2020
1
ISSN 1433-2620 > B 43362 >> 24. Jahrgang >>> www.digitalproduction.com
Publiziert von Pixeltown GmbH

Deutschland	€ 17,90
Österreich	€ 19,–
Schweiz	sfr 23,–

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

JANUAR | FEBRUAR 01:2020

Houdini
Houdini 18, Labs, Solaris, Modeling & Wordart

KI
Was steckt hinter dem Hype?

und vieles mehr
Dailies in Scratch, Paint Tools, Katana, BMD 6K ...

4 194336 217907 01

Zeitdruck. Übermüdet. Und dann noch Stricken!

Wer in dieser Branche arbeitet, dem ist dies zumeist ein altbekannter Ablauf. Doch wie können wir aus diesem Teufelskreis entkommen? Da die stets immer weiter steigende Nachfrage nach Content mit dem bisherigen Angebot bald nicht mehr abdeckbar sein wird, ohne die Qualität zu vernachlässigen, müssen wir als Content Creators unsere Workflows adaptieren. Doch welche Variablen können wir manipulieren, um diesem Problem entgegenzuwirken? Die einzige Antwort hierauf ist: Automatisierung.

von Luca Scheller

Ob Film-, Commercial- oder Spielebranche, alle haben gemeinsam, dass sich viele Arbeitsschritte wiederholen. Langsam, aber stetig beschleunigend entwickelt sich ein Automatisierungstrend auch in dieser Branche. Das Verhältnis zwischen Artists und Pipeline TDs/Software Developern fängt langsam an, sich zu invertieren. Das alles wird häufig unter dem Thema Pipeline zusammengefasst. Doch wer hat heutzutage Zeit, alles auf seine Bedürfnisse zurechtzuschneiden? Um dieser Problematik entgegenzukommen, habe ich in den letzten paar Jahren immer wieder kleinere Tools gebaut, um meine Arbeitsprozesse zu beschleunigen. Als letztes Jahr Moritz Schwind und Henry Foster MOPs aka Motion Operators, ein Motion Graphics Toolset für Houdini, veröffentlichten, kam mir die Idee, bestehende Tools und Ideen produktionsfertig zu machen und diese auch als Open Source zu veröffentlichen. Hiermit präsentiere ich LYNX VFX Tools – eine Sammlung diverser VFX-Tools, um schneller hochwertigen Output zu generieren. Alles

nachstehende ist frei verfügbar zum Download auf Github. Alle Tools haben sowohl eine vollständige Dokumentation als auch ausführliche Tutorials mit Beispieldateien. Jeder kann dazu beitragen, Feedback geben und jederzeit in die laufende Entwicklung Einblick nehmen. Ihr seid herzlich eingeladen, euch aktiv zu beteiligen, damit wir es als Community jeden Tag ein bisschen einfacher haben. Bisher ist der primäre Fokus, kleinere Arbeitsschritte durch hilfreiche Tools zu vereinfachen. Hierzu gibt es derzeit drei Houdini-Tools, welche ich nachfolgend präsentiere.

Die aktuelle Installationsanleitung und alle relevanten Informationen gibt es stets unter **github.com/LucaScheller/VFX-LYNX**. Für die fortgeschrittenen Nutzer und IT-Affinen unter euch: Man kann das Projekt auch Git-klonen und zum Dev (Development) Branch wechseln, um die latest & greatest Features zu testen. Da dies aber der Dev Branch ist, können natürlich aber auch einige Bugs auftreten.

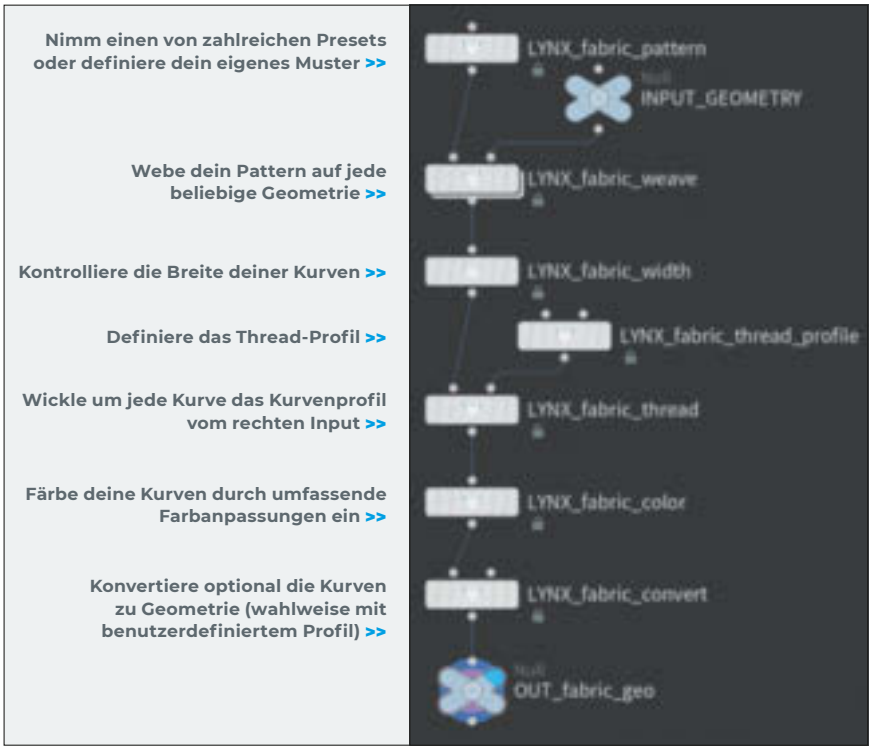


Abb. 2: Der LYNX_fabric Node Tree erzeugt durch das „LYNX fabric Configure“ Shelf Tool.

LYNX_fabric – Spielendleicht Webmuster erstellen

Wer kennt sie nicht? 3D-Web- und Strick-Setups sind seit Jahren immer wieder im Trend, spätestens zu den jährlichen Sportschuh-Werbungen sieht man wieder, wie sich ein Objekt mit Leichtigkeit aus dem nichts zusammenwebt. Doch auch in der Filmbranche werden sie für Nahaufnahmen immer wieder benötigt. Um das Ganze für Motion Designer und auch für CG-Generalists intuitiv bedienbar zu machen, ist LYNX_fabric kreiert worden. Der modulare Aufbau und die maximale Performance als Designziel hinter den Nodes erlaubt es jedem, sich an dieser Art von Effekt auszutoben.

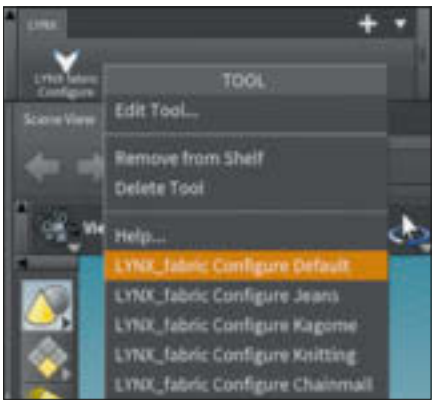


Abb. 3: Das LYNX Shelf bietet hilfreiche Presets.

