## 1. Treffen 22.03.2021

Beim ersten Treffen wurde das Team in eine Realisierungsgruppe und eine Algorithmengruppe eingeteilt.

Realisierung:

Joseph Heetel, Bernhard Strobl, Maurice Ach, Marco Vogt

Algorithmen:

Tobias Eder, Marvin Potempa, Maximilian Thiel, Ahmed Salame

Realisierungsgruppe hat 5 Alternativen vorgestellt:

1. ReSTIR - Vulkan Projekt

2. Nvidia Projekt aka Bachelor Arbeit von Adrian Huber

3. Sascha Williams Projekt

4. Kronos Group Projekt

5. All from Scratch

Algorithmen:

Hat verschiedenen Algorithmen vorgestellt und erklärt in welchen Teil der Pipeline diese angewendet werden.

## 2. Treffen 29.03.2021

Es wurde sind einstimmig dafür entschieden, dass wir als Projektgrundlage die Beispielprojekte von Sascha Williams nehmen, anstatt die NV Samples der Bachelorarbeit.

Es wurden 3 Teams gebildet. Die Realisierungsgruppe schaut sich die Beispielprojekte an und sortiert den Basiscode. Die zweite Gruppe schaut sich Machine Learning Algorithmen an. Die dritte Gruppe sucht nach Algorithmen ohne Machine Learning. Die Gruppe 1 und 3 werden nach einer Woche zusammenarbeiten.

1. Gruppe:

Joseph Heetel, Bernhard Strobl, Maurice Ach

1. Gruppe:

Marvin Potempa, Tobias Eder

1. Gruppe

Maximilian Thiel, Ahmed Salame

## 3. Treffen 12.04.2021

Die Gruppen berichten von Ihrem Fortschritt. Marvin aus der Machine Learning Gruppe hat sich in verschiedene Theorien und Machine Learning Paper eingelesen. Tobias hat sich mit der Praxis beschäftigt. Es wurde auch versucht ein Prototyp anzufertigen. Tobias und Marvin stellten Ihre Erkenntnisse vor.

Die Realisierungsgruppe hat ein Projekt angelegt, dass beim Kompilieren ein leeres Fenster ausgibt. Dafür wurde die CMake-Liste angepasst. Außerdem wurde die Begriffe aus der Realisierungsgruppe durchgearbeitet und vorgestellt.

Der Algorithmusgruppe hat sich die Monte Carlo Methode angeschaut. Diese wurde vorgestellt. Ein Projekt „RayTracing in one Weeking“ wurde vorgestellt. Außerdem wurde ein Überblick über alle Filterverfahren gegeben.

Bis nächste Woche soll sich das Team zusammensetzen und eine Pipeline aufstellen. Dafür soll sich ein komplette Programmaufgestellt werden, wo die Monte Carlo Abschätzung, BRDF, Deep Learning, Filter, und alle weiteren eingebaut werden. Nicht nur aus Realisierungssicht, sondern auch aus Mathematisch, Algorithmischer Sicht. Diese Pipeline soll auch als Grafik angefertigt werden.

Es wurden drei Gruppe gebildet. Die sich vorab mit der Pipeline beschäftig und sich dann wird sich in einer Großen Gruppe getroffen. Das einzelnen Teilnehmer sollen sich nun auch auf ein Thema festlegen.

1. Prepass:

Joseph Heetel

1. RT und Monte Carlo:

Ahmed Salame und Bernhard Strobel

1. Filter:

Marvin Potempa, Tobias Eder, Maximilian Thiel, Maurice Ach

## 4. Treffen 19.04.2021

Die Gruppen berichten das Sie in der Woche erledigt haben. Es wurde eine Pipeline aufgestellt und die Themen wurden vorläufig eingeteilt. Außerdem wurden grundlegende Bausteine implementiert. SVGF und BMFR wurden nochmal erklärt. Es wurde angemerkt das die BRDF Github Projekt auch als Code Basis benutzt werden kann. Die Themen wurden folgendermaßen aufgeteilt:

1. Machine Learning:

Tobias Eder

1. Prepass + Raytracing + BRDF + Monte Carlo: Entwicklung eine Path Tracers mit BRDF und Monte Carlo Funktion und Global Illumination

Ahmed Salame und Bernhard Strobel

1. Spatio-Temportal Reprojection mit Schnittstellen:

Maximilian Thiel

1. Filter für welche Verfahren brauchen wir welche Filter? Welche Räumlichen Filter und welche Zeitlichen Filter brauchen wir?:

Marvin Potempa

1. SVGF:

Maurice Ach

1. BMFR:

Joseph Heetel

**Maximilian Thiel**: Spatiotemporal Reprojection / Temporal Accumulation / History Buffer

**Tobias Eder**: Machine Learning

**Bernhard Strobel**: Prepass / Architektur / (RT, Monte Carlo)

**Ahmed Salame**: RT / Monte Carlo / BRDF

**Maurice Ach**: SVGF

**Joseph Heetel**: BMFR

**Marvin Potempa**: Filtertechniken

ALLE NAMEN SIND NOCH VORLÄUFIG!

Nun gilt Gliederung und Literatur mit dem Dozenten abzusprechen.

## 5. Treffen 26.04.2021

Die abgegebenen Ausarbeitungen mit Quellen und Inhaltsverzeichnis wurden besprochen und der Dozent gab kurzes Feedback. Eine einfache Implementierung von Raytracing und Rasterisierung wurde auf dem OPS aufgesetzt. Da das OPS eine temporales Lizenzierungsproblem hat wurde sich darauf geeinigt das die eigenen Branches schnell auf den Master gepushed werden.

## 6. Treffen 03.05.2021

Die Gruppe berichtet von Ihrem Fortschritt.

Tobias Eder hat sich mit der Intel Open Image KI beschäftig und hat hierbei eine Demo gezeigt. Diese nimmt Image Dateien und gibt dann ein rauschfreies Bild aus. Der Code soll so umgeändert werden, dass er die Eingabe direkt aus dem Framebuffer kommt.

Maurice Ach seine Ausarbeitung vom SVGF Filter vorangeführt.

Ahmed Salame hat weiter an seiner Ausarbeitung gearbeitet. Außerdem wurde an der Implementierung des Path Tracers gearbeitet.

Joseph Heetel hat mit der Schriftlichen Ausarbeitung begonnen und seine Quellen sortiert.

Maximilian Thiel hat weiter wissenschaftliche Paper gelesen und die Unterschiedlichen Ansätze von dem Papern in einer Ausarbeitung aufgeschrieben und sich einen Überblick geschaffen.

Bernhard Strobl hat eine Grafik erstellt, wie der Raytracing Output und der Pre Path aufgebaut wird. Dies soll nun von jeder Person mit ihrem Input die Sie für Ihren Filter brauchen ergänzt werden.

Jede einzelne Person sollte sich anschauen was sie für die einzelnen Filter brauchen. Am ende dieser Woche soll ein großes Dokument entstehen, was alle wichtigen Komponenten enthält, die wir brauchen werden.

## 7. Treffen 10.05.2021

Der Gitlab Server erlaubt wegen Lizenzproblemen nun keine Pushes oder Commits mehr. Deswegen muss nun in eigenen Branches gearbeitet werden. Die Hochschule versucht dieses Problem zu lösen. Es würde sich überlegt auf Github zu wechseln. Dafür wurde sich das Pricing Modell von Github angesehen. Es wurde sich vorläufig dazu entschieden das Projekt samt Historie auf ein Github Repository umzuziehen. Das Wiki bleibt hierbei aber unverändert.

Maurice Ach hat sich diese Woche um die Algorithmen im SVGF gekümmert. Hierbei wurde einige Projekte die SVGF implementiert haben angeschaut. Allerdings läuft der Code noch nicht. Diese wäre hilfreich zum Verstehen der Algorithmen. Die Buffer sollen dann mit Renderdoc ausgelesen werden. Dies würde dem Leser beim Verständnis helfen.

