

Abschlussausarbeitung

Aus Fest wird Flüssig: Simulation von schmelzendem Eis

In der Veranstaltung

Datenverarbeitung in der Medienproduktion

Wintersemester 2019/2020

Ausgearbeitet von

Florian Kudirka

Florian Pfützenreuter

Markus Schürmann

Dieses Dokument enthält Bilder, die nicht auf Urheber und deren Rechte geprüft wurden.

Daher bitte nicht außerhalb der Hochschule Furtwangen publizieren.

Inhalt

1. Einleitung	4
Ideenfindung	4
2. Projektvorstellung	5
Von Schnee zu Eis	5
3. Implementierung und Prozesse	6
Erste Schritte	6
Prototypen	6
4. Beschreibung des Algorithmus	8
5. Auftretende Probleme	10
Probleme auf dem Mac	10
Probleme auf Windows Rechnern	11
Houdinis Dokumentation	11
6. Fazit	11

1. Einleitung

Im Rahmen der Veranstaltung Datenverarbeitung in der Medienproduktion beschäftigten sich die Projektmitglieder „Aus Fest wird Flüssig: Simulation von schmelzendem Eis“ mit der Simulation eines Eisblocks, der über die Zeit schmilzt.

Ideenfindung

In der ersten Phase des Projekts haben wir uns mit der Ideenfindung beschäftigt. Daraus ergaben sich folgende weitere Projektideen:

- Erzeugung eines **prozeduralen Labyrinths**. Bei dieser Idee sollte man dynamisch und zufällig ein Labyrinth erzeugen können. Mithilfe von Inputboxen und Slidern, sollte der User Einfluss über Detailgrad, Dichte und Verzweigung des Labyrinths nehmen können.
- Bildung eines **Efeus**, der sich um einen **Baum windet**. Hier sollte es möglich sein, einen Efeu an einem Baum entlang wachsen zu lassen. Mithilfe eines Algorithmus sollte sich der Efeu selbst einen Weg um den Baum suchen. Hier war außerdem angedacht, wie bei dem Labyrinth, mit Hilfe von diversen Slidern die Größe und Dichte der einzelnen Blätter zu bestimmen.

Des Weiteren gab es weitere Ideen, die aber in früheren Stadien des Ideenfindungsprozesses verworfen wurden.

Das Eisschmelzprojekt entpuppte sich als präferierte Wahl, da es den nötigen Spagat zwischen bekannten Techniken und neuen Herausforderungen bot.

2. Projektvorstellung

Das Projekt beschäftigt sich, wie schon in der Einleitung beschrieben, mit der Simulation eines schmelzenden Eisblocks über die Zeit. Vorbild hierfür war ein quadratischer Eisblock (Abbildung unten).



Abbildung: Vorbild für das Projekt

Von Schnee zu Eis

Im Anfangsstadium der Projektkonkretisierung führen wir zeitweise zweigleisig. Die erste Idee setzte sich mit dem Fall eines Schneeblocks auf ein festes Objekt auseinander. Der Block sollte sich daraufhin in einzelne Teile auflösen.

Die zweite Idee setzte auf die oben skizzierte Eisblockschmelze. Dabei ließen wir uns sowohl bei der ersten, als auch bei der zweiten Idee von Tutorials inspirieren und bildeten unsere Entscheidung inkrementell und iterativ durch die gewonnenen Erkenntnisse.

3. Implementierung und Prozesse

Erste Schritte

Nach Einigung auf das Projektthema, begannen wir zuerst verschiedene Lösungsansätze zu eruieren. Prinzipiell gab es hierfür drei Ansätze.

Der erste Ansatz nutzte die Houdini eigene Skriptsprache HScript. Als Vorteil ist hier die enge Integration mit dem Node basierten System von Houdini zu erwähnen. Variablen für z.B. Punktpositionen können sowohl in einer Node als auch in HScript leicht verwendet werden.

Der zweite Ansatz setzte auf die Python Implementierung in Houdini. Da wir als Medieninformatiker schon viel Praxis im Bereich der Programmierung haben, fiel es uns einfacher mit Python zu arbeiten, als mit HScript. Mit Hilfe von Herrn Duda war die Einarbeitung in die Houdini spezifischen Klassen deutlich einfacher.

