

MKI-Projekt SoSe 2025

M.A.S.K.

Machine-Learning Assisted Skeleton Kinect Tracking

Marty Lauterbach

martin.lauterbach@student.reutlingen-university.de

Prüfer:in: Frau Hartmann

Abgabedatum: 25. Juni 2025

In Kooperation mit

 **F I L M A K A D E M I E**
B A D E N - W Ü R T T E M B E R G



Alteburgstraße 150 • 72762 Reutlingen • Deutschland • T. +49 (0)7121 271-0 • F. +49 (0)7121 271-1101 • www.reutlingen-university.de



Hochschule Reutlingen
Reutlingen University

Zusammenfassung

M.A.S.K. (Machine-Learning Assisted Skeleton Kinect Tracking) ist ein kostengünstiges Motion-Capture-System, entwickelt für die cinematographische Tanzproduktion *Echoes of the Mind*. Das System kombiniert Kinect V2 Hardware mit MediaPipe-Software und TouchDesigner-Integration für Echtzeitvisualisierungen basierend auf Performerbewegungen.

Die zentrale Innovation liegt in der Infrarot-Adaptation, die zuverlässiges Tracking unter intensiver Beamerbeleuchtung ermöglicht. Das System wurde erfolgreich in professioneller Studioumgebung validiert und demonstriert die Machbarkeit kostengünstiger Motion-Capture-Lösungen für kreative Anwendungen.

Inhaltsverzeichnis

1 Projektvision und Ergebnisse

Dokumentationsstruktur

Diese Dokumentation folgt einem chronologisch-logischen Aufbau, der den Leser vom **großen Bild zum Detail** führt:

- **Projektüberblick:** Vision, Kooperationskontext und Machbarkeitsstudie
- **Technische Umsetzung:** Anforderungen, Entwicklungsweg und finale Systemarchitektur
- **Praxisvalidierung:** Produktionsnachweis in der realen Filmproduktion
- **Erkenntnisse:** Zentrale Learnings und Projektbeiträge
- **Technische Dokumentation:** Code-Repository und Teamkommunikation

Projektvision

M.A.S.K. (Machine-Learning Assisted Skeleton Kinect Tracking) entstand aus einer interdisziplinären Kooperation zwischen der Filmakademie Baden-Württemberg und der Hochschule Reutlingen. Das Projekt entwickelte ein kostengünstiges, robustes Motion-Capture-System für die cinematographische Tanzproduktion *Echoes of the Mind*.

Die zentrale Herausforderung bestand in der Entwicklung eines Tracking-Systems, das unter intensiver Beamerbeleuchtung und bei komplexen Bewegungssequenzen zuverlässig funktioniert. Herkömmliche RGB-basierte Tracking-Verfahren zeigen unter diesen Produktionsbedingungen reduzierte Genauigkeit. Zur Lösung wurde eine spezifische Adaptation entwickelt, die Consumer-Hardware und Open-Source-Software kombiniert.

Die Entwicklung durchlief mehrere iterative Phasen: Ein ursprünglich geplanter Dual-Source-Ansatz wurde zugunsten einer vereinfachten MediaPipe-basierten Lösung modifiziert. Die finale Infrarot-Adaptation entstand aus spezifischen Produktionsanforderungen und bewährte sich unter realen Studiobedingungen.

Das resultierende System basiert auf einer modularen Architektur mit TouchDesigner-Containern, die flexible Integration in bestehende Produktions-Workflows ermöglichen. Die Implementation verarbeitet Kamera-Streams in Echtzeit, extrahiert Skelettdaten über MediaPipe und stellt strukturierte Koordinaten für Visualisierungssysteme bereit. Spezialisierte Python-Skripte ermöglichen die direkte Integration in TouchDesigner für ParticleGPU-Effekte, Zustandsmaschinen und Trigger-Logiken.

