

Projekt Digitale Spiegel

Dokumentation

Autor: [Christian Daus](#),

Lead Developer @ expanding focus GmbH,
August 2023



expanding focus

Einleitung

Projektbeschreibung

Die “digitalen Spiegel” sind drei große Monitore (inkl. PC), die mithilfe von Kameras den davor stehenden User “spiegeln” und digitale Inhalte, wie Kostüme und Masken hinzufügen. Durch Echtzeit-Body-Tracking mit zwei Kameras können die digitalen Inhalte den Bewegungen des Users folgen und an dessen individuelle Proportionen angepasst werden. Man spricht hier von einer Augmented Reality Installation (Augmented Reality = erweiterte Realität).

Damit bieten die “Spiegel” die Möglichkeit, Kunst aus dem Theater interaktiv zu erleben und kurzzeitig in eine Rolle bzw. in ein Kostüm oder das Make-up aus Oper oder Ballett zu schlüpfen.

Die detailgetreuen Kostüme bzw. Masken werden automatisch gewechselt, um eine kurzweilige und abwechslungsreiche Erfahrung zu schaffen.

Es wurden zwei große und ein schmaler Spiegel im Foyer der Oper installiert.



Das Projekt entstand in Zusammenarbeit folgender Parteien über etwa 6 Monate:

Deutsche Oper am Rhein

Abteilung Marketing und Kommunikation
Abteilung Dramaturgie
Kostümabteilung
Lena tom Dieck

Konzeption
Projektkoordination
Vorlagen für Kostüme

expanding focus GmbH

Hardware-Konfiguration
Softwareentwicklung

Stephan Karass

Bau & Installation Rahmen

Marc Suski

Folierung Rahmen

Benjamin Hohnheiser

Konzeption

Ziel dieser Dokumentation

Dieses Dokument soll einen Einblick in den Aufbau und die Funktionsweise der digitalen Spiegel geben. Gleichzeitig werden relevante Aufgaben angesprochen, die während des Lebenszyklus des Produktes anfallen.

Die Leserin bzw. der Leser soll dadurch befähigt werden, auftretende Probleme im System zu identifizieren und effektive Schritte zu deren Lösung ableiten zu können.

Zudem vermittelt die Dokumentation Details zu den Entscheidungen bei essentiellen Komponenten. Diese sollen dabei helfen, das System auf neue Bedürfnisse oder Rahmenbedingungen anzupassen und zukünftige Projekte zu unterstützen.

Dazu werden zunächst alle essentiellen Features der drei Spiegel erläutert.

Dann wird auf die verwendeten Hardwarekomponenten und deren Anforderungen für dieses Projekt eingegangen.

Es folgen ein Überblick über die allgemeine Software-Konfiguration und Details zur entwickelten Unity Anwendung. Dabei soll ein Grundverständnis über die Funktionsweise des Body-Trackings vermittelt werden.

Abschließend wird auf Möglichkeiten zur Anpassung und Fehlerbehebung der Software eingegangen.

Features

Hier werden alle essentiellen Funktionen der Software bzw. Hardware kurz beschrieben.
Details zur Funktionsweise befinden sich in den Kapiteln Software bzw. Hardware.

Ein- und Ausschalten

Zum Starten der Anwendung muss nur der PC eingeschaltet werden.
Das Programm startet nach kurzer Zeit automatisch.
Der PC beendet zu einer festlegbaren Zeit das Programm und fährt automatisch herunter.

Regulärer Betrieb

Die Anwendung besitzt zwei Modi: den "Sleep Mode" und den "Tracking Mode".

- nach Start: "Sleep Mode"
 - das Kamerabild der rechten Kamera wird angezeigt
 - es wird alle 1,0 Sekunden versucht, einen User im Kamerabild zu finden und zu tracken
 - bei erfolgreichem Tracking*: Wechsel zu "Tracking Mode"
- aktiver Betrieb: "Tracking Mode"
 - informatives UI wird eingeblendet
 - Logo der Oper am Rhein
 - Logo von Expanding Focus
 - Name des Kostüms, zugehöriges Stück und Komponist/Künstler
 - Tracking des Users wird ständig aktualisiert (etwa 30 mal pro Sekunde)
 - mithilfe beider Kamerabilder wird ein virtueller 3D Charakter des Users berechnet und animiert
 - aktuelles Kostüm wird animiert, gerendert und in das Kamerabild eingefügt
 - das Kostüm wird so animiert, dass es auf den virtuellen Charakter passt und damit auch auf den User im Kamerabild
 - nach festgelegter Zeit wird das Kostüm gewechselt
 - bei Tracking Fehler*
 - Kostüm Animation wird gestoppt
 - Kostüm wird langsam ausgeblendet (von opaque zu transparent)
 - nach bestimmter Zeit: Wechsel zu "Sleep Mode"

*: Tracking wird als erfolgreich betrachtet, wenn die (virtuellen) 3D Positionen der notwendigen Teile des Körpers zuverlässig berechnet werden können. Dazu müssen diese von beiden Kameras sichtbar sein. Der User sollte möglichst im Abstand von 1 bis 3 Metern, zentral vor dem Spiegel stehen.