Der dritte Ansatz war die klassische Implementierung mit dem Node basierten System von Houdini. Dieser Ansatz wählten wir schlussendlich für unser Projekt. Bei den Node basierten Techniken, wird mithilfe visueller Elemente programmiert.

Prototypen

Einer der Prototypen verfolgte die Python-Implementierung. Ziel war es mit wenigen Nodes und vielen Python-Bausteinen, die einzelnen Komponenten (z.B. Punkte innerhalb eines Würfels) zu erzeugen und verschwinden zu lassen. Dieser Ansatz erschien uns anfangs am schlüssigsten, da wir noch keine Erfahrung in der Node basierten Programmierung hatten. Als wir nach und nach die Techniken von Houdini lernten, stellte sich jedoch diese Implementierung als zu kompliziert heraus. Deshalb verfolgten wir diesen Ansatz nicht mehr weiter.

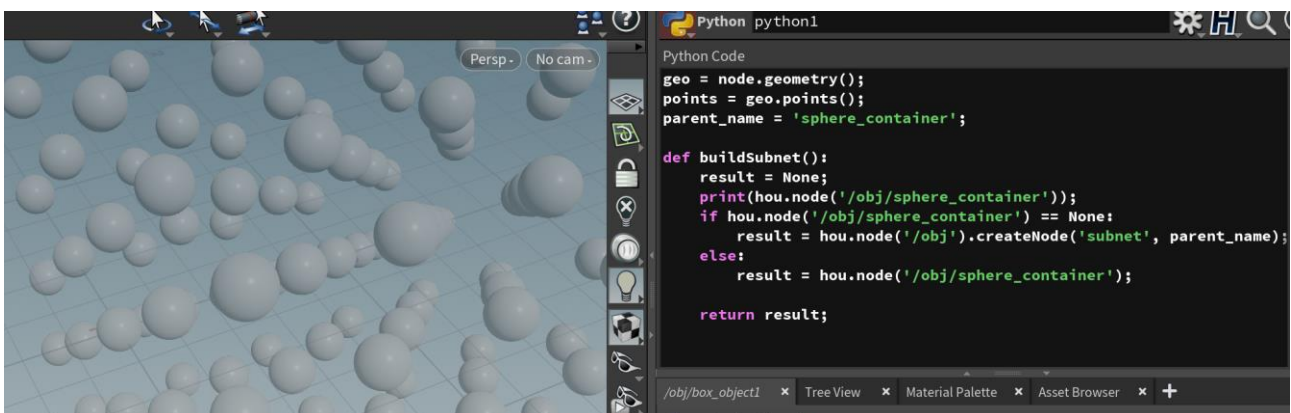


Abbildung: Links: Ansicht eines Würfels mit Unterpunkten Rechts: Python-Code

Ein weiterer Prototyp setzte sich mit der auf [Seite 5](#) skizzierten Idee auseinander, dass ein Schneeblock auf ein festes Objekt fällt und der Block sich daraufhin in einzelne Teile

auf löst. Dieser Prototyp wurde schon mit den Node basierten Techniken von Houdini erzeugt. Da jedoch unsere finale Idee der schmelzende Eisblock ist, fand dieser Prototyp bei uns ebenfalls keine Anwendung in der Endabgabe.

Danach erstellten wir einen weiteren Prototyp, welcher sich mit dem Erzeugen von Schnee befasste. Hierbei wurde eine partikelbasierte Lösung gewählt, in der die Partikel auf einer Fläche kleben bleiben und falls die Oberfläche zu steil ist, davon abrutschen.