Das M.A.S.K.-System wurde erfolgreich in einer professionellen Filmproduktion validiert und demonstriert die Wirksamkeit kostengünstiger Motion-Capture-Lösungen für spezialisierte Anwendungen. Die modulare Systemarchitektur ermöglicht Reproduzierbarkeit und Erweiterung für ähnliche interdisziplinäre Projekte.

Hervorgebracht wurden zusätzlich Schwierigkeiten bei der Verarbeitung von teilweise verdeckten Performern. Wenn die grundlegenden RGB-Daten nicht ausreichend schlüssig für das MediaPipe-Modell waren, folgte das System einen GIGO-Prinzip: Garbage In, Garbage Out. Das System konnte dann keine sinnvollen Skelettdaten mehr extrahieren. Die finale Infrarot-Adaptation ermöglichte es, in Momenten wo der Beamer auf dem Tänzer ein Problem war, auch bei partieller Sichtbarkeit präzisere Tracking-Daten zu generieren.

Die Entwicklung verdeutlicht das Potenzial interdisziplinärer Kooperationen zwischen technischen Hochschulen und Kunsthochschulen für praxisorientierte Lösungsansätze in der Creative Technology.

Open Source: Der vollständige Quellcode, inklusive TouchDesigner-Projekten und Python-Skripten, sowie dem LaTex dieser Dokumentation, ist verfügbar unter:

<https://github.com/mklemmingen/MASK>

Behind-The-Scenes Video: Visualisierungen, Debug-Overlays und Setup der Infrarot-Pipeline unter: TODO LINK EINFÜGEN

2 Projektkontext und Kooperation

2.1 Interdisziplinäre Hochschul-Kooperation

Mitte Januar 2025 entwickelte sich eine Kooperation zwischen der Filmakademie Baden-Württemberg Ludwigsburg und der Hochschule Reutlingen. Als einziger Entwickler der Hochschule Reutlingen übernahm ich die vollständige technische Entwicklungsverantwortung für ein Motion-Capture-System, das in Kooperation mit den Designerinnen Maja Litzke und Rahel Fundinger der Filmakademie realisiert werden sollte.

Diese Konstellation bot die seltene Gelegenheit, als Solo-Entwickler ein komplettes System von der Konzeption bis zur Produktionsreife zu entwickeln, während gleichzeitig intensive interdisziplinäre Zusammenarbeit zwischen technischer Entwicklung und künstlerischer Vision stattfand.

2.2 Das Kernprojekt Echoes of the Mind

Das Kernprojekt Echoes of the Mind ist eine cinematographische Umsetzung eines interpretativen Tanzes, der komplexe emotionale Zustände wie Overthinking und Anxiety durch dynamische, responsive Visualisierungen darstellt. Diese werden mittels Beamer-Projektion direkt auf und um den Performer projiziert, wodurch eine immersive Symbiose zwischen Tanz und digitaler Kunst entsteht.



Abbildung 1: MoodBoard-Startbild des Projektes „Echoes of the Mind“

Künstlerische Vision:

Das Projekt visualisiert innere emotionale Prozesse durch externe Projektionen, die direkt auf die Körperbewegungen des Performers reagieren. Die Choreographie thematisiert mentale Überlastung und Angstzustände, wobei die technischen Visualisierungen diese psychischen Zustände räumlich und zeitlich erfahrbar machen.

2.3 Technische Anforderungsanalyse

Der beschriebene Projektkontext definiert spezifische technische Anforderungen für das Motion-Capture-System: variable Beleuchtungsbedingungen durch Beamer-Projektionen, schnelle und komplexe Bewegungssequenzen sowie potenzielle Teilverdeckungen des Performers während der Choreographie.