Hardware

Rahmen

Der Rahmen ist eine Stahlkonstruktion, bestehend aus einem inneren Gestell aus verschweißten Hohlprofilen und einer dünnen Blechverkleidung, die mit dem Gestell verschraubt ist. Die Verkleidung wurde anschließend foliert.

Die groben Maße der Spiegel sind die folgenden:

- großer Spiegel: Breite 90 cm, Höhe 167 cm, Tiefe 20 cm
- kleiner Spiegel: Breite 50 cm, Höhe 140 cm, Tiefe 20 cm

An der Vorderseite befindet sich der Monitor. Der Rahmen ist relativ tief, sodass ein kleines PC Gehäuse (ITX) hinter dem Monitor Platz finden kann.

Die 2 schmalen Seitenbleche sind über jeweils zwei Schrauben am Gestell befestigt. An diesen Blechen sind auch die Kamerahalterungen verschraubt, und es gibt jeweils ein Loch für das Datenkabel von der Kamera zum PC.

Die Kamerahalterung ist ein flacher, zweifach gebogener Messing-Streifen. Er ist am Seitenblech mit Schrauben befestigt. An der Kamera wurde das ¼ Zoll Gewinde zur Befestigung genutzt.

Die obere Seite des Rahmens ist offen, um eine ungestörte Luftabfuhr von der Luftkühlung des PCs zu gewährleisten.

Die Unterseite ist ebenfalls nicht verkleidet, jedoch wurde ein bewegliches Gitter eingelegt, auf dem das Gehäuse des PCs stehen kann. Dieses Gitter schließt den Rahmen zum Teil nach unten ab, lässt links und rechts jedoch etwas Luft, um Zugang zum PC zu haben. Die Löcher im Gitter sorgen außerdem für die notwendige Luftzufuhr zum PC.

Der Monitor ist über das integrierte Befestigungssystem für Wandhalterungen (VESA Norm) an der Vorderseite des Gestells verschraubt. Ein weiteres Blech ist auf der Vorderseite mit vier Schrauben befestigt, welches den Rest der Vorderseite vollständig verkleidet.

Es wurden auch andere Rahmen-Designs in Betracht gezogen. Für eine flache Bauweise kann der PC unterhalb des Displays montiert werden. Der PC könnte auch nur über ein langes Kabel mit dem Display verbunden sein, aber sich außerhalb des Rahmens befinden, bspw. in einem Nebenraum.

Eine Montage der Kameras innerhalb des Rahmens, ohne äußere Halterungen, hätte diese vor äußeren Einflüssen geschützt, wäre aber komplizierter gewesen.

Monitor

Die wesentlichen Anforderungen an die verwendeten Monitore waren die folgenden:

- möglichst passende Größe, vergleichbar zu den originalen Spiegeln im Foyer
- ausgelegt für Dauerbetrieb, bzw. 16 Stunden pro Tag, 7 Tage die Woche
- im Hochformat montierbar
- schmaler, gleichmäßiger Rahmen
- möglichst geringe Tiefe
- ausreichend hohe Auflösung (4K für die großen, FHD für den schmalen Spiegel)
- gutes Kontrastverhältnis und hohe Helligkeit sind wünschenswert, aber keine notwendigen Kriterien, da die Spiegel nur im Inneren des Gebäudes genutzt werden
- große Reaktionszeiten sind zu vermeiden, um die Echtzeit Interaktionsfähigkeit des Gesamtsystems nicht zu gefährden
- Panel-Typ: entspiegelt und IPS wären zu bevorzugen, wegen besserer Bildqualität aus ungünstigeren Betrachtungswinkeln
- idealerweise geringer Stromverbrauch
- Verfügbarkeit und Preis

Die Zertifizierung für Dauerbetrieb begrenzte die Auswahl auf Digital-Signage Produktvarianten, sodass herkömmliche Fernseher oder Büromonitore keine Option waren.

Insgesamt wurden Displays von folgenden Herstellern bzw. Lineups als Option betrachtet:

- iiyama Prolite (LE, LH, ...-B1, ...-B3)
- Philips (X-Line, D-Line)
- Dell (e.g. C5519Q)
- Sharp NEC (M, MA, ME, P, PN, PN-HY, PN-HS, PN-HW, E, C, V)
- Samsung (QBB, QBR-B, QHB, QHR, QMB, QMR, QMR-A, QMR-B)

Je Displaygröße ergaben sich dabei folgende Kandidaten:

Zoll	Modell
32	Samsung QM 32 R-B
43	iiyama LH 4370 UHB-B1
50	Sharp NEC NEC ME501
50	iiyama LH 5070 UHB-B1
55	Sharp NEC ME551
55	iiyama LH 5552 UHS-B1
65	Sharp NEC ME651
65	iiyama LH 6570 UHB-B1
75	Samsung QH75B
75	Samsung QM75R

Wegen der geringen Breite wurde für den schmalen Spiegel der Samsung QM 32 R-B gewählt. Für die beiden großen Spiegel wurde der Sharp NEC ME651 gewählt.