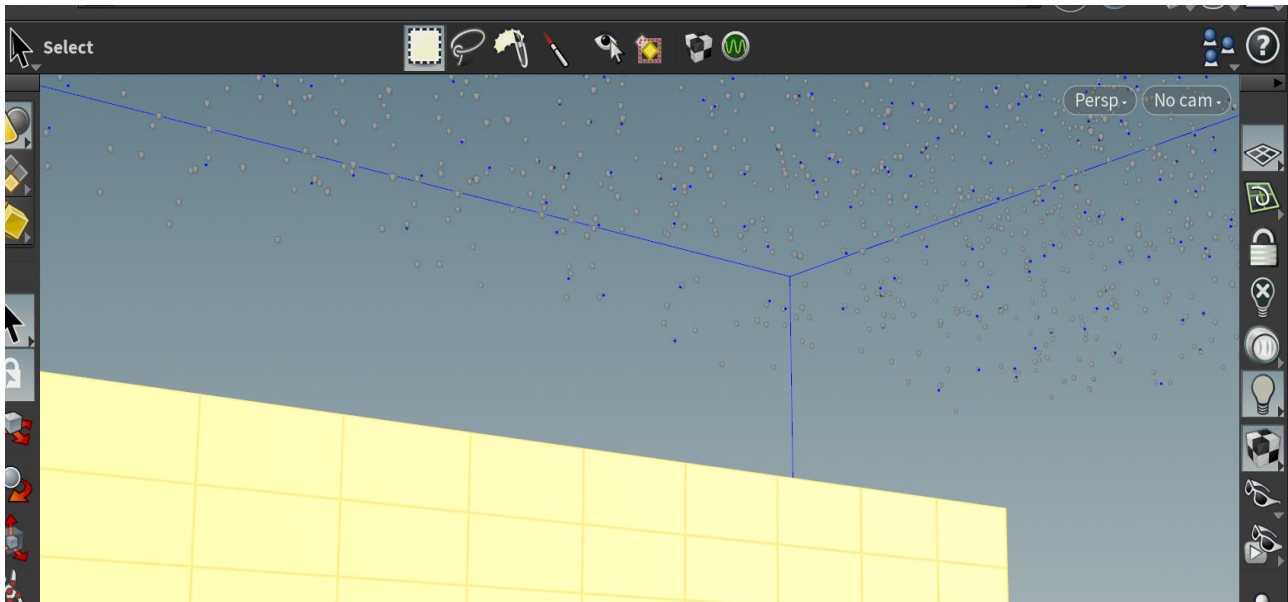


Abbildung: Graue und Blaue Partikel, die den Schnee simulieren.

Diesen Prototyp haben wir auch anhand eines Tutorials entwickelt.