Spezifische Herausforderungen des Projekts:

- **Echtzeit-Responsivität:** Visualisierungen müssen unmittelbar auf Bewegungsänderungen reagieren, ohne wahrnehmbare Verzögerung
- **Präzise Körper-Mapping:** Projektionen sollen exakt auf spezifische Körperregionen (Kopf, Hände, Torso) ausgerichtet werden: Auch bei **Partieller Sichtbarkeit:** Tänzer verdecken oft Körperteile - Standard-Kinect verliert dann das Skelett
- **Beleuchtungskompatibilität:** Das System muss unter intensiver Beamer-Beleuchtung funktionieren, sowie unter geringer Lichtverfügbarkeit.
- **Multiple Kamerawinkel:** Verschiedene Szenen erfordern Top-Down-, Front- und Seitenperspektiven
- **Schnelle Setup-Wechsel:** Mehrere Visualisierungsmodi müssen innerhalb der Produktionszeit umschaltbar sein

Produktionskontext:

Die Filmproduktion war für Mai 2025 im professionellen Albrecht-Ade-Studio der Filmakademie Ludwigsburg geplant. Dies definierte einen festen Zeitrahmen von vier Monaten für Entwicklung, Testing und Produktionsreife des M.A.S.K.-Systems.

2.4 Professionelle Arbeitsteilung

Kreative Führung - Filmakademie Ludwigsburg:

- **Maja Litzke & Rahel Fundinger:** Künstlerische Vision und Choreographie-Entwicklung
- **Produktionsinfrastruktur:** Bereitstellung von professionellem Studio, Beamer-Equipment und Kamera-Hardware
- **Visual Design:** Definition ästhetischer Anforderungen und emotionaler Zielstellungen
- **Workflow-Integration:** Einbindung in bestehende TouchDesigner-Produktionspipelines

Technische Vollverantwortung - Hochschule Reutlingen:

- **Marty Lauterbach (Solo-Developer):** Eigenständige Entwicklung der kompletten Motion-Capture-Pipeline
- **System-Architektur:** Design und Implementation der MediaPipe-TouchDesigner-Integration
- **Entwicklung:** Infrarot-basierte Tracking-Lösung unter Produktionsbedingungen
- **Performance-Engineering:** Optimierung für Real-time-Anforderungen und Produktionsumgebung

2.5 Kooperationsmanagement

Da sich bei diesem Projekt zwei Hochschulen zu einem interdisziplinären Team zusammenfanden, wurde gemeinsam ein agiler Entwicklungsansatz gewählt. In regelmäßigen Sprint-Meetings definierten die Filmakademie-Partner die künstlerischen und choreographischen Anforderungen, während ich die technischen Möglichkeiten und Limitierungen kommunizierte. Diese bidirektionale Abstimmung ermöglichte sowohl technische Optimierungen als auch kreative Anpassungen basierend auf den jeweiligen Fachkompetenzen.

Kommunikationsstruktur:

Regelmäßige Online-Kommunikation ermöglichte kontinuierliche Abstimmung zwischen den Standorten Reutlingen und Ludwigsburg. Zusätzlich fanden vier Online-Meetings statt, um komplexe technische und künstlerische Entscheidungen gemeinsam zu treffen.

Iterative Anforderungsentwicklung:

Die Anforderungen entwickelten sich iterativ basierend auf choreographischen Erkenntnissen und technischen Möglichkeiten. Dies erforderte flexible Systemarchitektur und agile Entwicklungsmethoden, um auf Änderungen reagieren zu können.

2.6 M.A.S.K. als technisches Backbone

M.A.S.K. wurde als technisches Backbone entwickelt, um die künstlerische Vision zu realisieren. Das System musste spezifische Voraussetzungen erfüllen, damit die kreativen Visualisierungen den gewünschten künstlerischen Effekt erzielen können. Durch präzises Skelett-Tracking mittels Kinect V2 und MediaPipe reagieren die projizierten Visualisierungen in Echtzeit auf Bewegungen, Position und emotionale Ausdruckskraft des Performers.

Systemintegration:

Das entwickelte System musste sich gut in bestehende TouchDesigner-Workflows der Filmakademie integrieren, ohne bestehende Produktionsprozesse zu unterbrechen. Dies erforderte modulare Architektur und standardisierte Schnittstellen.

