendung von Ellipsoiden ist performancetechnisch bedingt kein Haardicken-Erhöhung möglich.

## **25 Zufällige Haarkollisionsgröße „Hair-Bounding-Random“**

Zufallskomponente der Haarkollisionsgrößen. Dieser Schalter ist nur funktional, wenn die Simulation individuelle Haarkollisions-Parameter unterstützt.

## **26 Haarsteifheit „Bending Scale“**

Mit diesem Regler kann ein Wert zwischen 0.0 und 1,0 festgelegt werden, der die Steifheit der Haarsträhnen angibt. Dabei entspricht 1.0 einem starren, komplett unbiegbarem Haar, und 0.0 einem Haar komplett ohne Biegungswiderstand. Übliche Werte liegen bei Langem Haar bei ca. 0,2- 0,3 sowie bei Kurzhaarfrisuren bei 0,4-0,6.



### **27 Zufällige Steifheit „Bending Random“**

Zufallskomponente der Haarsteifheit. Dieser Schalter ist nur funktional, wenn die Simulation individuelle Haarkollisions-Parameter unterstützt.

### **28 Schwerkraftstärke „Gravity“**

Der Regler „Gravity“ gibt die Stärke der in der Simulation wirkenden Schwerkraft an. Die Schwerkraft wirkt stets in die Richtung Y-Negativ.

### **29 Manuelle Simulation „Test Simulation with 10 Steps“**

Durch drücken dieses Knopfes werden augenblicklich 10 Simulationsschritte auf das aktuelle Haarmesh ausgeübt. Vorher werden noch alle Kollisionsobjekte aus Cinema-4D gelesen.

### **30 Automatische Simulation „Auto Simulate“**

Diese Checkbox gibt an, ob automatisch nach jeder Veränderung eine Simulation durchgeführt werden soll.

| <b>Modus</b> | <b>Beschreibung</b>  |
|--------------|--|
| Aus          | Es wird nie eine Simulation durchgeführt, außer der Benutzer betätigt den Knopf „Test Simulation with 10 Steps“.   |
| An           | Bei jeder Veränderung eines Reglers wird eine Simulation mit 10 Schritten durchgeführt, so daß sich der Benutzer gleich vom Simulationsergebnis überzeugen kann. |

### **31 Exportmodus wählen „Exportmode“**

Der Exportmodus beschreibt, wie das aktuelle Haarset in die (z.B. für Avalon bestimmte) Export-VRML-Datei geschrieben werden soll. Hier gibt es 3 verschiedene Möglichkeiten:

| <b>Exportmodus:</b> | <b>Beschreibung</b>                       |
|---------------------|---|
| <b>Dynamic</b>      | Die Haare werden an das Simulationssystem |

|             |  |
|-------------|--|
|             | (CantileverBeamSystem in Avalon) übergeben und darin komplett dynamisch gehandhabt. Vor dem Export werden die Haare mit 10 Simulationsschritten á 0.1 Sekunden vorsimuliert, so daß diese bereits ihre etwaige Endposition auf dem Kopf haben.   |
| <b>Fix</b>  | Das Haarset wird als feststehendes, bereits simuliertes Mesh herausgeschrieben; Die VRML-Engine führt hierfür keine weitere Simulation durch.<br>Der Modus ist für nahezu unbewegliche Kurzhaarfrisuren und weiter entfernte Köpfe gedacht, da hier die Performance durch eine weitere Simulation nicht in Mitleidenschaft gezogen werden soll.<br>Zu beachten ist, daß für die Fixhairs gesonderte Parameter im Export-Template zur Verfügung stehen. |
| <b>None</b> | Das aktuelle Haarset wird gar nicht exportiert. Es ist nur im Cinema4D-Tool sichtbar. Dies ist während des Arbeitens manchmal sinnvoll, da man so zwischen mehreren alternativen Konfigurationen umschalten kann.  |

### ***32 Ausgabedatei wählen***

Die Eingabezeile, sowie der „...“-Knopf auf der rechten Seite dienen der Auswahl einer Zieldatei, in die der VRML-Export der Haare geschrieben werden soll. Dabei können beliebige Dateinamen und Zielverzeichnisse angegeben werden. Die VRML-Extension wird dem Dateinamen nicht automatisch hinzugefügt. Nach der Auswahl eines neuen Namens mittels Dialog wird die Datei augenblicklich exportiert.