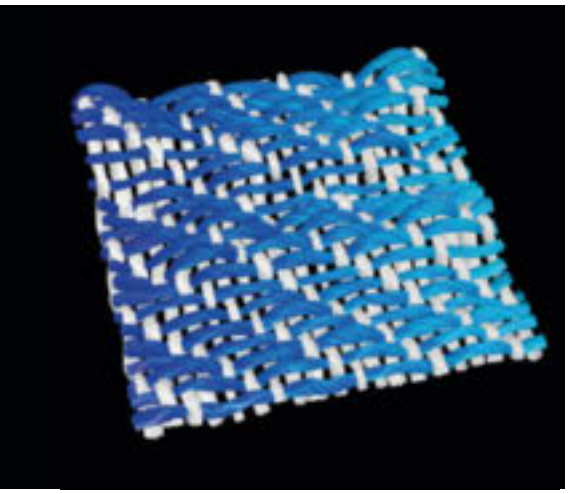


Abb. 5: Das Jeans-Preset erzeugt durch das „LYNX fabric Configure“ Shelf Tool

Die hier präsentierten Artikel sind nebenbei bemerkt vollständig mit LYNX_fabric generiert worden, die Szene hierzu liegt auch im Git Repository. Falls du also jemals den grauenvollsten Weihnachtspulli aller Zeiten designen wolltest, hast du jetzt die Chance!

The Basics

In **Abb. 2** sehen wir den standardmäßigen Aufbau der LYNX_fabric Nodes. Der

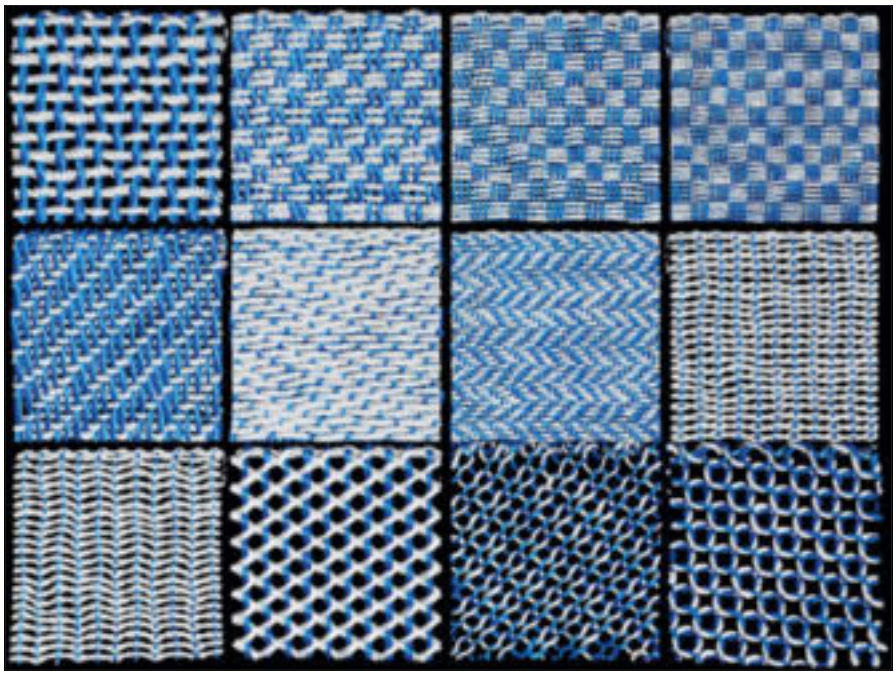


Abb. 4: Mit den Presets des LYNX_fabric_pattern Nodes lässt sich schnell der richtige Look finden.

Abb. 6: Texture Baking Viewport Normal vs. Displacement Map Preview (Seam-Fehler sind Viewport-bedingt)

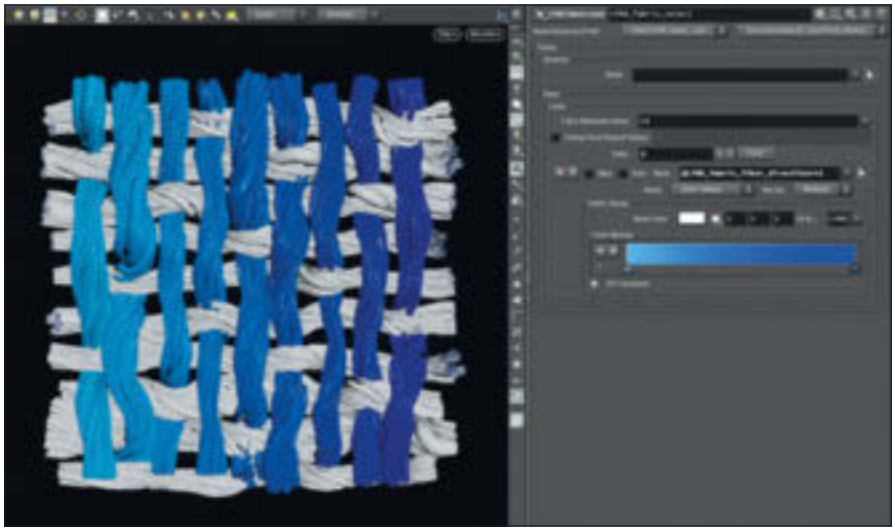


Abb. 7: Mit LYNX_fabric_color können wir unserem Setup mehr (farbliche) Variation verleihen.

schnellste Weg, diesen zu kreieren, ist über das mit LYNX mitgelieferte LYNX Shelf (**s. Abb. 3**).

Hier kannst du mit einem Knopfdruck auf den „LYNX fabric Configure" Shelf Button jede beliebige Polygongeometrie mit einem Webmuster bestücken. Die einzige Voraussetzung ist es, nicht überschneidende UVs zu haben (UDIMs werden unterstützt).

Ein Rechtsklick auf den Button lässt außerdem zwischen verschiedenen Presets wählen, vom Kettenhemd bis zum Jeansstoff lassen sich so schnell verschiedene Variationen erzeugen.

Die Nodes LYNX_fabric_width, LYNX_fabric_color, LYNX_fabric_thread und LYNX_fabric_resample sind dabei in jeder beliebigen Reihenfolge zu verbinden, denn solange wir den Haken bei „Transfer Attributes" gesetzt haben, werden die Attribute immer Downstream mit durchgeschleift und verrechnet.