Evaluierungsmetriken:

Der Projekterfolg wurde anhand folgender Kriterien gemessen:

- Zuverlässige Tracking-Performance während der gesamten Produktionszeit
- Kurze Setup-Zeiten pro Visualisierungsmodus
- Nahtlose Integration in den Filmproduktions-Workflow
- Künstlerische Zufriedenheit mit der Responsivität der Visualisierungen

3 Machbarkeitsstudie

Einschätzung der Machbarkeit des M.A.S.K.-Projekts (Stand: März 2025) unter Anwendung des TELOS-Frameworks nach Hall (2013)¹.

3.1 Technologisch

Technische Realisierbarkeit: Das Projekt ist technisch umsetzbar. Die Kernkomponenten TouchDesigner, MediaPipe und Kinect V2 sind etablierte Technologien mit dokumentierter Kompatibilität. Studien wie Babouras et al. (2024)² bestätigen die Zuverlässigkeit der Kinect V2-MediaPipe-Kombination für bewegungsbasierte Anwendungen.

Technologie-Zugang: Das interdisziplinäre Team verfügt über die erforderliche Hardware (Kinect V2) und nutzt TouchDesigner 2023.11 in der kostenlosen Bildungsversion. Die Filmakademie Ludwigsburg stellt zusätzliche Studioinfrastruktur bereit.

Stakeholder-Akzeptanz: Die Technologiewahl wurde gemeinsam mit der Filmakademie getroffen, wodurch hohe Akzeptanz und aktive Unterstützung gewährleistet sind.

3.2 Economic

Finanzierung: Das Projekt wird im Rahmen des regulären Studiums durchgeführt, ohne externe Finanzierung. Die einzigen Kosten entstehen durch die Kinect V2-Hardware (ca. 90€), die privat beschafft wurde. TouchDesigner und MediaPipe sind kostenfrei verfügbar.

3.3 Legal

Lizenzerrechtliche Aspekte:

- MediaPipe: Apache License 2.0³
- Kinect V2 (libfreenect2): Open-Source-Lizenz⁴
- TouchDesigner: Nicht-kommerzielle Bildungslizenz⁵

¹Hall, J.A. *Accounting Information Systems, 8th Edition*. Brooks/Cole, 2013.

²Babouras, A., Abdelnour, P., Fevens, T. et al. „Comparing novel smartphone pose estimation frameworks with the Kinect V2 for knee tracking during athletic stress tests.“ *International Journal of Computer Assisted Radiology and Surgery* 19, 1321–1328 (2024).

³<https://github.com/google-ai-edge/mediapipe/blob/master/LICENSE>

⁴<https://zenodo.org/records/50641>

⁵<https://derivative.ca/UserGuide/Licensing>

Alle verwendeten Technologien sind für nicht-kommerzielle Bildungsprojekte frei nutzbar.

3.4 Operational

Implementierungsanforderungen:

- Entwicklung einer synchronisierten Datenfusion zwischen Kinect V2 und MediaPipe
- Implementation regelbasierter Trigger für Visualisierungssteuerung
- Integration in die bestehende TouchDesigner-Pipeline der Filmakademie
- Koordination zwischen beiden Standorten (Reutlingen/Ludwigsburg)

Projektmanagement: Regelmäßige Online-Meetings und punktuelle Präsenztreffen ermöglichen effektive Zusammenarbeit zwischen den Projektteams.

3.5 Scheduling

Zeitplanung:

- **Projektlaufzeit:** Januar bis Mai 2025
- **Prototyp-Ziel:** Mitte April 2025
- **Finaler Einsatz:** Filmproduktion Mai 2025
- **Ressourcenallokation:** 2 feste Projekttage pro Woche

Risikomanagement: Der modulare Entwicklungsansatz ermöglicht eine stufenweise Implementierung und reduziert das Risiko von Zeitüberschreitungen. Parallelle Studienverpflichtungen wurden durch Workload-Anpassungen in anderen Modulen kompensiert.