### ***33 Jetzt exportieren „Export to VRML“***

Der Knopf ist gedacht, um die aktuelle Frisur direkt in das VRML-Format zu exportieren. Während des Arbeitens in Cinema4D kann man sich so schon mittels eines Knopfdruckes und dem anschließenden Starten von z.B. SAV.EXE vom Resultat überzeugen. Beim Export werden alle Haarsets noch einmal für 10 Iterationen simuliert, so daß diese bereits in etwa ihre Endlage erreicht haben. Dies kann evtl einige Sekunden dauern. Ist

der Export fehlerfrei verlaufen, bekommen Sie die Meldung: „`successfully exported to file`“.

Um den Export möglichst variabel zu gestalten, bedient sich der Plugin einer programmierbaren Template-Datei, die stadardmäßig im Pluginverzeichnis liegt. Hier können spezifische Änderungen vorgenommen werden. Diese werden auf Seite 24 beschrieben. Um zu erreichen, daß mehrere Modelle unterschiedliche Konfigurationen erfahren, kann die Template-Datei automatisch vom Plugin neben die Cinema-4D Datei kopiert werden und deren Namen mit der Erweiterung „.template“ bekommen. Ob dies erfolgen soll, wird über die Checkbox „Copy Template To C4D-File“ ausgewählt.

## Undo

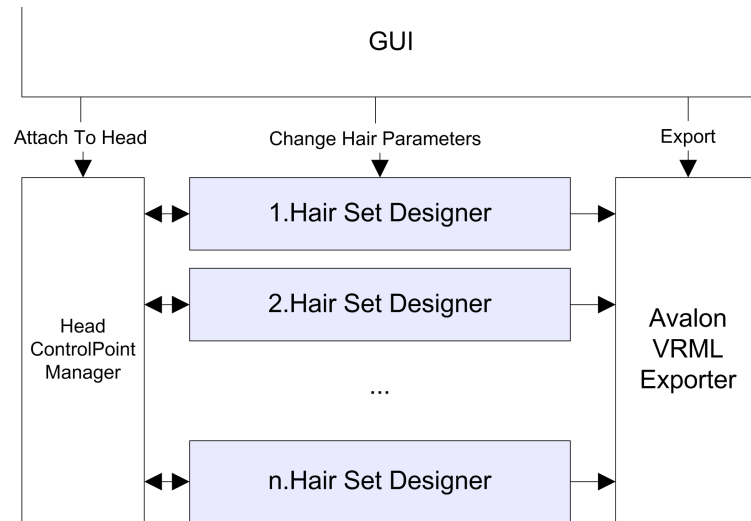
Eine Besonderheit ergibt sich bei Verwenden der „Rückgängig“ (UNDO)-Funktion in Cinema-4D: Nach Ausführung der UNDO-Funktion verliert der Plugin sämtliche Bindungen zum vorliegenden Haarobjekt, so als wenn die Datei geschlossen wurde. Um dennoch nach eine UNDO weiterzuarbeiten, „klinkt“ sich der Plugin erneut an die Geometrie an und rekonstruiert alle vorliegenden Parameter.

**Hinweis:** Da der UNDO erst bei der nächsten Plugin-Operation (wie beispielsweise einer Slider-Bewegung) „bemerkt“ wird, können vorherige Selektierungen zurückgesetzt werden.

**Tipp:** Drücken sie nach Verwendung der UNDO-Funktion zunächst die „Attach to Selected Hair-Obj“-Knopf, um den Plugin manuell an die Cinema4d-Geometrie anzuheften.