Next Steps: Texture Baking

Wenn wir doch das Problem haben sollten, zu viele Polygone zu generieren, besteht die Möglichkeit, jeden Pattern als Textur zu baken. Hierzu liegt auch eine Ready-to-Render-Szene im Git Repository frei zur Verfügung, die eine Diffuse Color, Normal und Displacement Map bakt. Optional lassen sich natürlich AOVs (Arbitrary Output Variables) wie Metallicness, Ambient Occlusion oder Curvature auch mit baken. Der Pattern wird hierzu erst kachelbar gemacht und anschließend in Polygone umkonvertiert und gebakt. Bei einer Auflösung von 2K braucht das Baking etwa 4 Minuten (natürlich abhängig von dem Rechner und der Renderqualität). In der bereitgestellten Datei können wir auch die Maps mit dem voreingestellten Shader im Viewport visualisieren. Hierbei lassen sich neben Normal Maps auch die Displacement Maps interaktiv in Echtzeit anzeigen. Die bereitgestellte Szene normalisiert (skaliert die Pixelwerte, sodass sie zwischen 0 und 1 liegen) auch automatisch die Displacement Map beim Rendern durch ein Houdini Compositing Operators Network (COP Net).

Cache Your Weave

Eine Besonderheit beim LYNX_fabric_weave Node ist, dass wir in den Node hineintauchen können, um das Webmuster zu cachen, bevor es auf die Geometrie transformiert wird. Dies hat den Vorteil, dass, wenn wir unsere Datei neu laden oder die Animation der Input-Geometrie verändern, kein Re-Cook passieren muss. Hierbei müssen wir aufpassen, dass wir die UVs nicht verändern. Ein Anwendungsbeispiel hierfür ist ein Charakter-Asset, das für mehrere Shots gecacht werden muss. Wir berechnen also nur einmal das Pattern und transformieren dieses dann mit der jeweiligen Geometrie der jeweiligen Shots.

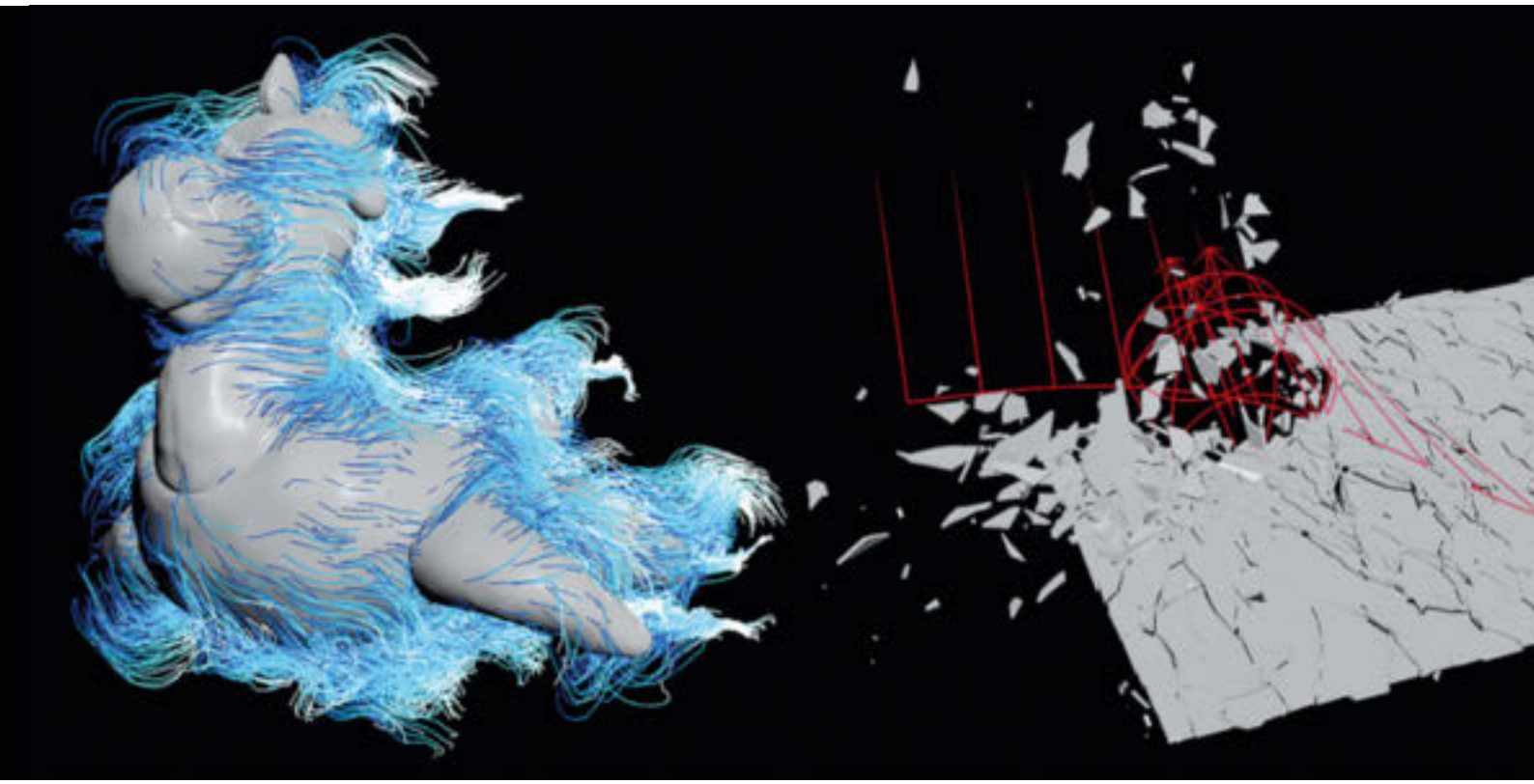


Abb. 8: Mit dem LYNX_fabric_color Node können wir auch jedes beliebige Attribut manipulieren, hier beispielsweise das Attribut für UV-Verzerrungen.

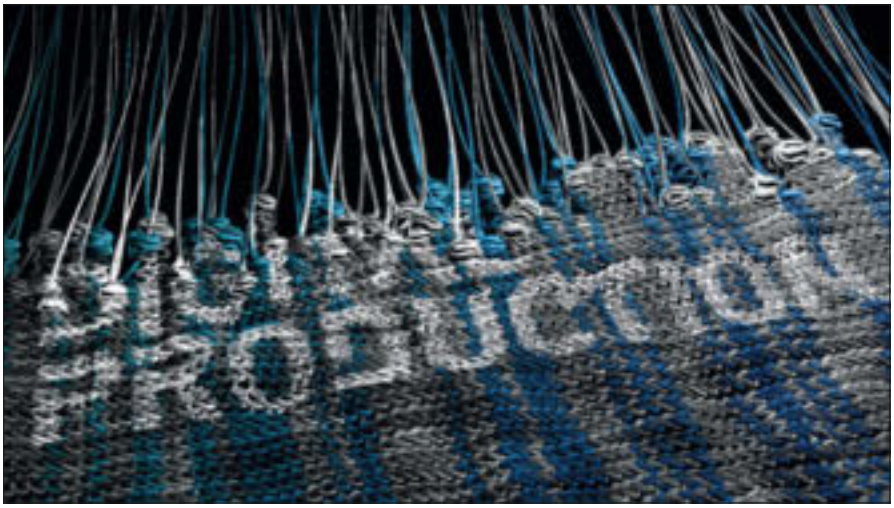


Abb. 9: Motion-Graphics-Web-Setups können durch den modularen Aufbau schnell iteriert werden (Szene steht im Github Repository zu Verfügung).