Die Displays erfüllten insgesamt die Anforderungen für das Projekt sehr gut. Beim schmalen Spiegel wäre eine höhere Auflösung als FHD wünschenswert gewesen, gleichzeitig fiel diese nicht störend auf.

Kameras

Die Kameras bilden das Interface zwischen dem Betrachter und dem Spiegel. Sie müssen zunächst ein flüssiges und qualitativ hochwertiges Bild liefern, das auf dem großen Display selbst aus geringer Entfernung gut aussieht.

Die Mindestvoraussetzungen dafür sind 30 Bilder pro Sekunde (FPS) und ein unkomprimiertes Videobild mit einer Auflösung von 1920x1080 Pixeln (1080p). Da die Kameras auch für das Body Tracking genutzt werden, sind 60 FPS zu bevorzugen, damit das System direkter auf Bewegung reagiert. Doch auch bei gleicher Auflösung kann die Bildqualität von Kamera zu Kamera stark variieren, da bspw. Sensorgröße, Linse oder (Auto-) Fokus auch eine wichtige Rolle spielen.

Ein möglichst großer Blickwinkel ist von Vorteil, um den Bereich zu vergrößern, in dem der User getrackt werden kann.

Ein möglichst unverzerrtes Kamerabild ist ebenfalls wünschenswert.

Gleichzeitig sollte die Kamera möglichst flach sein, oder zumindest nicht wesentlich tiefer als die geplante Tiefe des Rahmens.

Hochwertige Webcams erfüllen alle Kriterien sehr gut. Die Bildqualität ist aufgrund der sehr kleinen Bauform jedoch begrenzt. Es gibt auch die Option, andere Typen von Kameras (z.B. Action Kameras wie GoPro Hero 10 oder DJI Action 2) zu verwenden, und diese per Capture Card (z.B. CamLink) live abzugreifen. Das Setup ist jedoch etwas umständlich und fehleranfällig. Es wurde zum Zeitpunkt der Recherche (Oktober 2022) keine Lösung gefunden, die den Mehraufwand (und die Kosten) rechtfertigen konnte.

Alternativ könnte eine XBox Kinect für das Tracking genutzt werden, sodass nur eine hochwertige Kamera benötigt wird. Diese könnte dann z.B. mit 30 FPS und 4K Auflösung betrieben werden. Dieser Weg wurde nicht gewählt, da die Kinect ein sehr altes Gerät ist und die maximale Tracking-Qualität sehr beschränkt ist. Es wäre rückblickend die einfachere Lösung gewesen, hätte aber auch unerwartete Probleme verursachen können, da das Tracking auf wenige Nutzer beschränkt ist.

Eine weitere, einfachere Option wäre die Nutzung eines iOS Devices, in Kombination mit dem ARKit gewesen. Dabei fallen jedoch periodische Lizenzgebühren an, und es muss ein extra Gerät eingerichtet, verwaltet, eingebaut und manuell an- bzw. ausgeschaltet werden. Diese Lösung wurde im Vorfeld wegen zu vielen unkalkulierbaren Risiken ausgeschlossen.

Als Optionen für Webcams wurden folgende Modelle betrachtet:

- Logitech Brio 4K Pro
- Elgato Facecam (max. 1080p)
- Opal C1, Lumina 4K, Insta 360 One RS 4K

Logitech und Elgato kamen in die nähere Auswahl und wurden beide getestet. Schließlich wurde die Logitech Kamera gewählt, aufgrund der 4K Auflösung, guten Bildqualität, gutem Autofokus und der kompakten Bauweise.

PC

Die Anforderungen an den PC werden im Folgenden aufgelistet.

Physische Rahmenbedingungen:

- bildet Interface zwischen Kameras und Display
- flache Bauweise zur Integration in den Rahmen des Spiegels
- geringer Stromverbrauch, Geräuschpegel und Hitzeentwicklung
- stabil, langlebig
- günstige, verfügbare Komponenten

Computing Capabilities:

- Verarbeitung der Kamera-Bilder für Echtzeit Body Tracking
- Konstruktion eines virtuellen Charakters
- Animation von Kostümen und Compositing
- Rendern des finalen Bild in 4K bzw. 1080p mit 60 FPS

Anstelle eines separaten Windows PCs wäre auch die Verwendung eines Smart Display Modules (z.B. "Intel SDM-L") denkbar. Allerdings haben diese "Mini-PCs" keine dedizierte Grafikkarte und besitzen daher u.U. nicht genug Rechenleistung für Body Tracking in hoher Qualität. Dann wäre es theoretisch möglich, eine externe Grafikkarte zu nutzen. Diese Option wurde jedoch, wegen fehlender Erfahrungen damit, nicht weiter untersucht.