4. Beschreibung des Algorithmus

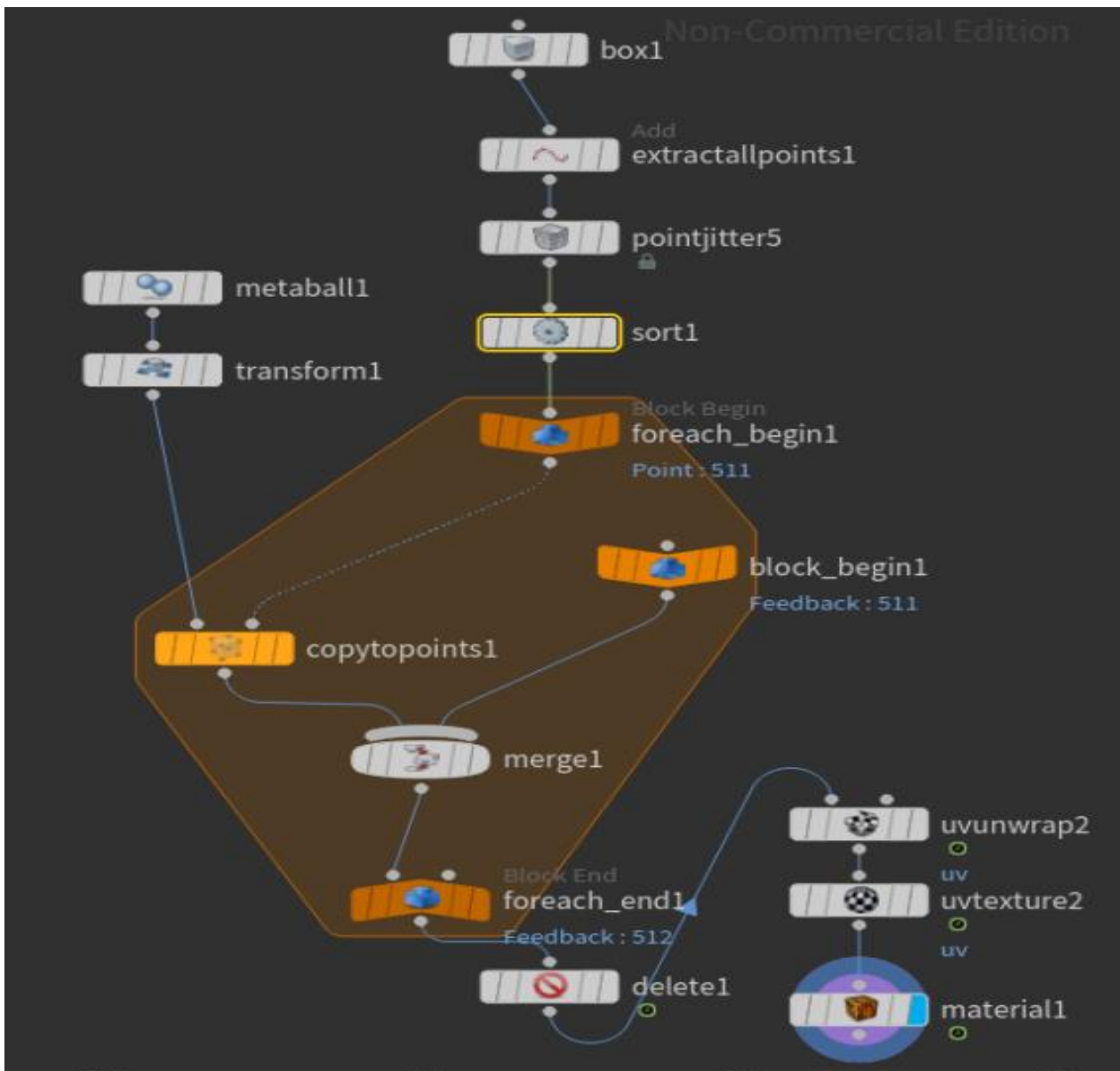


Abbildung: Kompletter Node-Tree

Als Grundgerüst für unseren Eisblock nahmen wir einen Würfel, da dieser schon die Grundform eines Eisblocks aufweist. Auch kann statt einem Würfel auch eine andere Figur als Grundform dienen.

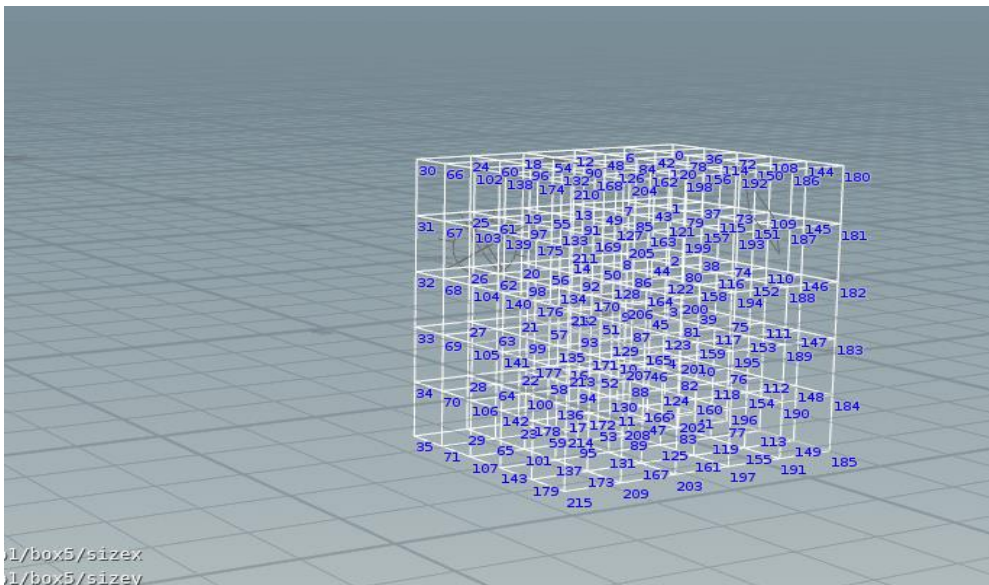


Abbildung: Würfelarundaerüst und Punkte

Getestet haben wir das Ganze mit dem Schweinekopf und einem Torus. Das Mesh muss hierbei viele Punkte aufweisen, die möglichst naheliegen.