Joseph Heetel hat sich die Implementierung weiter angeschaut und die Implementierung der ursprünglichen BMFR Leute sich erarbeitet. Das Mathematische Verfahrung wurde sich erarbeitet und wird dann in die Arbeit eingebaut. Die Idee die ganze Software auf der CPU zu implantieren hat bis jetzt nicht funktioniert.

Marvin Potempa hat sich die Aufteilung seine Arbeit überarbeitet. Er meldet bedenken, dass dies zu wenig Inhalt für die Ausarbeitung ergibt. Er meldet sich im weiteren verlauf bei dem betreuenden Dozenten.

Maximilian Thiel für die zeitliche Akkumulation der Filter mit den besprechenden Personen zusammenarbeiten muss. Max hat seinen Text angefangen. Außerdem wurde ein SVGF Beispiel heruntergeladen und der Beispielcode durchgegangen.

Ahmed Salame hat mit der Implementierung und der Ausarbeitung angefangen. Durch Krankheit wird er jedoch erst diese Woche damit anfangen.

Tobias Eder mit dem Intel Denoiser weitergearbeitet. Dieses wurde so umgebaut das es mit dem Projekt funktionieren könnte. Außerdem hat er mit Nvidia OptiX gearbeitet. Außerdem wurde die Ausarbeitung angefangen. Den Dozent interessiert wie die trainierten Netze von Nvidia aufgebaut sind.

Bernhard Strobel hat den bisherigen Code angepasst damit dieser auch ohne Flags funktioniert. Außerdem wurde die Nvidia Szene der Bachelorarbeit eingebaut. Leider funktioniert diese nur beim Raytracing. Da die Texturen einzeln reingeladen werden. Es wurde der Vorschlag gestellt das man sich auf wenige Szenen beschränkt. Für die Szenen wurde sich auf die Nvidia Cornell Box und Sponza geeinigt.

Das Repository auf Github wird nun aufgesetzt.

## 8. Treffen 17.05.2021

Der Code kann nun auch die benötigten Szenen laden. Am Raytracer muss allerdings noch viel angepasst werden. Der Code soll noch um einen Filter Renderpass erweitert werden. Außerdem müssen die Buffer implementiert werden. Die Hochschulrechner sind Softwareseite nicht auf den neusten Stand. Diese müsste für ein besseres zusammenarbeiten aktualisiert werden. Die anderen Teammitglieder sollen sich den Code auch anschauen damit diese wissen, wo sie ihren Part implementieren müssen. Der Ray Tracing Part steht so weit, dass die Grundlagen implementiert wurde. Die Monte Carlo Funktion funktioniert jedoch noch nicht.

Tobias Eder ist mit der Arbeit am Intel desnoiser durch. Nun arbeitet er an OptiX und probiert ob dies in unser Projekt eingebaut werden kann.

Marvin Potempa: Hat an seiner Ausarbeitung weiter gemacht. Hat sich die anderen Arbeiten anschaut, um seine zu erweitern.

Maurice Ach hat seine Ausarbeitung weiterbearbeitet

Die Basis ist so weit gut gelungen nun muss mit den Monte Carlo Funktion im Ray Tracer erweitert werden.

## 9. Treffen 31.05.2021

Josef Heetel berichtet, das Projekt nun einen funktionierenden G Buffer Pass und ein Motion Vector System hat. Mesh IDs wurden noch nicht eingebaut. Außerdem wurde ein Filterpass eingebaut, dieser ist bis jetzt aber noch nicht funktional.

Ahmed Salame hat an dem Path Tracer gearbeitet. Dieser ist so weit implementiert und funktioniert auch. Bei der übergabe der Gltf fehlt nur noch das die Materialen übergeben werden. Als andere wird schon richtig übergeben. Der Code funktioniert, soll aber noch aufgeräumt werden. Bernhard wird Ahmed dabei helfen und über den jetzigen Code einmal drüber sehen.