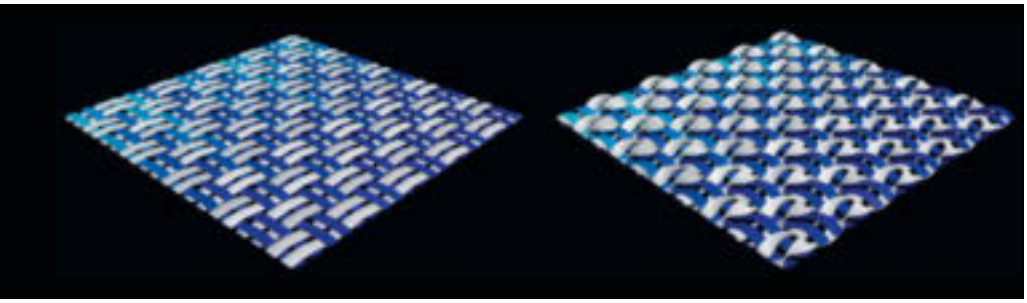


Abb. 10: Auch eigene Werte für die Attribute Width, Scale und N (Normals) werden von allen Nodes berücksichtigt.

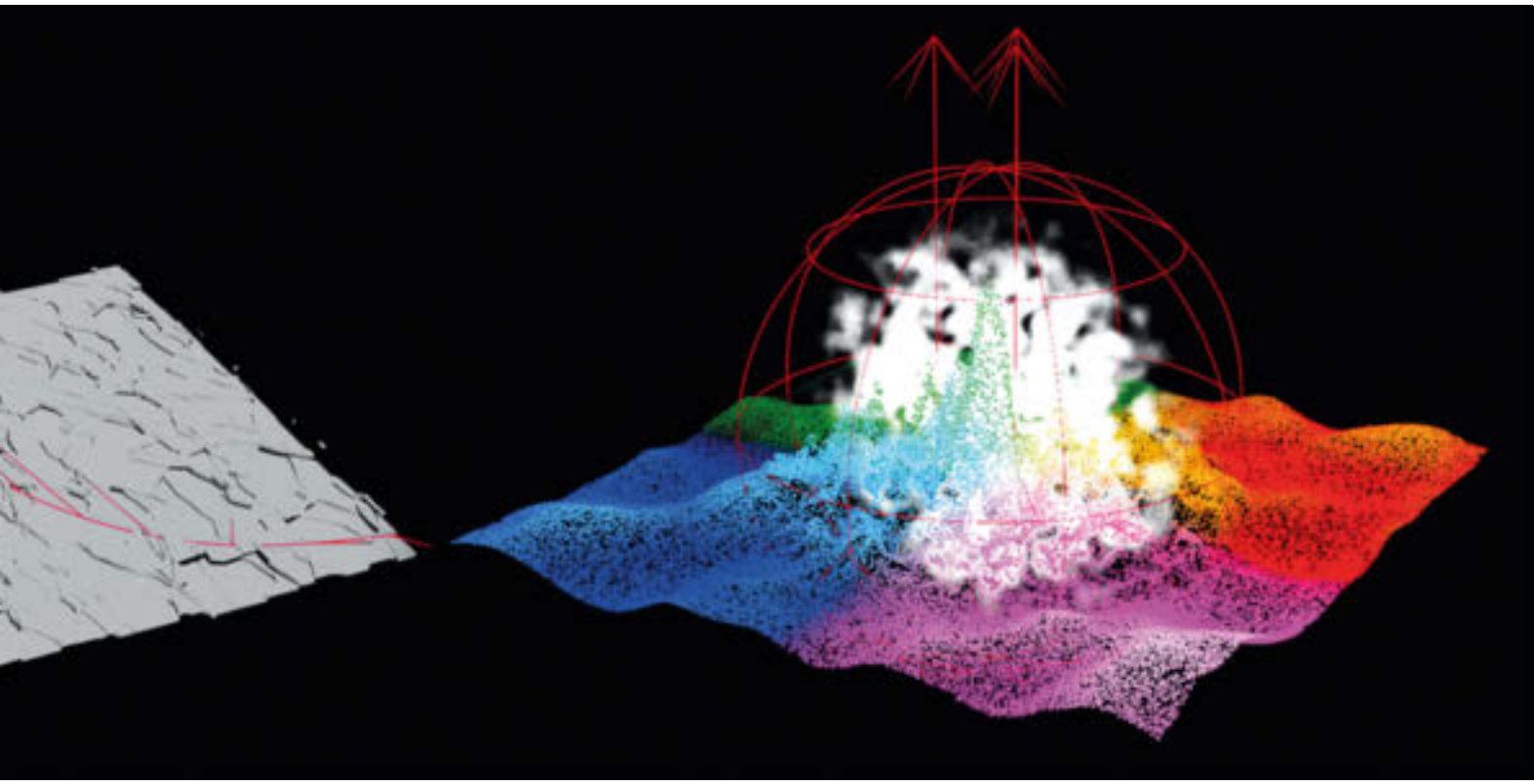
Da wir in den Node cachen, können wir eigene Cache-Lösungen mit einbinden.

Mask Your Edits

Mehrere LYNX_fabric Nodes haben auch die Möglichkeit, ihren Effekt auf Gruppen-/Attribut-/Punkt-Selektionen zu begrenzen. Wir können hier auch kleine Expressions mit eigenen Attributen oder den von LYNX erzeugten verwenden, wie zum Beispiel „@LYNX_fabric_fiber_direction==1", um nur vertikale Kurven zu manipulieren (**s. Abb. 7**).

Advanced Techniques: Adding Randomness

Altbekannte Motion-Graphics-Web-Setups lassen sich auch leicht aufsetzen, indem wir das Width-Attribut animieren und Kurvenpositionen blendshapen. Um dies zu demonstrieren, gibt es eine Beispieldatei im Github Repository. Um den Effekt visuell aufzubrechen (**s. Abb. 8**), können wir auch den LYNX_fabric_color Node zweckentfremden, indem wir statt des Cd-Attributs das UV-Attribut manipulieren. Wenn wir einen Noise Edit hinzufügen und diesen zu den bestehenden Werten addieren, erzeugen wir UV-Verzerrungen. Danach müssen wir lediglich ein Attribut kreieren und animieren, das diese neue UV nutzt und dann unser Blendshape und unsere Width steuert. Voilà, und schon haben wir ein völlig interaktives Web-Setup mit mehr Variation. Wir können auch jederzeit zwischen Pattern Presets durchwechseln, um schnell verschiedene Variationen auszuprobieren (**s. Abb. 9**).



