Mixed Reality Media: Integration of live video feed in 3D environments

---

Martin Zier

July 3, 2017
Version: Initial Drafting

Beuth University of Applied Sciences

CleanThesis

Department VI: Computer Sciences and Media

Bachelor Thesis

Mixed Reality Media: Integration of live video feed in 3D environments

Martin Zier

1. Reviewer Kristian Hildebrand
Department VI: Computer Sciences and Media
Beuth University of Applied Sciences

2. Reviewer Prof. Dr.-Ing. René Görlich
Department VI: Computer Sciences and Media
Beuth University of Applied Sciences

Supervisors Kristian Hildebrand and Joachim Quantz

July 3, 2017

Martin Zier
Mixed Reality Media: Integration of live video feed in 3D environments
Bachelor Thesis, July 3, 2017
Reviewers: Kristian Hildebrand and Prof. Dr.-Ing. René Görlich
Supervisors: Kristian Hildebrand and Joachim Quantz

Beuth University of Applied Sciences
Department VI: Computer Sciences and Media
Luxemburger Straße 10
13353 Berlin

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Acknowledgement

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

1 Introduction

> *If a technological feat is possible, man will do it.*
> *Almost as if it's wired into the core of our being.*
>
> — **Motoko Kusanagi**
> (Ghost in the Shell)

Extending reality with the help of computer generated imagery is no new concept. Ever since real time 3D graphics was possible there was an attempt to extend the understanding of reality. Within the recent years there have been great successes in the industry, most notably in image augmentation was "Pokémon Go" with an estimated install base of 750 million downloads worldwide in June, 2017. [Ann17] Just before this thesis started, Apple and Google showed off their consumer-ready hard- and software for augmented reality experiences.
Virtual Reality Head Mounted Displays have had a similar push in sales with an approximate of 5.83 million sold devices, which range in a sales price between 80 - 900€ for a VR kit, ranging from the very simple Google Daydream View and the very sophisticated HTC Vive. [Erg17] And in these figures are the sales of Google Cardboards missing, which is approximated at around 80 Million.
This generation of computer systems, in which are PC workstations, game consoles and smartphones, is finally sophisticated enough in computation speed and sensor-sensitivity to allow low latency tracking, precise to just a few millimeters.

1.1 Overview

The idea of Virtual Reality (VR) and Head Mounted Displays (HMDs) stems from a cultural need to switch into roles of foreign worlds. Through the advancing development of hard- and software over the last decades emerges a medium which has unmatched immersion and creates an unique, transforming experience into any imaginable environment.

VR and HMDs are now advanced enough for consumer markets - but it stumbles at communicating the experience. Without having ever put on a VR-Headset it is nearly impossible to understand - or even imagine - what the virtual reality experience

means. Any observer of Virtual Reality, usually done by showing what the VR actor is seeing, will not be able to get an understand of the importance and shift of reality perception without wearing the headset himself.
Showing the video output from a HMD as marketing material is contradicting with classic motion video productions. There is even only one famous example where the perspective of a First Person Shooter is reenacted, which was in the overwhelmingly negatively received Doom (2005) movie.

The VR industry, including but not limited to game developers, exhibition creators and creative studios is in need of better communication of their products that includes more than the current headset wearer and allows for a similar, adapted and immersive experience.
The currently method is called "Mixed Reality" (MR) and uses an external camera with the same tracking hardware of the headset to produce a video signal that shows the real world actor with the environment around him. There are currently three main ways of producing MR footage - where as only one variant allows for live compositing with highly accurate imaging results.

1.2 Motivation

My early teenage years started around the time where digitalization and global interconnectivity begun and broadband Internet became commercially available. Suddenly remote multiplayer games, unlimited image sharing - and yes, music sharing, too -, Java-Applets, Flash, HTML framesets and "Marquee" CSS emerged in that medium. 3D Acceleration became a de-facto standard and even simple office PCs got weak, but dedicated graphics processing units built in. The mass of pixels by increasing the resolution of displays was basically a yearly iteration in greater, better, smaller and brighter.

I am personally very interested and invested in Virtual/Mixed/Augmented Reality to succeed and liked the idea to merge multiple forms of media into one - which is, in my personal opinion, a great summary of my studies and its contents. This thesis represents my interests and the reasons why I chose these studies.

1.3 Problem Statement

Initially I will research motion video productions, computer generated imagery and color theory. This leads to the knowledge to implement basic, interactive live motion video.