## Der interne Aufbau des Plugins

Der Plugin ist modular aufgebaut und besteht im wesentlichen aus folgenden Haupt-Komponenten:

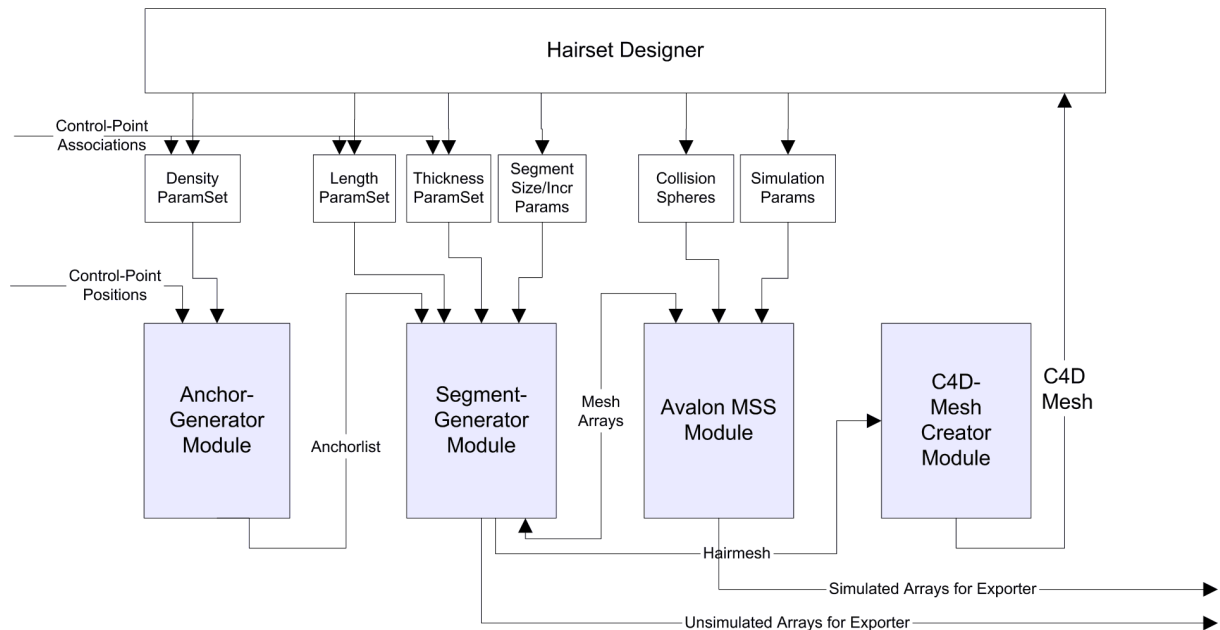


**Gesamtaufbau des Plugins**

| Komponente             | Kurzbeschreibung  |
|------------------------|---|
| GUI:                   | Stellt die Benutzeroberfläche dar und besitzt nur Bedienlogik.                                  |
| Control-Point-Manager: | Die Kommunikation mit der Kopf-Geometrie und die Parameter-Logiken liegen in dieser Komponente. |
| Hair-Set-Designer:     | Im Hair-Set-Designer befindet sich die eigentliche „Logik“ des Haargenerator.                   |
| Avalon-VRML-Exporter:  | Der VRML-Exporter dient dem Export aller Haarsets nach Avalon.                                  |

### ***Hair-Set-Designer***

Der den Hauptteil des Plugins wird durch den Hair-Set-Designer ausgemacht. Außerdem kann es vom Hair-Set-Designer im Gegensatz zu den anderen Komponenten mehrere Instanzen geben. (Durch Drücken der Knopfes „Add new Hairset“ wird jeweils eine weitere Instanz hinzugefügt). Der Aufbau der Komponente sieht folgendermaßen aus:



## Aufbau einer Hair-Set-Designer-Instanz

Der Datenfluß findet im wesentlichen von links nach rechts statt. Der Hair-Set-Designer besteht im wesentlichen aus folgenden Subkomponenten:

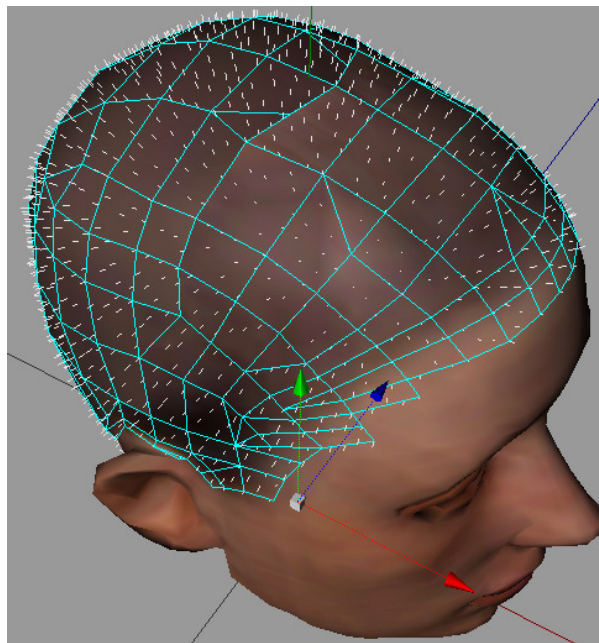
| Sub-Komponente     | Beschreibung  |
|--------------------|---|
| Anchor-Generator:  | Der Ankergenerator erzeugt mittels der Kontrollpunkte auf dem Kopfmesh eine Liste von Haarwurzeln, die über die Parametersätze z.B. mit unterschiedlicher, lokaler Dichte ausgestattet sein können. Das Ergebnis dient dem Segment-Generator.   |
| Segment-Generator: | Dieses Modul ist für die Erstellung der einzelnen Haarsträhnen aus den Haarwurzeln verantwortlich. Hier werden eine große Reihe von Parametern mit einbezogen. Das Ergebnis ist eine fertige Haargeometrie, sowie die Metadaten in Form einer Strähnenliste für spätere Zuordnungen.                                    |
| MSS-Simulation:    | In diesem Modul ist die aus Avalon stammende Haarsimulation „CantileverBeamSystem“ untergebracht, welche die Haare in Echtzeit simuliert. Die simulierten Ergebnisse werden zurück in den Ausgabepuffer des Segment-Generators geschrieben. In diesem Modul findet auch die Kommunikation mit den in Cinena4D liegenden |

|                   |   |
|-------------------|---|
|                   | Kollisionskugeln statt.   |
| C4D-Mesh-Creator: | Der Mesh-Creator erstellt aus der vom Segment-Generator generierten Geometrie eine Cinema4D-Mesh und stellt dies auf der Cinema4D-Oberfläche dar. Hier werden auch Erweiterungen wie die „Auto-Subdivision“ durchgeführt, die eine rein visuelle Unterteilung des Meshes für den C4D-Benutzer vornimmt. |

### ***Control-Point-Manager***

Der Hair-Set-Designer hat eine enge Verbindung zum Control-Point-Manager, der sie mit entsprechenden Parametersätzen versorgt. Die Aufgabe des Control-Point-Managers liegt darin, eine möglichst homogene Abbildung der Kopfoberfläche zur Verfügung zu stellen, auf der eine Punktgenaue Parametrisierung stattfinden kann.

Da die Polygondichte der Kopf-Geometrie sehr unterschiedlich sein kann, erstellt der Control-Point-Manager alternativ ein neues Netz von Kontrollpunkten auf der Polygonoberfläche, welches sich durch eine homogene Dichte auszeichnet. Diese Dichte wird relativ zur Kopfgröße gewählt:



**Der Control-Point-Manager legt ein Netz von Kontrollpunkten auf der Kopfgeometrie an, welches eine gleichmäßigere Dichte hat und für die weitere Parametrisierung verwendet wird.**