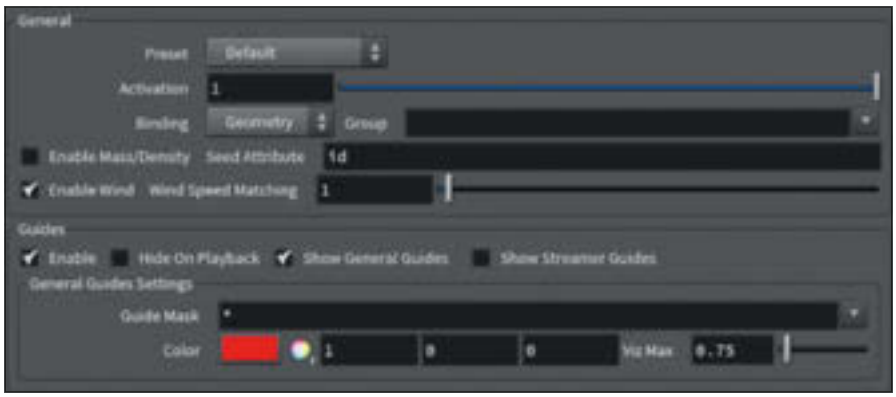


Abb. 13 : In den allgemeinen Einstellungen können wir einstellen, welche Daten wir manipulieren wollen und wie die Guides im Viewport visualisiert werden sollen.

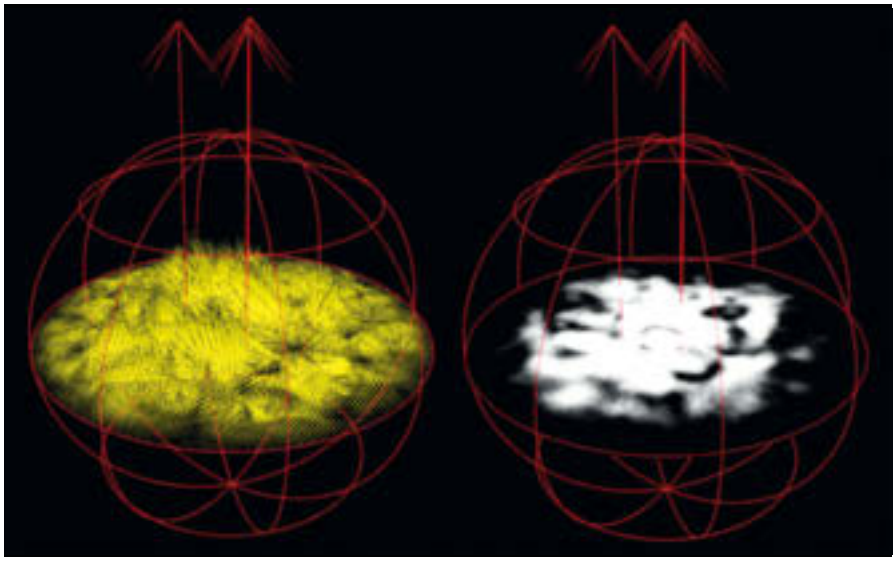


Abb. 14: Neben den General Guides bietet die Streamer (Slice-Plane) Guide-Visualisierung eine hilfreiche Vorschau. Dabei kann man zw. einer Vector- (links) und Gradient-Vorschau (rechts) auswählen.

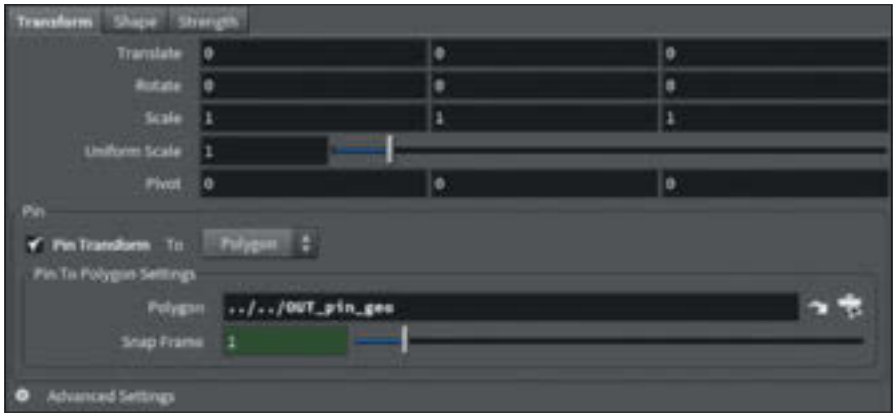


Abb. 15: Über Pin Transform kann man die Force an Kurven oder Polygongeometrien befestigen.

Custom Scaling

Die Attribute Width, Scale und N (Normals) werden von den Nodes sinnvoll verrechnet je nach Node-Funktion. Optional können wir auch eigene Werte definieren für N oder Scale, um die Kurven ungleichmäßig zu drehen/skalieren. Ein Beispiel hierfür ist ein klassisches Korbgeflecht, wo die Streben nur aus Flächen bestehen. Im LYNX_fabric_pattern Node können wir dies über eine Checkbox aktivieren und einstellen. Es empfiehlt sich, hierzu den LYNX_fabric_convert Node zu nutzen, um die Skalierungen anzuzeigen (**s. Abb. 10**).

Under the Hood

Um maximale Performance zu erreichen, werden alle Geometrien immer nur auf dem eingestellten Rest-Frame generiert. Nodes, die diesen Prozess nutzen, haben einen Rest-Frame-Parameter. Damit können selbst größere Setups noch recht performant bleiben. Sollte dein Setup doch pro Frame neu evaluiert werden, kannst du „$FF" in den jeweiligen Rest-Frame-Parametern eingeben. Somit werden alle nötigen Kalkulationen immer neu ausgeführt, sprich veränderte Polygon-Counts auf der Input-Geometrie stellen kein Problem mehr dar.

LYNX_force_general – Allzweck-Force Field the Houdini Way

Wer von anderen DCC-Applikationen kommt, der ist meist vertraut mit Force Fields, bei welchen einfach die Force transformiert und auf den Bereich maskiert, wo man sie braucht. Dies in Houdini zu machen, ist nicht so intuitiv, wie man vermuten würde, und man kann schnell verzweifeln, vor allem wenn man sich hauptsächlich um das Design der Simulation kümmern will. Genau hier soll LYNX_force_general Abhilfe schaffen. Gemacht für diejenigen, die sich mehr um den Look als um die Technik der Simulation kümmern wollen, aber dennoch die Flexibilität und Eingriffsmöglichkeiten, die Houdini einem bietet, beibehalten wollen. The „Cherry on top" ist hier, dass vor allem auf das UX-Design viel Wert gelegt wurde: Über hilfreiche Visualizer sieht man, schon bevor man den Sim-Knopf drückt, wie die Kräfte wirken. Das Force Field funktioniert auch für jeden Typ von Daten, sprich: keine getrennten Forces mehr für Geometrie und Volumes.