The core aspect will be integrating a multitude of Hardware in a software that allows for dynamic video compositing in 3D environments at runtime while a user is interacting with the virtual reality scene. This allows that the person using the Vive HMD to be composited into the scenery and it looks like he is in that scene standing. The essential difference between classic post production is, that this system is planned to operate on runtime, allow additional observers to get an interesting composited imagery of what the VR actor is experiencing.
An additional extension is to dynamically track the camer position, allowing for dynamic camera movement and a freely moving actor.

1.4 Relevance & Challenges

1. Komplette Produktionsworkflow
2. Chroma Keying
3. Bildreproduktion
4. Latencymitigation
5. Tracking

Probably should be called Relevance & Scope.

1.5 Results

Result stuff

1.6 Thesis Structure

This thesis gets contemplated by digital, mostly motion video, material hosted on GitHub. Print is a great medium, but lacks the ability for short demonstrations of video imaging solutions, problems and edge cases. To visualize these problems properly, all video media will have an annotation for cross referencing on the website. It is strongly suggested to follow these links, they will be sorted by chapters.

2 Extending Reality

> *You are an aperture through which the universe is looking at and exploring itself.*
>
> — **Alan W. Watts**
> (Philosopher)

The well known urban legend of "L'Arrivée d'un train en gare de La Ciotat" in which a train arrives at the La Ciotat station, is, that "the audience was so overwhelmed by the moving image [...] coming directly at them that people screamed and ran to the back of the room". [Wik] With that a new medium was created, which matured into a new art form of film and movies.

This sounds more like prosa text.

2.1 Motion Video Production

2.2 CGI & Video Composition

2.2.1 History of Green & Blue Screen Productions

2.3 What's VR - Differentiation of AR, VR & MR

2.4 Immersion vs Communication

2.4.1 Evolution of Virtual Reality Footage

2.5 Mixed Reality and its use cases

Summarize.

3 System Setup

The following section describes the hard- and software components used for the thesis and results. All demonstrations have been performed on that environment. All dependencies have been explicitly marked to allow a similar, but not exact, setup to reproduce these results.

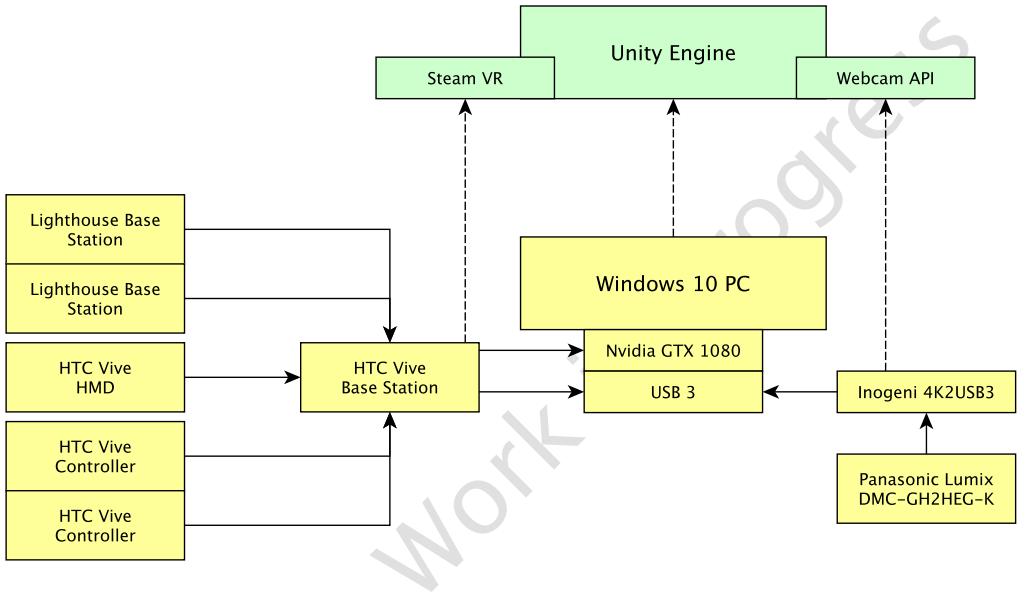

Fig. 3.1: Diagram of hard- and software components.

3.1 Hardware Configuration

The hardware configuration is split in three main parts:

1. Windows PC Workstation
2. Virtual Reality Tracking Solution
3. Motion Video Input Feed

Each individual configuration is basically interchangeable with other systems, as long as predefined conditions are met. Each condition is listed first in each subsection.