Jeder dieser Kontrollpunkte kann während der Arbeit mit dem Plugin unterschiedliche Parameter wie z.B. Haarlänge, Dicke, etc. besitzen. Jeder Parameter, der „pro Kontrollpunkt“ vorhanden ist, nennt sich „Paramset“ und wird als gesondertes Mesh abgespeichert. Die Subkomponenten des Hair-Set-Designers greifen dann auf genau diese Parametersätze zu, wenn sie die einzelnen Haare individuell erstellen. Momentan werden folgende Parametersätze verwendet, um individuelle Haareigenschaften zu erreichen:

| Name           | Mesh                 | Datentyp | Funktion                |
|----------------|----------------------|----------|-------------------------|
| ParamLength    | IGD_Params_Length    | FLOAT    | Individuelle Haarlänge  |
| ParamDensity   | IGD_Params_Density   | FLOAT    | Individuelle Haardichte |
| ParamThickness | IGD_Params_Thickness | FLOAT    | Individuelle Haardicke  |

Alle anderen Parameter werden als Attribute des Knoten „IGD\_Hairset“ abgespeichert und diesen den uniformen Parametern des Haargenerators:

| Parameter       | Datentyp | Funktion   |
|-----------------|----------|--|
| ExportMode      | INT      | Nummer des Exportmodus (0=None, 1=Fix, 2=Dynamic)                  |
| GravityScale    | FLOAT    | Stärke der Schwerkraft   |
| CollisionScale  | FLOAT    | Größe der Kollisionskugeln der Haarmassen in Simulation            |
| CollisionIncr   | FLOAT    | Größenänderung der Kollisionskugeln der Haarmassen in Simulation   |
| CollisionRandom | FLOAT    | Zufallswert der Kollisionskugelgrößen der Haarmassen in Simulation |
| BendScale       | FLOAT    | Steifheit der Haare  |
| ColorRandom     | FLOAT    | Zufallswert der auf Haarfarbe                                      |
| Scheitel1       | FLOAT    | Stärke des Scheitels   |
| Scheitel2       | FLOAT    | Überlappung des Scheitels  |
| SegmentSize     | FLOAT    | Start-Segmentgröße   |
| SegmentIncr     | FLOAT    | Erhöhung der Segmentgröße  |
| DensityRandom   | FLOAT    | Zufallswert der Haardichte   |
| LengthRandom    | FLOAT    | Zufallswert der Haarlänge  |
| ThicknessRandom | FLOAT    | Zufallswert der Haardicke  |

## Export-Templates

Um den Export nach Avalon möglichst flexibel konfigurierbar zu gestalten, verwendet der Plugin ein sog. „Export-Template“. Dabei handelt es sich um eine VRML-Datei, welche die Struktur der Ausgabedatei klar vorgibt und nur mit entsprechenden Werten gefüllt wird.

Der Plugin parst diese Datei und sucht nach bestimmten Schlüsselwörtern, die dann durch einen dynamischen Inhalt ersetzt werden. Momentan werden folgende Schlüsselwörter verwendet:

| Schlüsselwort         | Beispiel  | Beschreibung   |
|-----------------------|---|--|
| ?HAIRVERTICES?        | 3.44423, 11.22112, -<br>4.12111, (...)  | Stellt ein Array aller verwendeten Vertexkoordinaten zur Verfügung, durch Komma getrennte Fließkommazahlen.  |
| ?INFO?                | # Generator XXX<br># Version XXX<br># Strands XXX<br>(...)                                      | Schreibt einen Header mit wichtigen Informationen als Kommentarblock in die Datei. Unter anderem mit der Programmversion, der Anzahl Strähnen, Polygonen, etc.   |
| ?HAIRCOLORS?          | 0.94423, 0.22112,<br>0.12111, (...)   | Wie HAIRVERTICES, allerdings mit Farbwerten für jeden Vertex, im Bereich von [0...1]   |
| ?HAIRQUADS?           | 1,2,3,4,-1, 4,23,1,2,-1,<br>(...)   | Schreibt ein Index-Array für die komplette Haargeometrie. Die Indices beginnen mit 0 und beziehen sich auf die Vertex-Arrays. Jedes Quad wird mit dem 5.Index „-1“ terminiert.   |
| ?DIRECTHAIRQUADS?     | 1,2,3,4,-1, 5,23,5,3,-1,<br>39,9,6,88,-1, 6,43,4,5,-<br>2, 12,2,3,25,-1,<br>46,7,1,22,-2, (...) | Wie HAIRQUADS, allerdings werden die Indices sortiert nach Strähnen aufgeführt, so daß nach den Quads wie gewohnt eine „-1“ kommt und nach jeweils dem letzten Quad in der Strähne eine „-2“.<br>Diese Version sollte immer HAIRQUADS bevorzugt werden, da Avalon hier eine optimiertere Laderoutine verwenden kann. |
| ?HAIRCOLLSPHERES?     | 13.11414 11.15332<br>14.22321, 4.11121 0<br>(...)   | Gibt ein Array aller Haarkollisionskugeln zurück, beginnend für jeden Eintrag mit XYZ des Zentrums, sowie durch Komma getrennt R, dem Radius der Kugel und einer zusätzlichen „0“.   |
| ?HAIRCOLLSPHERECOUNT? | 2   | Gibt die Anzahl der Kollisionskugeln   |