Es lassen sich auch die Standard-Forces, wie Gravity oder Wind hiermit problemlos eins zu eins ersetzen.

The Basics

Anders als bei vielen Houdini-Tools ist der Workflow hierbei begrenzt auf einen Node. Die Entscheidung, hier mit dem „Houdini Way of Life" zu brechen, hat einen primären Grund: User Experience/Usability, womit sich auch die Hauptzielgruppe ergibt: Motion Designers und CG-Generalists. Doch auch für erfahrene Nutzer und FX-Artists eignet es sich hervorragend, um einer Sim den letzten Feinschliff zu geben.

General Settings

Im oberen Bereich haben wir die allgemeinen Einstellungen. Ihr könnt aus einer Vielzahl an Presets auswählen, die verschiedene Use Cases zeigen. Einige dieser bilden auch die von Houdini mitgelieferten Forces eins zu eins nach. Des Weiteren können wir auswählen, auf welchen Data Container die Force wirkt (Geometrie, Volumes oder Custom Data wie z.B. Constraint-Geometrie), sprich

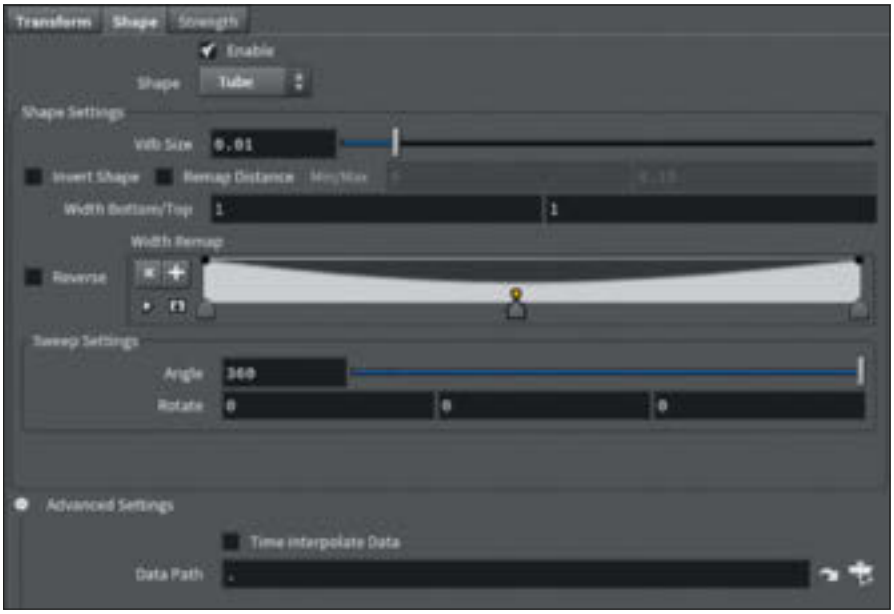


Abb. 16: Die Shape Settings definieren den Maskenbereich.

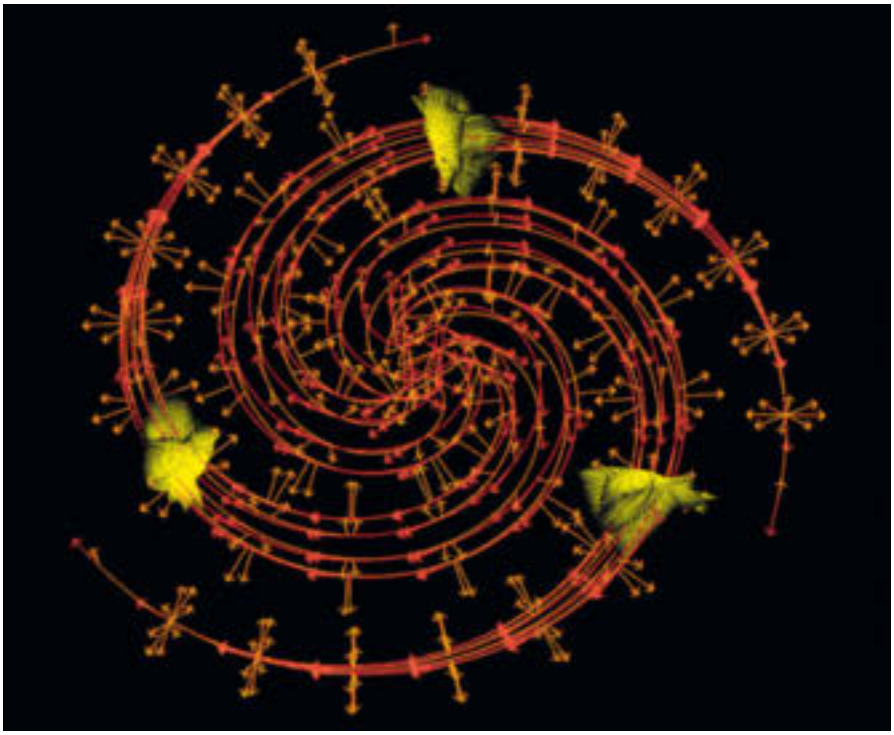


Abb. 17: Wenn man Kurven als Shape-Typ auswählt, stehen auch hier alle Guides zur Verfügung.

wir können sogar gleichzeitig beide Simulationstypen manipulieren. Auch die Guide Controls befinden sich hier. Sehr nützlich erweist sich der Knopf „Hide On Playback", der die Guides beim Abspielen verbirgt und so optimale Performance im Viewport garantiert. Sollte der Viewport vor lauter Guides zu chaotisch werden, kann man auch einzelne Guides ausblenden, hierbei wird die „*" Wildcard-Syntax unterstützt.

Transform

Der Transform hat Einfluss auf die Masken und auf die Flow Forces (**siehe Absatz Strength**). Hierzu gibt es auch einen Handle im 3D-Viewport, mit dem wir interaktiv die Force verschieben können. Des Weiteren können wir die Force an Kurven oder Polygongeometrie pinnen, somit entfällt manch nerviges Animieren vollständig.

Shape

Hier stellen wir die Maske ein. Es gibt viele Grundformen zur Auswahl, wie z.B. Sphere oder Tube, aber es besteht auch die Möglichkeit, eigene Geometrien zu nutzen. Wichtig ist hierbei, dass die Geometrie wasserdicht ist sprich ein geschlossenes Volumen hat, da intern VDBs von der Geometrie erstellt werden. Eine weitere Möglichkeit ist es auch, Kurven zu nutzen. Der Vorteil hier gegenüber der von Houdini mitgelieferten POP/Gas Curve Force ist einerseits die optimierte Performance, anderseits, dass die Force für Geometrie und Volumes funktioniert. Ein weiterer Vorteil ist, dass wir die Kurven nur als Shape-Maske verwenden können, um z.B. Noise Forces zu maskieren.