3.1.1 PC Workstation

As the software is built in the Unity Engine, the workstation is limited to either Windows or Mac OS X systems the only requirement - besides being powerful enough to render the 3D scenes - is two USB3 ports to ensure enough data throughput for the video and virtual reality solution, as well as two video outputs for a monitor and its headset.
The configuration used here is:

```
CPU: Intel i7-4700K @ 4.00 GHz
RAM: 16GB DDR4
GPU: Nvidia GTX 1080
```

This system configuration is to date a high end workstation that has an abundance of render performance, allowing it to process and keep enough framebuffer for the operation described further in.

weak text.

3.1.2 Inogeni 4K2USB3

The Ingoeni 4K2USB3 converter is a standalone box that allows to receive any HDMI source and converts it as external webcam video feed. It's advantage is by the arbitrary choice of video cameras and a very simple integration with any software. With the help of the converter box it's possible to request a webcam as video resource and process that video feed as a texture on the GPU.

3.1.3 Panasonic GH2 Systemcamera

This camera provides a direct video feed via HDMI with low latency. It can directly feed into the Inogeni 4K2USB3 and produces a stable, high quality video feed with a low signal to noise ratio in well lit environments.

still unclear if this remains the target camera.

3.1.4 HTC Vive with Controllers and Lighthouses

The current best virtual reality and tracking device is the HTC Vive. It includes two infrared sending stations called "Lighthouse", two Vive Controllers and a Headset,

both systems with 6 degrees of freedom (6DOF) tracking. The tracking system is a blackbox, in which only the transformation matrices for the hand controllers and the HMD can be accessed. By default this transformation has a normalized length of 1 unit to 1 meter. Designing scenery and sense of size is therefore rather easy. The data providing is done by a library called "SteamVR for Unity", which makes the usage in engine transparent.

3.1.5 Vive Controller Tripod Mount

Most cameras have a standardized way of mounting tripods. Since the Vive controllers have no reference plane and minuscule differences in mounting angles changes the projection parameters to noticeable effects, it was necessary to build a mount for the camera to keep controller and video equipment transformation in sync, I built a mount that fits on tripod attachment points and keeps the controller locked in the same position.

Fig. 3.2: Camera mount for a HTC Vive controller

add example for incorrect projection and model of mount

3.2 Software

The software of choice is Unity3D, which is free for students, non-profit organizations and small studios. It provides an easy introduction to game / 3D engine program-

ming and has a huge development community. While it is not the technologically most advanced engine, its fairly easy usage and fast development cycles make it a great tool for a bachelor thesis.
Thankfully, the high abstraction of system APIs means that cross-platform development only needs a single code base and makes excruciating tasks like webcam access simple - so much so that it boils down to one line of code.
It's weaknesses is usually API documentation and - on the downside, too - high abstraction levels from most APIs. In example, Unity relies on its own shading language which cross-compiles to HLSL, OpenGL and WebGL - this leads to problems in framebuffer management, which cannot be controlled well inside the engine.

4 From Video to Mixed Reality

Needs better sourcing.

To achieve a real time rendering environment, as previously mentioned, there are two main production cycles. The one discussed in this thesis resolves this problem by staying inside on application with multiple render cycles per frame.
The first and foremost render cycle is the stereoscopic output of the Vive HMD, which has a set framerate of either 45 or 90 frames per second. It is important to have a consistent performance, otherwise the experience for the actor with the HMD will have a terrible experience.
The secondary render cycle has to be done on the same frame, which is a virtual camera inside the virtual scene and the relative position of the real world HMD and real world camera. Since the SteamVR library for the HTC Vive already exposes a normalized, synchronized tracking, it is easily possible to position the virtual camera at an accurate location.

The following chapter describes the techniques used to transform motion video inside a greenscreen into a mixed reality image. As brief overview, the steps required are performed in referential order from the motion video from the camera feed. This is different to the render order but gives a better understanding of the techniques used to achieve mixed reality imagery.