Bernhard Strobel versucht den GUI Renderpass von den Rest abzutrennen, da dieser im konflikt zu den anderen Renderpässen von Sascha Willams steht. Als nächstes will er die Command Buffer bearbeiten, sodass diese parallel laufen können. Dies geschieht in Zusammenarbeit mit Ahmed.

Tobias Eder hat ein convolutional neural network aufgebaut. Die Grundidee dabei ist, dass Tobias sich sein eigenes neuronales Netzwerk aufbaut. Dieser kann mit einen Datensatz lernen und kann richtige vorhersagen machen. Dieser Woche möchte er sich mit dem Autoencoder beschäftigen und den besten Aufbau suchen. Für das Training wurden 10000 Bilder aus der Sponza Szene gerendert.

Marvin Potempa hat den anderen Mitglieder bei der Implementierung geholfen. Bei der A-Trous wurden GLSL Beispielimplementierungen gesucht. Bei der Ausarbeitung hat den groben Teil geschaffen wird aber noch um einige Sachen ergänzt. Marvin wird in seiner Ausarbeitung diverse Begriffe erklären und hat sich Gedanken zu der bevorstehenden bevorstehenden Präsentation gemacht und würde das Video auch bearbeiten und zusammenschneiden.

Maximilian Thiel äußert bedenken das seine schriftliche Ausarbeitung zu kurz werden könnte. Herr Dreier könnte sich vorstellen das Bilder von Spatial Temportal Accumalation gemacht werden können und in seiner Ausarbeitung erklärt werden können. Diese Woche soll die Temportal Accumolation im Code umgesetzt werden sobald der Path Tracer die richtigen Bilder darstellen kann.

Maurice Ach hat sich versucht eine SVGF Projekt von GitHub zum Laufen zu bringen. Leider blieben die Versuche ohne Erfolg. Er versucht nun das Beispiel von TheVaffel aufzusetzen und implementieren. Herr Dreier stellt dafür die Inputbilder bereit. Um das Projekt zum Laufen zu bekommen wurde Linux auf dem Rechner installiert. Außerdem wurde ein Dokument mit Fachbegriffen wurde eingeführt.

Der Dozent schlägt bei der Präsentation vor, dass das Projekt zusammen vorgestellt wird, am besten in der Form eines Videos. Danach sollen die einzelnen Mitglieder sagen wer welchen Anteil an dem Projekt hatte. Zeitlich könnte es jedoch schwierig werden alle miteinander kombiniert in vier Wochen am Laufen zu haben. Dies ist jedoch nicht schlimm. Für die Demo wäre es auch nicht schlimm, wenn es mehrere Executables gibt die vorgestellt werden.

Der Dozent würde sich wünschen, dass Algorithmen oder einzelne Schritte mithilfe von Bildern erklärt werden können. Diese können als veranschaulichen für die schriftliche Ausarbeitung dienen.

Der Dozent würde sich wünschen, dass ein Dokument angefertigt wird, auf die Schnittstellen definiert werden. Damit soll die Variablen erklärt werden. Dieses Dokument wurde von Josef Heetel eingerichtet.

## 10. Treffen 07.06.2021

Bei der Evaluation waren die Studenten zufrieden und fanden das Thema interessant, aber auch anspruchsvoll.

Ahmed Salame hat diese Woche an dem Path Tracer gearbeitet. Dieser gibt jetzt die Albedo Werte zurück. Das Bild ist aktuell, aber noch nicht verrauscht. Die Reflektionseigentschaften sollte dann das Verrauschen bringen. Als nächstes soll der Path Tracer die Materialen bekommen. Der Path Tracer ist jetzt auch in den Renderpass eingebaut.

Joseph, Bernhard und Max werden sich nun auch mit den Path Tracer beschäftigen, damit dieser so schnell wie möglich funktioniert. Bis jetzt wurde eine Pipeline aufgestellt, mit der mehrere Passes hintereinander geschalten werden können. Diesen kann man beliebige Attachments mitgeben kann, die dann jeweils in den folgenden Renderpasses auch weiterverwendet werden können. Die GUI ist jetzt auch konfigurierbar.