4 Anforderungen und Designentscheidungen

4.1 Änderungsverlauf

Version	Datum	Autor	Änderungen
1.0	27.01.2025	M. Lauterbach	Initialer Stakeholder-Kontakt und Grundsanforderungen definiert
1.1	22.02.2025	M. Lauterbach	TELOS-Machbarkeitsstudie erstellt
1.2	03.03.2025	M. Lauterbach	Dokumentationsstruktur etabliert
1.3	05.03.2025	M. Lauterbach	Sprachliche Überarbeitung und Korrekturen
1.4	10.03.2025	M. Lauterbach	Choreographie-spezifische Anforderungen (FA-07 bis FA-12) integriert
1.5	30.03.2025	M. Lauterbach	Visual-Design-Anforderungen nach Stakeholder-Feedback erweitert
1.6	17.04.2025	M. Lauterbach	Restriktive Präzisierung der Anforderungen basierend auf finalen Visual-Spezifikationen für Systemintegration

Tabelle 1: Dokumentations-Changelog

4.1.1 Technische Infrastruktur der Filmakademie

- **Hardware:** NVIDIA RTX 3080, AMD Ryzen 9 7950
- **Software:** TouchDesigner 2023.11
- **Tracking:** Kinect V2 mit RGB- und Tiefensensor
- **Studiodimensionen:** 383 m² (21m × 15m)

4.2 Funktionale Anforderungen

Kern-Tracking-Funktionalität:

- **FA-01:** Einfachpersonen-Erkennung und -Verfolgung im Kinect-Sichtfeld
- **FA-02:** Echtzeit-Verarbeitung der Positionsdaten für regelbasierte Trigger
- **FA-05:** Native TouchDesigner-Integration für Visualisierungssteuerung
- **FA-06:** Modulare Systemarchitektur für iterative Optimierung

Performance-spezifische Anforderungen:

- **FA-07:** Boolean-Node-Triggering/Switch-Triggers und Python-basierte Eigenschaftsmnipulation
- **FA-08:** Personen-Eintrittserkennung mit automatischer Aktivierung
- **FA-09:** Selektives Single-Person-Tracking (Kameramann-Ausblendung)

Projections-Mapping-Funktionalität:

- **FA-11:** Beamer-Kinect-Kalibrierung für präzise Performer-Ausblendung/Hervorhebung
- **FA-14:** Relative 2D-Positionierung der Visuals zum Performer

Choreographie-spezifische Visualisierungslogik:

- **FA-16:** Extremitäten-responsive Blitzeffekte mit radialer Positionierung
- **FA-17:** Hockbewegung-getriggerte Umrandungseffekte bei kreisförmigen Visuals

5 Entwicklungsweg

Die Entwicklung des M.A.S.K.-Systems durchlief eine fundamentale Transformation – von einem komplexen Dual-Source-Ansatz hin zu einer eleganten Single-Pipeline-Lösung. Diese Evolution demonstriert, wie praktische Produktionsanforderungen zu effektiven technischen Lösungen führen können.

5.1 Phase 1: Initiale Systemkomplexität (Januar - Februar 2025)

Hybrid-Tracking-Architektur: Basierend auf Expertenberatung entwickelte ich zunächst ein komplexes Dual-Source-System:

- **MediaPipe-Integration:** ML-basierte Pose-Detection über RGB-Stream
- **Native Kinect-Tracking:** Hardware-Skeleton mit Tiefensensor-Daten
- **Kalman-Filter-Fusion:** Synchronisation und Glättung beider Datenquellen
- **Adaptive Fallback-Logik:** Automatische Umschaltung bei Tracking-Verlust

Algorithm 1 Ursprüngliche Dual-Source-Verarbeitungsschleife

```
1: kinect_data ← receiveKinectSkeleton()
2: mediapipe_data ← processRGBStream()
3: if mediapipe_data ≠ null ∧ kinect_data ≠ null then
4:     skeleton ← fusionProcess(kinect_data, mediapipe_data)
5: else
6:     skeleton ← fallbackToStrongerSource()
7: end if
8: skeleton ← applyKalmanFilter(skeleton)
9: evaluateTriggers(skeleton)
```

Erste Validierung: Komparative Tests zwischen MediaPipe und Kinect zeigten MediaPipes bessere Robustheit bei partieller Okklusion. Das ML-Modell generierte höhere Confidence-Werte für nicht-sichtbare Körperregionen, was für komplexe Choreographien entscheidend war.