4.1 Chroma Key

Beginning from the camera, the video signal travels through the Inogeni converter and is accessible with the system API for webcams. (See figure 3.1)

The initial step is to remove the green background from the image, which should be greenscreen. For a reference green, there has to be a color picked manually in the material editor of Unity - this was made easy by a checkbox to show raw output from the camera. Then a middle-ground green can be picked. This is an important setup step, since lightning situations can vary greatly and minor differences in light setups can have a great effect on the outcome of visible green background captured by the camera.
An extreme example case is used for comparing these chroma keying variants:

Fig. 4.1: Comparison Image[Vim] - sRGB Output

4.1.1 Initial Assumption

Each RGB color can be represented as a discrete 3-Vector of (red, green, blue) values in range of [0, 1]. The composition between two colors can be summarized as equation as following, where a foreground image F and a background image B - α_B is assumed to be 1:

$$I(x,y) = \alpha(x,y)F(x,y) + (1 - \alpha(x,y))B(x,y) \tag{4.1}$$

This matting equation has to be generalized, where α is a value between [0, 1] on fore- and background, yielding a total Alpha of α_T as following:

$$\alpha_T = \alpha_B * (1 - \alpha_F) \tag{4.2}$$

$$I(x,y) = (1 - \alpha_T)F(x,y) + \alpha_T B(x,y) \tag{4.3}$$

4.1.2 Euclidean RGB Difference

Assuming a source pixel color C_S and a reference color C_R we can calculate the euclidean distance between these colors.

$$\alpha = \sqrt{(C_R R - C_S R)^2 + (C_R G - C_S G)^2 + (C_R B - C_S B)^2} \tag{4.4}$$

This is a computationally very low cost and works well enough for tell a difference between two separate colors. It fails to accommodate for colors that are perceived as different, but are tinted by the reference colors. Since the greenscreen will never achieve 0% reflectivity, some residue of the background color will mix with the filmed actors.

4.1.3 Euclidean YCgCo Difference

YCgCo stands for Luminance (Y), chrominance green (Cg) and chrominance orange (Co) and helps decorrelating color spaces. Since it is a fast, lossless color transformation it is used in example for H.264 video encoding. The two chrominance channels are then split into green to magenta and orange to blue color values and allow for a more accurate distance calculation between two colors.
Transforming any arbitrary RGB color to YCgCo can done with a single matrix multiplication:

$$\begin{bmatrix} Y \\ Cg \\ Co \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.5}$$

Fig. 4.2: Chroma Keying by using euclidean RGB distance

Given two colors, one from the video source C_S and a reference color C_R it is now possible to calculate the euclidean distance on the two chrominance channels:

$$\alpha = \sqrt{(C_R Cg - C_S Cg)^2 + (C_R Co - C_S Co)^2} \tag{4.6}$$

Since the increased decorrelation, the result is more accurate and shows less artifacting on unwanted pixels.

4.1.4 Euclidean Lab Difference

The International Color Consortium (ICC) defined 1976 *Lab* ΔE as a standard way of calculating color differences with *Lab* colors. The final distance calculation is the linear euclidean distance as with all other models, but accommodates for perceived color differences.

this is very rough and only contains equations currently used

sRGB conversion to linear RGB in respect of energy per channel:

Fig. 4.3: Chroma Keying by using euclidean YCgCo distance

$$v \in \{r, g, b\} \wedge V \in \{R, G, B\} \tag{4.7}$$

where:

$$v = \begin{cases} V/12.92 & \text{if } V \leq 0.0405 \\ ((V + 0.055)/1.055)^{2.4} & \text{otherwise} \end{cases} \tag{4.8}$$

from there l

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.9}$$

where:

$$\left[M\right] = \begin{bmatrix} RX_r & GX_g & BX_b \\ RY_r & GY_g & BY_b \\ RZ_r & GZ_g & BZ_b \end{bmatrix} \tag{4.10}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \left[M\right] = \begin{pmatrix} X_r/Y_r & X_g/Y_g & X_b/Y_g \\ 1 & 1 & 1 \\ \frac{1-X_r-Y_r}{Y_r} & \frac{1-X_g-Y_g}{Y_g} & \frac{1-X_b-Y_b}{Y_b} \end{pmatrix} \tag{4.11}$$

Where $\left[M\right]$ for RGB D65 is:

$$\begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \tag{4.12}$$

Based on a reference white $U_r \in \{X_r, Y_r, Z_r\}$:

$$U \in \{X, Y, Z\} \wedge W \in \{L, a, b\} \tag{4.13}$$

$$\epsilon = 0.008856 \wedge \kappa = 903.3 \tag{4.14}$$

where:

$$w_r = \frac{U}{U_r} \tag{4.15}$$

$$f(w) = \begin{cases} \sqrt[3]{w_r} & \text{if } U > \epsilon \\ \frac{\kappa w_r + 16}{116} & \text{otherwise} \end{cases} \tag{4.16}$$