Für einfachere Anwendungen kann auch das integrierte Android Betriebssystem des Displays genutzt werden.

Dabei ist zu beachten, dass sowohl ein integriertes Android Betriebssystem, als auch die Möglichkeit für ein Smart Display Modul, abhängig vom Modell des Displays sind.

Als nächstes wird konkret auf die einzelnen Komponenten des PCs eingegangen.

Gehäuse

Die Gehäusemaße müssen zum Design des Rahmens passen, also möglichst flach sein, und trotzdem muss eine dedizierte Grafikkarte Platz finden. Das beschränkt die Auswahl auf spezielle Small Form Factor Gehäuse. Optimal wäre ein Custom angefertigtes Gehäuse für diesen Use-Case.

Folgende Gehäuse Optionen wurden in Betracht gezogen:

- Formd T1
- Dr. Zaber Sentry 2.0
- SFF Time P-ATX V3
- Skyreach 4 Mini
- Fractal Design Node 202
- Silverstone: "SST-ML05B", "SST-PT13B, HTPC", "ML 06 E"
- Raijintek: "Pan Slim White, HTPC", "Ophion 7L"
- Raijintek
- Dan A4
- Ghost S1

- Ncase M1
- Streacom DA2
- Cryorig Taku
- NZXT H200
- GEEK: "a31s", "A50", "A50S"
- Velka 5, Velka 7
- Sliger: "SV590 Mini-ITX Vertical", "CL520 Mini-ITX"

Aufgrund von Verfügbarkeit, Modularität und passenden Maßen wurde das "DAN A4 V4.1" gewählt.

Netzteil

Beim Netzteil gibt es nicht besonders viel Auswahl, da nur bestimmte Small Form Factor Varianten in das Gehäuse passen. Abgesehen davon sollte die maximal mögliche Leistung des Netzteils deutlich höher sein, als die maximal benötigte Leistung des Systems. Eine hohe Effizienzklasse ist ebenfalls sehr wichtig, um den Stromverbrauch zu reduzieren. Es müssen außerdem genug Stromkabel für die Grafikkarte zur Verfügung gestellt werden. Daher wurde das "Corsair SF750 80 plus Platinum SFX" gewählt.

Motherboard

Anforderungen an das Motherboard sind hauptsächlich:

- Format: Mini ITX
- Kompatibilität zur gewünschter CPU (Bios Update)
- mindestens zwei dedizierte highspeed USB controller (3.0), um die Datenrate für den Videostream der Kameras nicht zu limitieren
- Langlebigkeit, Stabilität

Es wurde sich für ein AMD X570 Motherboard entschieden. Grund dafür sind zum einen die höherwertigen Komponenten, um ein möglichst robustes und langlebiges System zu schaffen. Zum anderen ist die Plattform (AM4) sehr erprobt und energieeffizient.

Als Referenz wurde ein fertiger Build aus dem Internet ausgewählt, um sicherzugehen, dass alle Komponenten zusammenpassen und im Gehäuse Platz finden.

Wegen mangelnder Verfügbarkeit wurden zwei verschiedene Modelle verwendet:

- (zwei) große Spiegel: Gigabyte AMD X570 I AORUS PRO WIFI (rev. 1.0)
- (ein) kleiner Spiegel: ASRock X570 Phantom Gaming-ITX/TB3

SSD

An die Festplatte des Systems bestehen keine besonderen Anforderungen, außer dass es eine M.2 SSD sein sollte. Weder Betriebssystem, noch die Anwendung benötigen viel, oder besonders schnellen Speicher. Langlebigkeit wäre von Vorteil.

Es wurden folgende SSDs verbaut (wegen mangelnder Verfügbarkeit):

- (zwei) große Spiegel: Kingston NV2 NVMe PCIe 4.0 M.2 Typ 2280, 500 GB
- (ein) kleiner Spiegel: Crucial P3 Plus NVMe PCIe 4.0 M.2 Typ 2280, 500 GB

CPU

An die CPU bestehen folgende Anforderungen:

- AM4 kompatibel
- geringe Leistungsaufnahme, da nur ein flacher Luftkühler im Gehäuse möglich ist
- keine hohe Auslastung erwartet, unabhängig vom Modell
- energieeffizient

Es wurde der Ryzen 5 5600 gewählt.

Das Bios der Gigabyte Motherboards wurde dafür aktualisiert.

RAM

Der Arbeitsspeicher muss weder besonders schnell sein, noch eine große Kapazität besitzen. Es ist jedoch eine flache Bauweise notwendig.

Es wurde folgender RAM gewählt: Corsair Vengeance LPX DDR4-3600, 16 GB (2x 8 GB).