Strength

Hier finden sich alle Einstellungen über die Einflussstärke des Force Fields. Die unter dem Kateireiter „Flow" zusammengefassten Forces orientieren sich an der Ausrichtung des Transforms. Außerdem lassen sich hier auch Vex Overrides machen, mit denen sich das Force Field leicht zweckentfremden lässt, um z.B. Gruppen, Farben oder Simulationsattribute basierend auf der Maske zu manipulieren.

Next Steps: Manipulating Attributes

Nicht nur die Attribute targetv/airresist, force und v lassen sich steuern, sondern auch jedes andere Simulationsattribut wie pscale, mass, viscosity, w (Rotation bei RBD Sims). Mit ein bisschen Vex Code (**s. Abb. 19**) lässt sich jede Menge interaktiver Spaß haben, ideal für abstrakte Effekte. So lässt sich schnell der richtige Weg zu einer erfolgreichen Sim finden.

Extending The Mask

Wir können die Force auch außerhalb der Force-Maske wirken lassen. Hierzu müssen wir lediglich den Haken bei „Remap Distance" anmachen. Das intern verwendete SDF (Signed Distance Field) blendet zwischen einem Wert von 0 an der Grenze der Geometrie und -1 im Innersten der Geometrie je nachdem, wie weit wir uns in dem Geometrievolumen befinden. Dieser Wert wird dann anschließend auf die 0-bis-1-Range gemappt, um für die Falloff Ramps nützlich zu sein. Durch das SDF-Remapping verschieben wir die Range, die auf die Ramp gemappt wird. Dadurch ermöglicht sich ein erweiterter Maskenbereich. Dies ist visuell leicht zu sehen durch die Slice Plane Visualizer (**s. Abb. 21**).

Advanced Techniques: Data Sharing between LYNX_forces

Um hier auch wieder die maximale Performance herauszuholen, werden Volumes der Maskengeometrie bei allen Shapes (bis auf Kurven) angelegt, um die Maskenbereiche zu berechnen. Hierdurch können Force-Field-Berechnungen nur auf die sich

Abb. 18: Die Strength Settings bieten eine Vielfalt an Einstellungsmöglichkeiten, hier exemplarisch alle aktiviert.

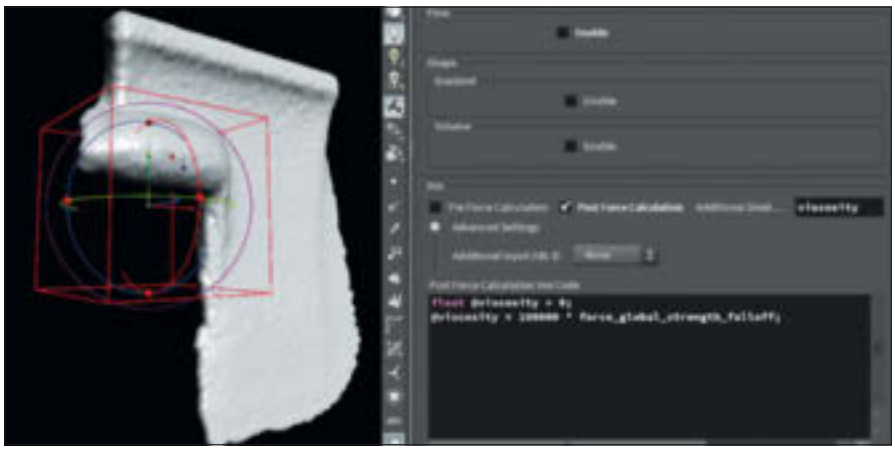


Abb. 19: Mit ein bisschen Vex Code lässt sich die Viskosität einer FLIP-Simulation interaktiv steuern.

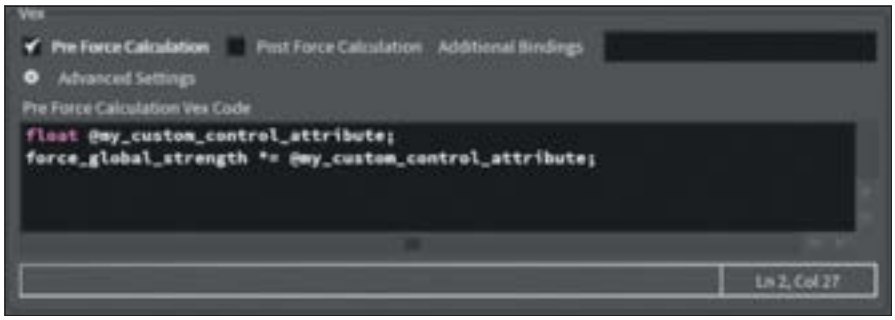


Abb. 20: Wenn wir die Force-Wirkung durch bestehende Attribute limitieren wollen, können wir mit Vex durch folgende Zeile die Stärke beeinflussen.

tatsächlich im Maskenbereich befindende Geometrie beschränkt werden bzw. sind VDB-Abfragen wesentlich schneller als reine Point Look-ups. Der einzige Nachteil dieser Methode ist, dass VDB-Daten sehr groß werden können. Um hier Abhilfe zu schaffen, können wir die VDB-Daten zwischen mehreren LYNX_forces teilen, vorausgesetzt sie besitzen die gleiche Shape. Im Shape Tab finden wir hierzu unter „Advanced Settings" den Parameter „Data Path", mit dem wir andere LYNX_force_general Nodes auswählen können. Haben wir dies gemacht, teilen sich die Forces die Daten, was auch im Viewport mit dem External Data Visualizer Text sichtbar gemacht wird.

Curve Forces

Bei Kurven bietet sich auch die Möglichkeit, ein eigenes UV-Attribut anzugeben. So können wir, wenn man Teile der Kurve löscht oder Carving nutzt (Zurechtstutzen der Kurvenenden), verhindern, dass die Radien, Falloffs und Strengths springen dank der sonst per Substep neu berechneten UV. Nebenbei noch als Quick-Tipp: Die einzelnen Kurveneinstellungen wie Radius oder Strength können auch über bestehende Attributwerte überschrieben werden, statt sie über die Ramp-Parameter zu definieren (siehe hierzu die mitgelieferte Houdini-Dokumentation).