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} 116 f_y - 16 \\ 500(f_x - f_y) \\ 200(f_y - f_z) \end{bmatrix} \tag{4.17}$$

With this conversion from sRGB to linear RGB to XYZ to Lab we can now calculate the euclidian linear distance between two colors C_1 and C_2, which already have been converted to Lab:

$$\Delta E = \sqrt{(C_2L - C_1L)^2 + (C_2a - C_1a)^2 + (C_2b - C_1b)^2} \tag{4.18}$$

These values are rated by their perceptive difference [MW]:

0.0 . . . 0.5	the difference is unnoticeable
0.5 . . . 1.0	the difference is only noticed by an experienced observer
1.0 . . . 2.0	the difference is also noticed by an unexperienced observer
2.0 . . . 4.0	the difference is clearly noticeable
4.0 . . . 5.0	fundamental color difference
> 5.0	gives the impression that these are two different colors

Now it's possible to map alpha values for each pixel based on ΔE distances between m, n by clamping and biasing ΔE:

$$f(\Delta E) = x = \frac{\Delta E - n}{m - n} \tag{4.19}$$

$$\alpha_{\Delta E} = \begin{cases} n & \text{if } x \leq n \\ x & \text{if } n \leq x \leq m \\ m & \text{if } m \leq x \end{cases} \tag{4.20}$$

$$\alpha(I(x, y)) = 3\Delta E^2 - 2\Delta E^3 \tag{4.21}$$

Fig. 4.4: Chroma Keying by using ΔE distance

4.2 Camera Offsets

After rather simple integration of the keyed video signal into the scene, where the 3D environment is used as background, I have observed an offset between the render image from the scene and the captured footage from the camera. After reading further into the specification of the Inogeni 4K2USB3 where it states that the conversion takes two intrinsic frames for encoding. The sent framerate of the camera is at 25 frames per second. This would mean, in theory:

$$t = 2 * 1/fps \tag{4.22}$$

Assuming 30 frames per second, that is $\frac{1}{30}s$:

$$t = 2 * 1/30 \frac{1}{second} \tag{4.23}$$

$$t = 66ms \tag{4.24}$$

The observed offset is however far longer and is at about 260ms and therefore noticeable in any motion video.

There will be a sample video here.

To mitigate this offset there are two options:

1. Change the camera - in example for a webcam, which usually has a lower offset, ranging between 5 - 50ms. That would degrade the image quality significantly but would enable perfect rendering conditions inside the engine.
2. Capture virtual images of the 3D environment and keep them on the GPU until a video frame is loaded onto the GPU for usage. This keeps the image quality but needs reasonable effort to reproduce the rendering conditions when the video frame was taken.

My solution uses the secondary solution, since it is able to minimize any kind of offset between render image and video image, the setup can stay dynamic and it is little to no difference to switch to a webcam-integrated solution, than an encoding box.