Grafikkarte

Die GPU ist die Kernkomponente des Systems. Sie übernimmt den Hauptteil des Trackings sowie das Rendering des finalen Bildes in 4K bzw. 1080p mit 60 FPS.

Gleichzeitig sollte genug Performance Headroom bleiben, um die Tracking Qualität ggf. zu erhöhen. Es wurde eine moderne Karte gewählt, um eine möglichst hohe Energieeffizienz und dadurch geringe Abwärme zu erzielen. Aus diesem Grund wäre u.U. ein Nvidia Quadro eine bessere (langfristige) Option. Außerdem muss die GPU in das Gehäuse passen.

Es wurde folgende GPU gewählt: Gigabyte RTX 3070 Eagle.

Kühlung

Eine Wasserkühlung bietet eine flexiblere Bauweise, bessere Kühlleistung und geräuscharmen Betrieb. Um den Build Prozess und Wartung zu vereinfachen, wurde jedoch eine Luftkühlung verbaut. Luftkühlung bietet üblicherweise auch ein langlebigere Lösung.

Konkret sind folgende Kühler für die CPU verbaut:

- (zwei) große Spiegel: Noctua NH-L9a-AM4
- (ein) kleiner Spiegel: Noctua NF-A9i (wegen Motherboard Kompatibilität)

Zusätzlich sind zwei Gehäuselüfter verbaut:

- Noctua NF-A9x14 PWM (14 mm)
- Noctua NF-A9 PWM (25 mm)

Wartung

Um Überhitzen und dadurch Schäden an der Hardware zu vermeiden, sollte der PC regelmäßig (2 mal pro Jahr) entstaubt werden. Dazu kann z.B. das äußerste Gehäuseblech des PCs demontiert werden und mit handelsüblichem Druckluftspray gearbeitet werden (vorsichtig). Bei starker Staubbelastung können Staubfilter außen am Gehäuse befestigt werden. Die Linsen der Kameras sollten sauber gehalten werden.

Software

Als nächstes wird auf die Einrichtung des Betriebssystems eingegangen. Dabei werden auch alle zusätzliche Programme aufgelistet, die auf dem PC installiert wurden. Es folgen Details zur entwickelten Anwendung.

Betriebssystem

Es wurde Windows 10 Home installiert.

Eine möglicherweise bessere Option wäre Windows IoT (Core/Enterprise) gewesen. Dieses ist speziell für den Anwendungsfall Digital Signage gedacht und ermöglicht u.a. Remote Management der Systeme über die Cloud. Die Option wurde nicht gewählt, da keine Erfahrung mit dieser Windows Variante vorlag und die PCs offline betrieben werden sollten.

Folgende Settings wurden innerhalb von Windows angepasst:

- Login: Name: "User", Password: "1234", Sicherheitsfragen (3): "a", "a", "a"
- automatischer Login
- alle wireless Kommunikation wurde disabled (WLAN, Bluetooth)
- Autostart der App kurz nach dem Hochfahren (Task Scheduler)
- Beenden der App und Herunterfahren zu festgelegter Zeit (Task Scheduler)
- Auto-Start für nicht relevante Anwendungen deaktivieren
- alle Benachrichtigungen deaktivieren
- Power Settings:
 - 100% beste Leistung
 - kein automatischer Ruhezustand oder Ausschalten des Bildschirms
 - Festplatte ausschalten: Nie
 - Deaktivieren: Hybrid Standby, selektives USB Sparen, Fast Boot

Folgende Software wurde installiert und konfiguriert:

- aktuelle Treiber (für System Stabilität):
 - Motherboard Chipset Driver
 - Intel (LAN) Driver
 - Nvidia Studio Driver
- 7zip (effizienter Datentransfer)
- HWinfo64 (Überwachung der Hardware beim Testen)
- Kamera Software (LogiTune)
 - Benachrichtigungen deaktiviert
- Unity Hub, Login notwendig (Anpassen und Bauen der Anwendung)
 - install: Unity Version 2021.3.9.f1
 - select: Universal Windows Platform Build Support
 - select: Windows Build Support (IL2CPP)
 - select: Documentation
- Visual Studio 2022 (Anpassen von Code)
 - Setup über config File
- Chrome (Herunterladen zusätzlicher Software)
- Anydesk (Remote Zugang)

Anwendung

Zusammenfassung

Die Anwendung wurde mit der Unity Game Engine umgesetzt und für Windows 10 erstellt.

In beiden Kamerabildern des Users werden mithilfe eines Convolutional Neural Networks (CNN) die Bildkoordinaten von relevanten Feature-Points, wie z.B. Schulter, Hüfte, Hand und Fuß bestimmt.

Nach einer entsprechenden Kalibrierung der Kameras können dann aus diesen Bildkoordinaten für alle Feature-Points virtuelle 3D Koordinaten berechnet werden.