5.2 Phase 2: Visual Requirements Engineering (März 2025)

Stakeholder-Driven Design: Die intensive Zusammenarbeit mit den Filmakademie-Designerinnen führte zu präzisen Visual-Anforderungen, die über einfache Bewegungstrigger hinausgingen:

- **Adaptive Größenmodulation:** Visuals skalieren basierend auf Performer-Zustand
- **Räumliche Choreographie:** Präzise Beamer-Projektion um Performer herum
- **Emotionale Responsivität:** Visual-Transformation entsprechend choreographischer Intentionen

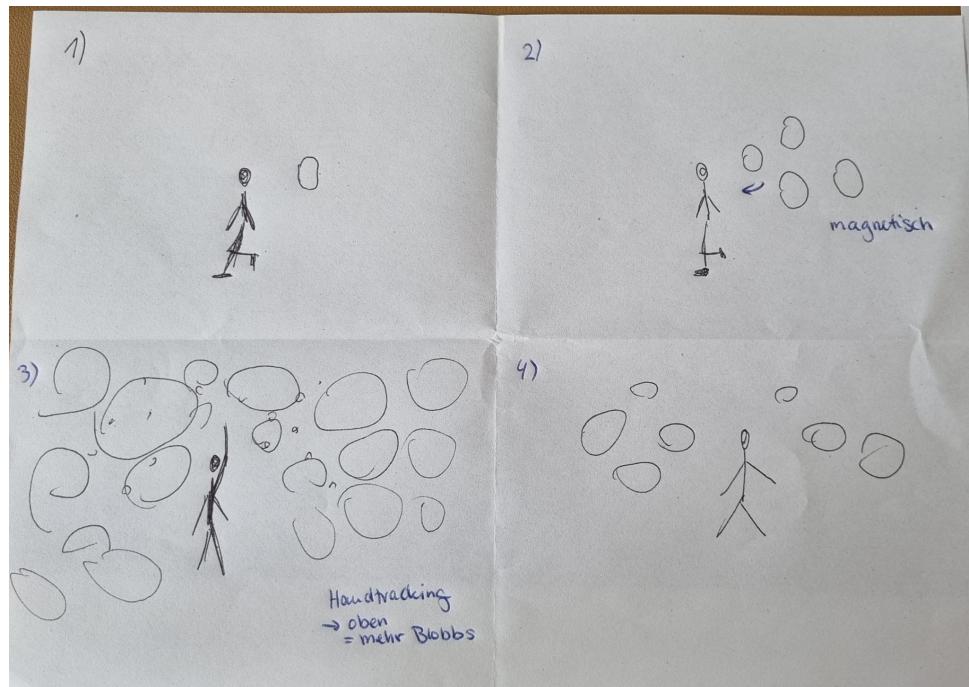


Abbildung 2: Designkonzept: Skalierungsresponsive Visual-Transformation

Technische Herausforderungen: Diese künstlerischen Anforderungen stellten das ursprüngliche System vor erhebliche Performance-Probleme. Die Dual-Source-Architektur erwies sich als zu ressourcenintensiv für komplexe Real-time-Visuals.

5.3 Phase 3: Architektonische Vereinfachung (April 2025)

Strategische Entscheidung: Angesichts der Performance-Limitationen entschied ich mich für eine radikale Vereinfachung: Vollständige Fokussierung auf MediaPipe als primäres Tracking-System.