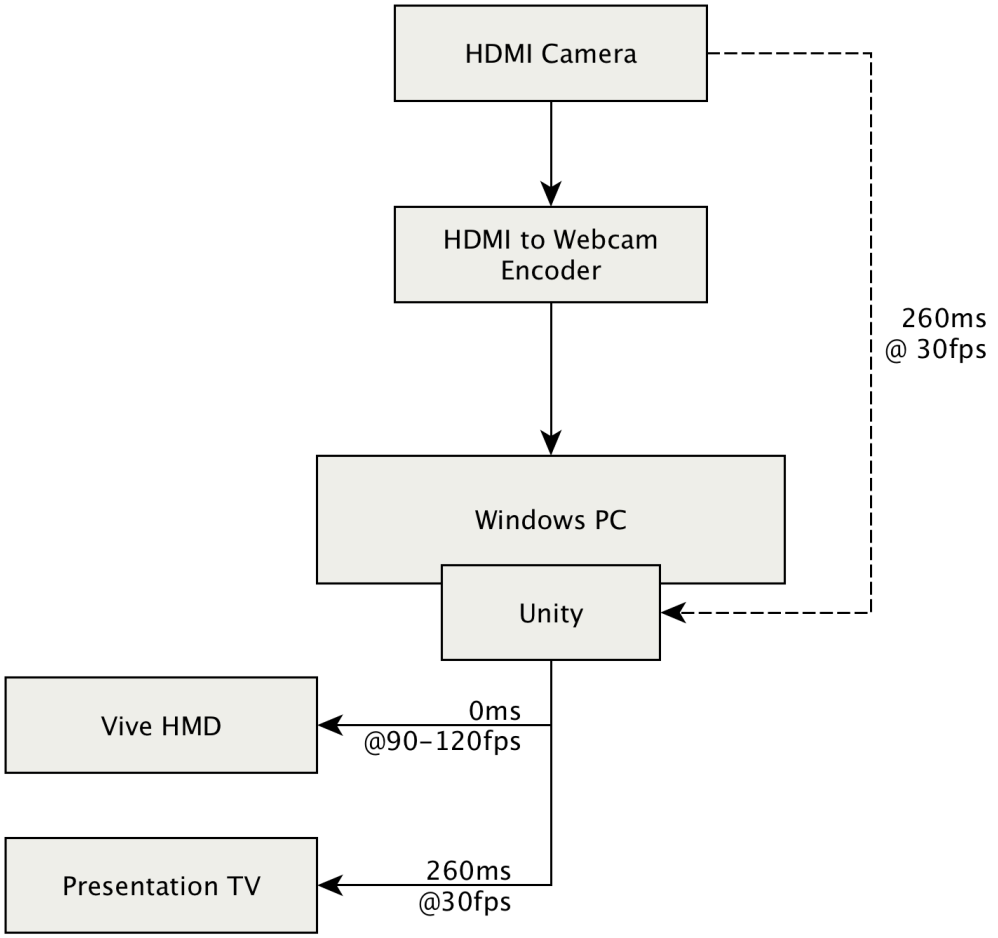

Fig. 4.5: Components in considering timing offsets

Based on the component diagram 4.5 there are two important notes: The time offset between camera to Unity and the TV and Vive HMD framerate differs. At time of writing Unity does not support dynamic framerates on multiple cameras. It is possible to manually initiate a render, however, this causes the render loop to mistime and yields to inconsistent frame timings inside the HMD.

kinda weak text, it's a lot of random words imo

To conclude: The software has to store a set amount of framebuffers and cycle them at the right frame to guarantee minimal delays between camera and 3D environment.

Noteworthy is that the render loop can be 45 or 90 fps, depending on scene complexity, slow system performance or - in case of Unity - garbage collection, that could halt the engine. To account for this a strategy is needed in which Unity's

`Time.deltaTime` property is used, which describes the time between the last and current frame.

4.2.1 Framebuffer Swapper implementation

Unity has a well engineered engine loop. It has a

As initial data needed is `cameraFPS` , `cameraOffset` and `Time.deltaTime` . From there it is possible to calculate the remaining data for this algorithm:

```
frameWindow = 1.0 / cameraFPS;
delayCnt = cameraOffset / frameWindow;
frameDelay = (int) delay * frameWindow;
fractionDelay = delay % (1 * frameWindow);
innerTimer = 0.0;
absoluteTimer = 0.0;
while(true) {
        innerTime += Time.deltaTime;
        absoluteTimer += Time.deltaTime;
        localTime = innerTimer - fractionDelay;
        if(localTime < frameWindow ||
           absoluteTimer < initialDelay) {
                continue;
        }

        innerTimer %= frameWindow;
        absoluteTimer %= (1f + initialDelay);
}
```

4.3 Mitigating Frame Jitter

Bibliography

[Ann17] App Annie. *App Annie 2016 Retrospective*. Retrospective Report. 2017, pp. 2, 25 (cit. on p. 1).

[MW] Tatol M. Mokrzycki W.S. *Colour difference ΔE - A survey*. Survey. Faculty of Mathematics, Informatics, University of Warmia, and Mazury, p. 20 (cit. on p. 17).

Websites

[Erg17] Deniz Ergürel. *The latest virtual reality headset sales numbers we know so far. As of March 2017*. 2017. URL: https://haptic.al/latest-virtual-reality-headset-sales-so-far-9553e42f60b5 (cit. on p. 1).

[Vim] *#INTRODUCTIONS (2015)*. Vimeo. 2015. URL: https://vimeo.com/125095515 (cit. on p. 12).

[Wik] *L'Arrivée d'un train en gare de La Ciotat*. URL: https://en.wikipedia.org/wiki/L%27Arriv%C3%A9e_d%27un_train_en_gare_de_La_Ciotat (cit. on p. 5).

List of Figures

List of Tables

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Berlin, July 3, 2017

Martin Zier