Mit diesen 3D Feature-Points wird ein virtueller Charakter konstruiert und animiert. Anhand der Bones dieses Charakters werden die 3D Modelle (Kostüme, Masken) animiert.

Schließlich werden das animierte Kostüm, das Kamerabild und zusätzliches UI zu einem finalen Bild kombiniert, welches auf dem Display gerendert wird. Weiterhin wird das Kostüm nach einer gewissen Zeit gewechselt.

Wenn längere Zeit kein User im Kamerabild erkannt wurde oder das Tracking fehlschlägt, wird der "Sleep Mode" aktiviert. Dabei wird das (aktive) Tracking ausgeschaltet und nur wenige Male pro Sekunde ein User im Kamerabild gesucht.

Einige dieser Schritte werden nachfolgend genauer erläutert.

Ein Großteil des Setups für das 2D Tracking kann einem Online Tutorial von Christian Mills entnommen werden: <https://christianjmills.com/posts/barracuda-posenet-tutorial-v2/part-1/>

Kamerabild Verarbeitung

Um das trainierte CNN nutzen zu können, müssen die Kamerabilder angepasst werden.

Zunächst müssen die Live-Feeds der beiden Kameras in RenderTextures kopiert werden ([1], [2]). Dabei kann auch die Orientierung des Bildes angepasst werden, über einen entsprechenden Image-Effect-Shader ([3]).

Die RenderTexture wird später in einem bestimmten Format (Auflösung, Color Range) benötigt, die Konvertierung der Auflösung erfolgt mit [3], für die neue Color Range wird ein Compute Shader benötigt.

Mit einem weiteren Image-Effect-Shader und [3] wird außerdem die Verzerrung des Kamera-Bildes kompensiert. Die Parameter dieses Shaders wurden zuvor speziell für die verwendete Kamera angepasst. Die Verschiebung der UV-Koordinaten wurde mit einer 2D Bezier-Surface (Abbildung von float2 → float2) realisiert. Die Parameter dieser Verschiebungen (5 x 7 Vektoren (2D)) wurden in einer kontrollierten Testumgebung bestimmt - sodass alle physisch geraden Linien in einer Szene auch in der finalen 2D Projektion (Bild der Szene) als gerade Linien erscheinen.

Dieser Schritt ist wichtig für das CNN und essentiell für eine exakte räumliche Kalibrierung der Kameras zueinander, bzw. zu deren virtuellen Gegenständen in der Engine.

Relevante Klassen und Funktionen:

- [1] UnityEngine.WebCamTexture → Constructor, Play()
- [2] UnityEngine.WebCamDevice.devices
- [3] UnityEngine.Graphics.Blit(...)

2D Feature Point Extraction

Die Bestimmung der Bildkoordinaten von Feature-Points des Users erfolgt mithilfe eines trainierten CNN, welches online unter einer MIT-Lizenz auf [GitHub](#) veröffentlicht wurde.

Das Modell kann unter Verwendung des Barracuda Packages in Unity verwendet werden. Dieses Package bietet verschiedene Möglichkeiten, Input an Neural Networks zu übergeben und deren Output zu verarbeiten, solange das Modell im .onnx Format vorliegt.

Um beide Kamerabilder parallel verarbeiten zu können, wurden zwei Instanzen verwendet. ([1], [2])

Es wurde die Verarbeitung auf der GPU [3] gewählt, da dies die beste Qualität und Performance ermöglicht.

Um die Framerate des Systems nicht zu beeinträchtigen, wurde eine asynchrone Methode gewählt, um die Resultate aus dem Speicher der GPU zu lesen [4]. Danach ist eine Synchronisation der beiden Instanzen notwendig.

Relevante Klassen und Funktionen:

- [1] Unity.Barracuda.Model
- [2] Unity.Barracuda.IWorker
- [3] Unity.Barracuda.WorkerFactory.Type.ComputePrecompiled
- [4] Unity.Barracuda.Tensor
 - Constructor (from RenderTexture)
 - capacity: Tensor.GetTensorDataStatistics().maxCapacity
 - Tensor.data.ScheduleAsyncDownload(capacity)

3D Feature Point Calculation

Liegen die 2D-Keypoints vor, können durch die virtuellen Positionen der beiden Kameras die 3D-Koordinaten der sichtbaren Features berechnet werden. (minimaler Abstand zweier Geraden)

Die virtuellen Positionen und Rotationen der Kameras müssen zuvor mit einer Kalibrierung bestimmt werden.

Dabei ist zu beachten, dass es bei mehreren Usern zu Problemen kommen kann, da nicht offensichtlich ist, welche 2D-Features zu welchem User gehören - insbesondere, da Features eines Users kurzzeitig auch verdeckt werden können.

Diese Probleme werden dadurch umgangen, indem direkt versucht wird, einen Körper mit "sinnvollen" Maßen zu erstellen.