|                    |  |  |
|--------------------|--|--|
|                    |  | zurück.  |
| ?HAIRBEND?         | 0.55332  | Gibt den für die Simulation erforderlichen Biegungswert zurück. Diese liegt zwischen 0 und 1.  |
| ?HAIRRADIUS?       |  | Gibt den Radius der Strähnen zurück, der in der Simulation verwendet wird. Min und Maxradius wird zwar im Plugin generiert, in der Schnittstelle jedoch im Moment nicht unterstützt.                               |
| ?HAIRGRAVITY?      | 0 -9.81 0  | Gibt den Schwerkraftvektor zurück, XYZ jeweils als Fließkommazahl mit maximal 5 Stellen Genauigkeit.   |
| ?FIXHAIRVERTICES?  | 3.44423, 11.22112, -4.12111, (...)                 | Gibt das vollständige Array mit allen Vertexkoordinaten der Haargeometrie zurück, sofern diese mit „Fix“ exportiert wurde. Der Aufbau ist gleich zu dem von HAIRVERTICES.  |
| ?FIXHAIRCOLORS?    | 0.94423, 0.22112, 0.12111, (...)                   | Analog zu FIXHAIRVERTICES, allerdings mit den Farbwerten RGB im Bereich 0...1.   |
| ?FIXHAIRTANGENTS?  | 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, (...) | Analog zu FIXHAIRVERTICES, allerdings mit dem Tangentenvektoren, normalisiert auf Länge 1. Diese werden von der Simulation berechnet und diesen dem Shader bei der Darstellung.                                    |
| ?FIXHAIRNORMALS?   | 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, (...) | Analog zu FIXHAIRVERTICES, allerdings mit dem Normalenvektoren, normalisiert auf Länge 1. Diese werden von der Simulation berechnet und diesen dem Shader bei der Darstellung.                                     |
| ?FIXHAIRCENTERS?   | 10.0, 40.0, 40.0, 20.0, 23.0, 4.0,                 | Enthält die Mittelpunkte aller Quads, notiert in XYZ. Diese werden von der Simulation berechnet und diesen dem Shader bei der Sortierung der Haare nach Z. Achtung: Die Feldlänge beträgt die Anzahl der Strähnen. |
| ?FIXHAIRTEXCOORDS? | 0.0, 0.123, 1.0, 0.123, (...)                      | Enthält die Texturkoordinaten für jeden Vertex, notiert in UV im Bereich von 0...1. Die V-Achse liegt entlang der Strähne.   |
| ?FIXHAIRQUADS?     | 1,2,3,4,-1, 4,23,1,2,-1, (...)                     | Analog zu HAIRQUADS, hier sind alle Indices der Fixhaargeometrie enthalten.  |

**ACHTUNG:** Hinter den Schlüsselwörtern muß ein Leerzeichen eingegeben werden, sonst kann eine fehlerfreie Ausgabe nicht garantiert werden.