Grundlage eingebaut.

Maurice Ach hat sich mit der Implementierung von TheVaffel gekümmert dies ist noch mit Problemen besetzt. Der Dozent gab den Tipp, dass eine externe Dependancy nicht installiert wurde. Es soll nun noch das andere SVGF Projekt, welches in QT geschrieben wurde, angeschaut werden. Dieses Projekt eignet sich besser für unser Projekt, da der Shader Code in GLSL geschrieben wurde.

Marvin Potempa hat sich mit der Implementierung von A-Trous gekümmert. Dies wurde auch versucht zu implementieren allerdings fehlte dafür noch ein Grundbaustein.

Tobias Eder hat Referenzbilder aus Blender exportiert und versucht damit mit einen einfachen convolutional Autoencoder Bilder zu rekonstruieren. Es wird nun versucht diesen aufzubauen und zu trainieren. Aufgrund der Speicherlimitierung kann das Netzwerk nur mit einzelnen Bildern trainiert werden.

## 11 Treffen 14.06.2021

Die Termine für die verschiedenen Abgaben wurden festgelegt. Nach der Abgabe würde der Code auch auf GitHub veröffentlich werden.

Joseph Heetel hat den Attachment Manager aktualisiert, außerdem wurde das GUI umgebaut. Die UBO wurden zusammengefasst und in einer Klasse organisiert. Beim Post Process Pass wurde Fehler behoben und eine paar Features wurden hinzugefügt. Max und Josef haben dann den Temporal Akkumulation Shader geschrieben. Dann wurde das Vulkan GLTF Model angepasst, sodass Material ID und Mesh ID übergeben werden. Danach wurde an der Implementierung des BMFR Filters gearbeitet. Der Dozent merkt hierbei an, dass es bei Compute Shader Sinn machen würde mit OpenCL anstatt GLSL zu arbeiten.

Ahmed Salame hat den Path Tracer in den Master gemerged. Dieser ist nun auch nutzbar. Hierbei musste lediglich ein Fehler behoben werden. Die Materiale werden nun richtig geladen sowie die Texturen werden nun auch richtig in die Shader geladen diese sind jedoch noch nicht in den Algorithmus integriert.

Tobias Eder ist mit seinen ursprünglichen Programm auf einen Error gestoßen und hat deswegen mit einer anderen Library weitergearbeitet. Die Features von den alten Programm wurden versucht auf das neue zu übertragen. Da der Intel Endcoder jedoch nur auf der CPU funktioniert müssten die Daten von der GPU wieder auf die CPU geladen werden, um dort decodiert zu werden. Die das decodierten des Intel Endocers schon allein 200ms braucht, wäre diese Lösung nicht echtzeitfähig. Der Vorschlag des Dozenten war es, den Encoder mit einzelnen Bildern zu füttern.

Marvin Potempa wird den Á-Trous diese Woche nun implementieren, dafür wurden Beispielprojekte gesucht. Außerdem wurde an der Ausarbeitung weitergearbeitet und Überlegungen getätigt, wie das Projekt vorgestellt werden könnte.

Maurice Ach hat das Projekt von Qt umgesetzt und hat und konnte Bilder mit RenderDoc exportieren. Diese wurde in eine Grafik verpackt und in die Ausarbeitung eingebaut. Dann wurde sich das Qt Project Code technisch untersucht, um einen Eindruck zu erhalten, ob der Code für unser Projekt benutzbar wäre.

Maximilian Thiel zeigte die temporal Akkumulation welche eingebaut wurde.

Der Dozent teile mit, dass es durchaus denkbar ist das entweder SVGF oder BMFR implementier wird. Hintergrund ist, dass es besser ist ein gut eingebautes Verfahren zu haben als viele schlecht eingebaute.