Virtual Character Construction

Mit den 3D Feature-Points eines Users zu einem bestimmten Zeitpunkt kann dann über mehrere Frames hinweg ein virtueller Charakter erstellt und animiert werden.

Die Rotationen von Armen und Beinen stellten dabei eine große Herausforderung dar, da nur jeweils 3 Punkte als Referenz herangezogen werden können.

Der Kopf konnte auch nur schwer animiert werden, da die 5 Feature-Points nur sehr selten von beiden Kameras sichtbar sind.

Kostüm Animation

Die Bones der zu animierenden 3D Modelle werden in der Hierarchie als Children zu den Bones des virtuellen Charakters hinzugefügt.

Dabei werden die Bones des Rückens (Spine) gesondert berechnet.

Aus den 4 Feature-Points von Schulter (links, rechts) und Hüfte (links, rechts), wird eine Bilineare Fläche konstruiert. Innerhalb dieser Fläche können die Bones des Models über festgelegte "lokale Koordinaten" berechnet werden.

Anpassungen

Voraussetzung

Für jede Art von Änderung muss das Unity Projekt geöffnet werden, die Szene bzw. Scripte angepasst werden, und ein neuer Build des Projektes erstellt werden.
Dieser neue Build (.exe File) muss dann noch im Task-Scheduler referenziert werden, damit die Anwendung von Windows automatisch gestartet und beendet werden kann.

Kostüme

Neue Kostüme können auf die folgende Weise eingebaut werden.
Zunächst wird das 3D-Modell inklusive Animation-Rig (als .fbx) benötigt.
Dieses muss in Unity importiert werden und die Materials müssen angepasst werden.

Danach muss das Animation-Rig mit dem animierten Charakter verknüpft werden.
Dazu müssen die einzelnen Bones des Rigs an die entsprechende Stelle in der Transform-Hierarchy des Charakters gezogen werden.

Das finale Modell muss dann noch im Script "Compositing Controller" referenziert werden, und einem "Mode" zugewiesen werden.

Zusätzlich muss der Text für das UI Overlay definiert werden.
Dazu muss ein neues Scriptable Object vom Typ "Subtitle" erstellt werden und in der Hierarchie "Rendering/UI/Subtitles Canvas/Subtitle" dem Script "SubtitleDisplay" via Inspector hinzugefügt werden. Zusätzlich muss im Script "CostumeAnimation" ein neuer Type hinzugefügt werden.
Dieser Typ muss dann im neuen Subtitle-Asset ausgewählt werden.
Dort werden dann auch der Name für das Kostüm, das Stück und den Autoren/Komponisten des Stückes festgelegt.

Mögliche Verbesserungen

Verbesserungen sind in nahezu allen Teilen der Software möglich.
Es wird nachfolgend nur eine kleine Auswahl vorgestellt.

Visuelle Aspekte

Für realistisches Rendering kann zum einen das Lighting den realen Bedingungen in der Oper besser angepasst werden. Durch das Erstellen bzw. Messen einer HDRI (vor Ort) können Reflektionen deutlich realistischer berechnet werden.

Die Animation der Kostüme kann durch Cloth-Simulation auf ein völlig neues Level gebracht werden.

Das Kamerabild kann durch eine höhere Auflösung (4K statt 1080p) noch (etwas) besser aussehen. Beim Testen wurde versucht, beide Kameras mit dieser höheren Auflösung zu betreiben - dies führte zu einer starken Reduzierung der Framerate. Für ein besseres Kamerabild des Users muss jedoch nur eine Kamera mit 4K aufnehmen. Das sollte möglich sein, wurde jedoch nicht getestet.

Tracking

Zunächst kann die Tracking-Frequenz erhöht werden, indem z.B. zwei weitere Instanzen des Barracuda Modell Workers parallel arbeiten. Diese würden die aktuellen Kamera-Bilder auswerten, während auf die Ergebnisse der anderen Worker gewartet wird - sodass idealerweise keine Frames geskippt werden. Dieser Ansatz sorgt allerdings auch für eine deutlich höhere Auslastung der GPU.

Bei der Berechnung des virtuellen Charakters könnte ein physik-basiertes Modell für bessere Stabilität und plausiblere Resultate sorgen. Dabei würden dem Charakter Collider hinzugefügt, die nur bestimmte Posen zulassen und die Bewegungen würden durch eine dynamische Simulation der verknüpften Körperteile natürlicher aussehen.

Die Kalibrierung der Kameras kann mit einem anderen Ansatz mit mehr Parametern (z.B. FoV) deutlich verbessert werden. Beim Mapping von realen Punkten im Raum, zu virtuellen Punkten, konnte nur eine Genauigkeit von etwa 5 cm erreicht werden.

Durch Variieren der Kameraposition und -winkel könnte außerdem die Tracking Stabilität erhöht werden.

Das Face Tracking sollte am besten über ein separates neuronales Modell erfolgen. Mit nur fünf Referenzpunkten, die nur selten für beide Kameras (gleichzeitig) sichtbar sind, ist die Bestimmung von Position und Rotation des Kopfes sehr kompliziert und ungenau.