Die Size und die Divisionanzahl sind vom Nutzer freiwählbar. Durch die Anpassung der Divisions, entstehen mehr oder weniger Punkte. Danach ziehen wir die Punktinformationen per „extractallpoints“-Node heraus. Darauf folgend setzen wir auf die Punkte einen „pointjitter“, damit die Punkte nicht alle komplett geordnet angezeigt werden. Der Nutzer hat hierdurch weitere Anpassungsmöglichkeiten, so kann nicht nur ein Eisblock entstehen, sondern unterschiedliche Formen, wie Schneebälle. Nach dem „pointjitter“-Node, setzen wir die „sort“-Node zum Sortieren der Punkte ein, damit im späteren Verlauf der Anwendung, der Würfel korrekt von den äußeren Eckpunkten, zum Inneren abschmilzt. Danach bilden wir eine ForEach-Schleife, welche über jeden Punkt läuft. Zuerst wird in einer „Attribute-Expression“-Node dafür gesorgt, dass die Punkte zufällige Werte bekommen, damit später alle Löcher gefüllt sind im Eisblock. Außerhalb der ForEach-Schleife wird ein Metaball erstellt, welcher über die „transform“-Node in Abhängigkeit von den „size“-Attributen des Würfels durch die „division“-Attribute des Würfels skaliert werden. So wird sichergestellt, dass bei einer höheren Size und niedrigerer Divisionanzahl, die Metabälle immer groß genug skaliert sind, um sich zu einem Volumen zu verbinden. In der „copyToPoints“-Node werden die einzelnen Metabälle an die Punktpositionen der ForEach-Schleife gesetzt und mit der Merge-Node zusammengefügt. Hierbei muss eine BlockBegin-Node eingefügt und in den Merge einbezogen werden, damit das Ergebnis in der ForEachEnd-Node miteinbezogen werden kann.

Hiernach fügten wir eine Delete-Node ein, welche per "Delete by Range"-Einstellung über alle Punkte iteriert und vom 0ten Punkt bis zum Punkt in Abhängigkeit des jetzigen Frames geht.

Dabei werden 4 von 4 Punkten gelöscht pro Frame. Zum Schluss setzten wir per UVUnwrap, UVTexture- und Material-Node einen Shader und Textur auf den Würfel, sodass er wie Eis aussieht.