Under the Hood

Wie schon erwähnt werden intern Volumes (VDBs) kalkuliert, die für die Force-Berechnung nötig sind. Ein weiterer wichtiger Punkt diesbezüglich ist, dass, wenn wir eigene Geometrie als Shape verwenden, die sich bewegt, die VDBs per Frame neu berechnet werden müssen. Falls die Input-Geometrie-

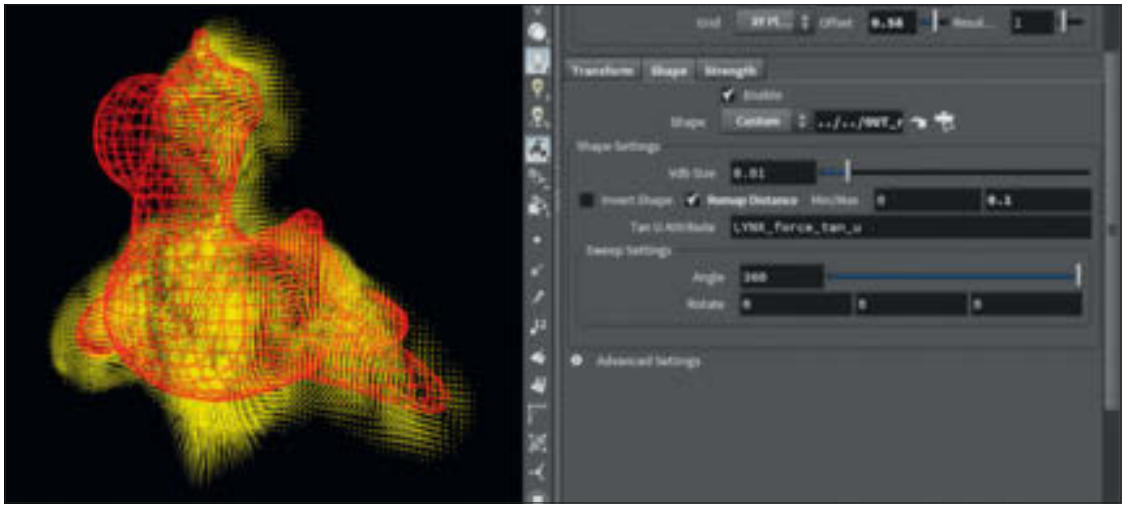


Abb. 21: Durch den Haken bei „Remap Distance" können wir den Maskenbereich auch auf außerhalb der Geometrie erweitern.

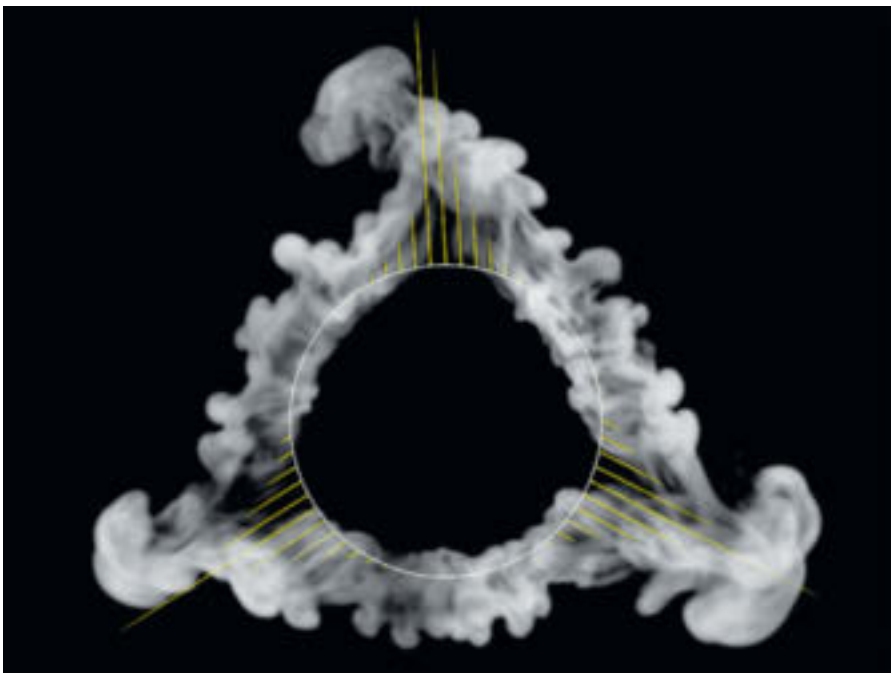


Abb. 22: Durch Edge Selections können wir gezielt eigene Velocities setzen ...

Abb. 23: ... oder auch durch Velocity Grooming.

Topologie statisch ist und nur der Transform sich verändert, empfiehlt es sich, Pinning anzumachen und die Geometrie unter Shape auf einem Frame Time zu freezen, um Re-Cooks zu vermeiden. Dadurch behält sie den Transform bei und verhindert gleichzeitig Re-Cooks.

Last but not least: LYNX_velocity

Der seit Houdini 17 mitgelieferte Velocity Node ist eine großer Sprung zum altbekannten Trail Node. Wem das an Feintuning-Möglichkeiten aber immer noch nicht genügt, ist mit dem LYNX_velocity Node bestens aufgehoben.

Der Node ermöglicht dir, sowohl Velocities über Edge Selections zu steuern, als auch Velocities zu kämmen. Da der Node einen beschaulicheren Umfang aufweist als die bisherigen, besprechen wir hier nur das wichtigste.

The Essentials

Die beiden Hauptfunktionen des Nodes lassen sich über entsprechende Haken an- und ausschalten. Velocities über Edge Selections eignen sich wunderbar, um z.B. Tendrils für kleine Wasser-Sims oder Rauchringe zu initialisieren. Hierzu wählen wir eine Edge an oder geben an, über die Checkbox „Use Boundary" alle Randkanten zu nehmen. Dann ermöglichen diverse Parameter, die Velocities entlang der Edge zu manipulieren (**s. Abb. 22**).

Wem das nicht reicht, der kann auch Velocities kämmen. Zum jetzigen Zeitpunkt ist diese Funktion beschränkt auf Stillframes, sprich wenn sich die Input-Geometrie (und deren Velocities) ändert, werden die durch den Node veränderten Attributwerte nicht per Frame mittransformiert. Sollten wir die Topologie der Input-Geometrie oder deren Velocities ändern wollen, besteht die Möglichkeit, die Groom Strokes neu zu projizieren und damit die Änderungen zu übernehmen (**s. Abb. 23**).

Wie geht's weiter?

Um es als Community-Projekt zu etablieren, lade ich euch ein, LYNX herunterzuladen, zu testen, Spaß damit zu haben und hoffentlich auch mit zu entwickeln. Die aktuelle Roadmap befindet sich auf Trello. Zu jedem Tool gibt es ausführliche Tutorials auf meinem Vimeo-Channel sowie eine Dokumentation in den Houdini-Docs bzw. auf meiner Website. Alle wichtigen Links hierzu sind über das Github Repository verlinkt. An dieser Stelle auch noch mal ein großes Dankeschön an alle, die bisher Feedback gegeben haben, vor allem Chris Kelch (**instagram.com/calicephotography** bzw. **calice-gallery.com**), Patrick Zeller, Philipp Engelhardt (**deltahoch3.com**) und Ryan Leasher (**ryanleasher.com**). **› ei**

Derzeit in München als FX (Pipeline) TD tätig, ist Luca Scheller ein großer Fan von Open Source Software und liebt es, Wissen mit der Community zu teilen und Prozesse zu verbessern, sei es durch effektiveres Management oder neue VFX-Workflows.

LYNX GitHub

LYNX Demo

Links

Download
▷ **github.com/LucaScheller/VFX-LYNX**

Vimeo (Tutorials und Showcases)
▷ **vimeo.com/lucascheller**

Website
▷ **lucascheller.de**