Fehlerbehebung

Wichtige Hinweise zum Umgang mit der Hardware

Generell ist beim Umgang mit Stromkabeln Vorsicht geboten. Bei Änderungen der Kabelkonfiguration sollte zuvor möglichst der gesamte Strom für das System abgeschaltet werden.

Zusätzlich sollte beim Umgang mit dem PC darauf geachtet werden, dass es nicht zu statischen Entladungen kommt, da diese die Hardware beschädigen können.

Beim Umgang mit Datenkabeln ist zu beachten, dass die Verbindung zum PC bzw. Kamera oder Display immer nur bei ausgeschaltetem PC getrennt/verbunden werden sollte. Gleichzeitig sollte darauf geachtet werden, dass die Trennung/Verbindung des Kabels mit den entsprechenden Buchsen ordnungsgemäß erfolgt - also ohne übermäßigen Krafteinsatz, immer mit gerader Bewegung und immer vollständig.

Es wird empfohlen, Fotos von den verwendeten Buchsen zu machen, bevor Kabel entfernt werden - da die Kabelkonfiguration nicht geändert werden darf. Jedes Kabel muss nach der Trennung (und vor dem nächsten Betrieb) wieder exakt so, wie vorher angeschlossen sein.

Schritte zur Fehlerbehebung

Nachfolgend werden die am wahrscheinlichsten auftretenden Probleme aufgeführt, mögliche Ursachen aufgelistet und die Schritte zu deren Behebung erläutert.

Die Schritte sind dabei in der vorgegebenen Reihenfolge zu prüfen und abzuarbeiten. Ausnahme besteht bei offensichtlichen Auffälligkeiten - diese können auch als erstes überprüft werden.

A) Der Rechner startet nicht / Das Display bleibt schwarz.

- PC Neustart
- Ist das Display eingeschaltet?
 - Das Display hat einen Schalter für die Stromzufuhr und einen Knopf zum Ein- bzw. Ausschalten. Deren Positionen sind in der Bedienungsanleitung (z.B. online) zu finden.
- Sind alle Kabel ordnungsgemäß angeschlossen? (gerade, vollständig)
 - Strom: PC, Display
 - HDMI Kabel (PC - Display)
 - USB Kabel (PC - Kameras)
- Scheint sich viel Staub auf bzw. im PC zu sein?
 - Der Staub sollte entfernt werden. (-> [Wartung](#))
 - Ggf. kann ein Staubfilter installiert werden, um das Problem zukünftig zu reduzieren.

- Genaue Inspektion des PCs
 - nach Absprache mit Verantwortlichem bei expanding Focus
 - idealerweise Vor Ort Termin
 - Sind Auffälligkeiten zu erkennen?
 - Sind alle internen Kabel ordnungsgemäß angeschlossen?
 - Untersuchung der einzelnen Komponenten des PCs
 - CPU neu einbauen, neue Wärmeleitpaste
 - ...

B) Die Anwendung startet nicht.

- PC Neustart
- Unterbricht ein Windows-Dialogfenster den Betrieb?
 - 1. Versuch: Neustart
 - 2. Dialogfenster schließen, dann Neustart
- Ist der Task Scheduler korrekt eingerichtet?
 - Neueinrichtung über Remote Zugriff (Kontakt: expanding focus)

C) Die Anwendung läuft fehlerhaft: kein oder schlechtes Tracking des Users.

- Sind die Linsen der Kameras sauber?
- Sind die USB-Kabel der Kameras richtig angeschlossen? (an PC und Kamera)

Diagnose innerhalb der Anwendung:

- Peripherie anschließen: (mindestens) Maus
- in der laufenden Anwendung: Mouse-Click aktiviert/deaktiviert den Debug-Modus
 - Es sollten links bzw. rechts die Kamerabilder der entsprechenden Kameras angezeigt werden. Darunter sind Debug Bilder, in denen das aktuelle Tracking überprüft werden kann.
 - 1) eine Seite zeigt kein Bild an
 - Kamera Kabel überprüfen
 - 2) virtueller Charakter kann nicht konstruiert werden
 - kurzes Video aufnehmen: mit aktivem Debug Mode, während der Spiegel genutzt wird (User versucht getrackt zu werden)
 - Video an expanding Focus per Email senden
 - Best Case 1: die Anwendung muss neu gebaut werden
 - Remote Zugriff auf PC notwendig
 - Worst Case 2: das Kamerasystem muss neu kalibriert werden
 - Option 1: Vor Ort Termin
 - Option 2: Remote Kalibrierung
 - erfordert (einmalig) Entwicklung einer neuen Möglichkeit zur Kalibrierung (ca. eine Woche Arbeit)
 - Die Remote Kalibrierung kann dann innerhalb weniger Stunden (~2) durchgeführt werden.