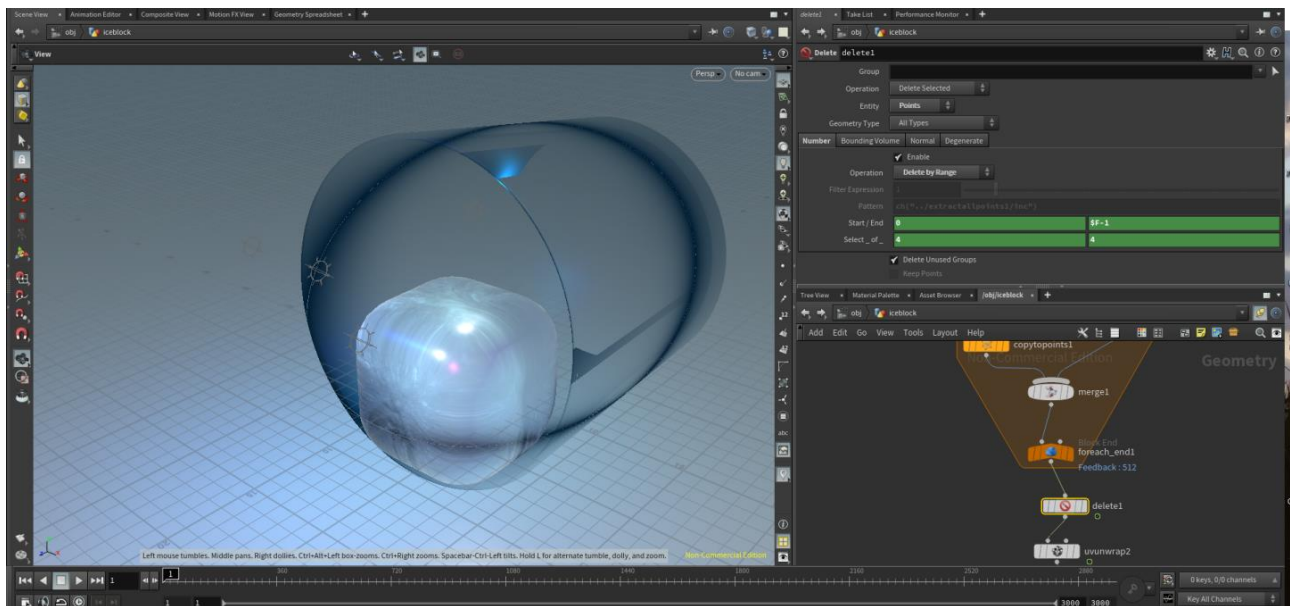


Abbildung: Aufbau der Delete-Node

5. Auftretende Probleme

Probleme auf dem Mac

Florian Kudirka verwendete Houdini auf dem Mac. Hierbei gab es einige Probleme mit Houdini. Das schwerwiegendste Problem, war ein Problem mit dem Houdini Lizenzserver. Die Problemfindung und Lösung gestaltete sich als Herausforderung. In Zusammenarbeit mit Lutz Weigold, konnten die Probleme jedoch gelöst werden. Auslöser hierfür war, dass sich der Hostname des Macs im Uni-WLAN verändert. Der neue Hostname wurde jedoch nicht in den Lizenzserver übernommen, sodass er die vorhandenen Lizenzen nicht finden konnte. Die Lösung bestand darin, den Houdini-Servernamen auf den neuen Hostnamen des Rechners zu setzen.

Probleme auf Windows Rechnern

Auf Windows gab es bedeutend weniger Probleme. Hier waren hauptsächlich Abstürze des Programms, insbesondere beim Laden von aufwendigen Geometrien, zu verzeichnen. Des Weiteren gab es gelegentlich komplette Systemabstürze.

Houdinis Dokumentation

Als weiteres Problem möchten wir die Dokumentation¹ von Houdini anführen.

Die Dokumentation ist unübersichtlich gestaltet, es bedarf zum Teil längerer Recherche bis man zum gewünschten Eintrag bzw. Ergebnis kommt. Einhergehend sind leider oft die Erklärungen zu kurz für Studierende, die Houdini zum ersten Mal bedienen. Es gibt sehr oft keine Beispiele zur gezeigten Node bzw. Methode. Dies macht das Nachvollziehen und Anwenden der Funktionsweise sehr schwierig.

6. Fazit

Nach dem endgültigen Ergebnis lässt sich sagen, dass sich Houdini für die Durchführung einer Eis Simulation zwar eignet, aber durch seine schwer zu verstehende Dokumentation, das Fehlen von weiterführenden Debug Tools abseits der Bypass-, Active Flags und der häufig undeutlichen oder unzureichenden Fehler Codes, stets nicht eingängig genug für das Projekt war. Die Einarbeitung in Houdini benötigte einen Großteil der Projektarbeit und stellte dadurch ein schlichteres Ergebnis dar, als mit mehr Zeit, KnowHow und einer einsteigerfreundlichen Dokumentation möglich gewesen wäre, sowohl von Seite der Simulation, wie auch von Visueller Seite. Probleme mit dem Render-system, Abstürzen und Instabilitäten haben neben Zeitmangel, das visuelle Endergebnis beeinträchtigt gegenüber der Anfangsvision. Einen häufigen Wechsel der Lösungsansätze hat insbesondere den Fortschritt häufig zurückgesetzt, diese Vorgehensweise hat uns aber auch Houdinis individuelle Pipelines nähergebracht. Und auch wenn uns die Zusammenhänge der Networks immer noch nicht hundert prozentig klargeworden sind, war es dennoch interessant, zu beobachten wie eine Lösung für die Eisschmelze in einem DOP Network, entgegen einem reinen SOP Network aussehen würde. Auch die Lösungsansätze mittels der programmierten Python Nodes waren aufschlussreich, wenn

¹ Zu finden unter <https://www.sidefx.com/docs/houdini/>

auch hier die Dokumentation etwas zu kurz kam und die Arbeit letzten Endes mit normalen SOP Nodes weitergeführt wurde. Durch diese Erkenntnisse können wir das Potenzial von Houdini erkennen, letztlich auch z.B. die Attribute im Geometry Spreadsheet besser steuern, um eine Schmelze zu simulieren. Da wir im Nachhinein erkannt haben, dass das Lösen der Problemstellung unser gewähltes Thema schwieriger war als zuerst angenommen, ist das Ergebnis für uns zufriedenstellend.