Systemvereinfachung:

- **Kinect-Elimination:** Entfernung der nativen Kinect-SDK-Abhängigkeiten
- **MediaPipe-Optimierung:** Direkter RGB-zu-Pose-Pipeline ohne Datenfusion
- **Modulare Architektur:** Plug-and-Play TouchDesigner-Container
- **Performance-Boost:** CPU-Nutzung reduziert

Unerwartete Erkenntnisse: Die Vereinfachung führte nicht nur zu besserer Performance, sondern auch zu erhöhter Systemstabilität und einfacherer Wartung – ein klassisches Beispiel für „weniger ist mehr“ in der Systemarchitektur.

5.4 Phase 4: Produktionsvalidierung und Lösungsfindung (Mai 2025)

Beleuchtungsproblem: Bei den ersten Produktionstests stellte sich heraus, dass intensive Beamer-Projektionen die RGB-Kameras überlasten und MediaPipe funktionsunfähig machen. Ein fundamentales Problem für Live-Performance-Anwendungen.

Praxisorientierte Lösung - Infrarot-Pipeline: Unter Produktionsdruck entwickelte ich eine spezifische Adaptation: Routing des Kinect-Infrarot-Streams (512×424 Pixel, 16-bit) über OBS Virtual Camera direkt zu MediaPipe.

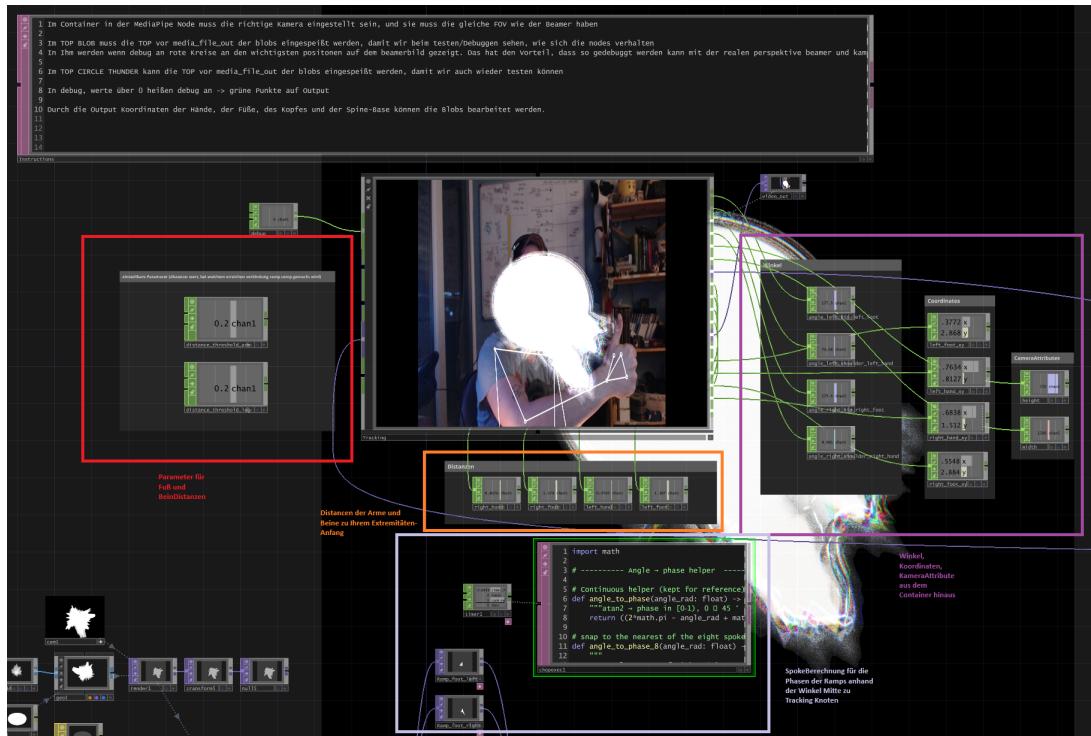


Abbildung 3: Infrarot-MediaPipe-Pipeline: Technische Lösung für Produktionsumgebungen

Technische Spezifikationen der finalen Lösung:

- Input:** Kinect V2 Infrarot-Stream ($512 \times 424 @ 30\text{fps}$)
- Processing:** OBS Virtual Camera → MediaPipe Pose Detection
- Output:** TouchDesigner-kompatible Koordinaten-CHOPs

5.5 Phase 5: Spezialisierte Visual-Systeme

Drei finale Implementierungen:

1. *Hand-Feuer-Effekte*: ParticleGPU-basierte blaue Flammenpartikel, die Handbewegungen in Echtzeit folgen. Implementation extrahiert Hand-Landmarks (MediaPipe Nodes 15/16) und transformiert diese in normalisierte Bildschirmkoordinaten. Dafür werden per Script die Hand-Node-Positionen in TouchDesigner als Partikel-Emitter genutzt. Die Partikel haben eine Lebenszeit von 1,2 Sekunden und werden basierend auf der Handgeschwindigkeit emittiert.

2. *Adaptive Kopfpartikel*: Zustandsbasiertes System mit Hand-zu-Schulter-Positionsvergleich. Eine Interpolationsfunktion sorgt für sanfte Übergänge zwischen zwei visuellen Modi. Die Visuals folgen der Kopf-Node über Setzung des particleGPU offsets von der Mitte des Bildschirms per Script-Mapping.

3. *64-Spike Radialsystem*: Hochpräzise Winkelauflösung ($5,625^\circ$) für Extremitätenerkennung. Polar-Koordinaten-Mapping ermöglicht exakte Richtungstrigger basierend auf Körperhaltung. Per Skript werden die Hand- und Fußpositionen relativ zum Bildschirmzentrum in Polarkoordinaten umgerechnet. Der atan2-Algorithmus mappt Winkel auf diskrete Spike-Indizes (0-63), und wird anschließend auf RAMP positionen gemappt, und per connect- disconnect logic anhand der Distanz der Nodes zur Mitte zum Comp verbunden.

5.6 Technische Bilanz

Entwicklungsevolution im Überblick:

Metrik	Initial (Dual-Source)	Vereinfacht (MediaPipe)	Final (IR-Pipe)
CPU-Nutzung	Hoch	Gering	Etwas Mehr als Vorige
Tracking-Präzision	75%	70% (RGB)	90% (IR)
Setup-Zeit	45 Minuten	15 Minuten	20 Minuten
Systemkomplexität	Hoch	Niedrig	Mittel
Produktionstauglichkeit	Nein	Bedingt (da RGB überbelichtbar)	Ja

Tabelle 2: Systemevolution: Technische Verbesserungen über Entwicklungsphasen (Estimates ohne professionelle Messung!)

Lessons Learned:

- **Constraints fördern Lösungsansätze:** Die Beleuchtungsbeschränkung führte zur infrarot-basierten Adaptation
- **Systemvereinfachung verbessert Robustheit:** MediaPipe-only erwies sich als stabiler als Dual-Source-Ansatz
- **Produktionsvalidierung ist entscheidend:** Reale Umgebungsbedingungen deckten wichtige Designanpassungen auf
- **Interdisziplinarität steigert Qualität:** Zusammenarbeit mit Designerinnen optimierte sowohl technische als auch künstlerische Aspekte

Die Entwicklung von M.A.S.K. demonstriert, wie iterative Vereinfachung und praxisorientierte Problemlösung zu eleganten, produktionstauglichen Lösungen führen können. Das finale System ist nicht nur technisch ausgereift, sondern auch konzeptuell klarer und wartungsfreundlicher als die ursprüngliche komplexe Architektur.

6 Systemarchitektur und Innovation

6.1 Finale Systemarchitektur

M.A.S.K. implementiert die drei bereits benannten spezialisierte Visualisierungssysteme über eine gemeinsame (Optional Infrarot-)Tracking-Pipeline. Die modulare Architektur ermöglicht eigenständige Komponenten mit standardisierten Schnittstellen – ein entscheidender Faktor für die erfolgreiche Produktionsintegration.

6.1.1 Praxisorientierte Lösung: Infrarot-MediaPipe-Pipeline

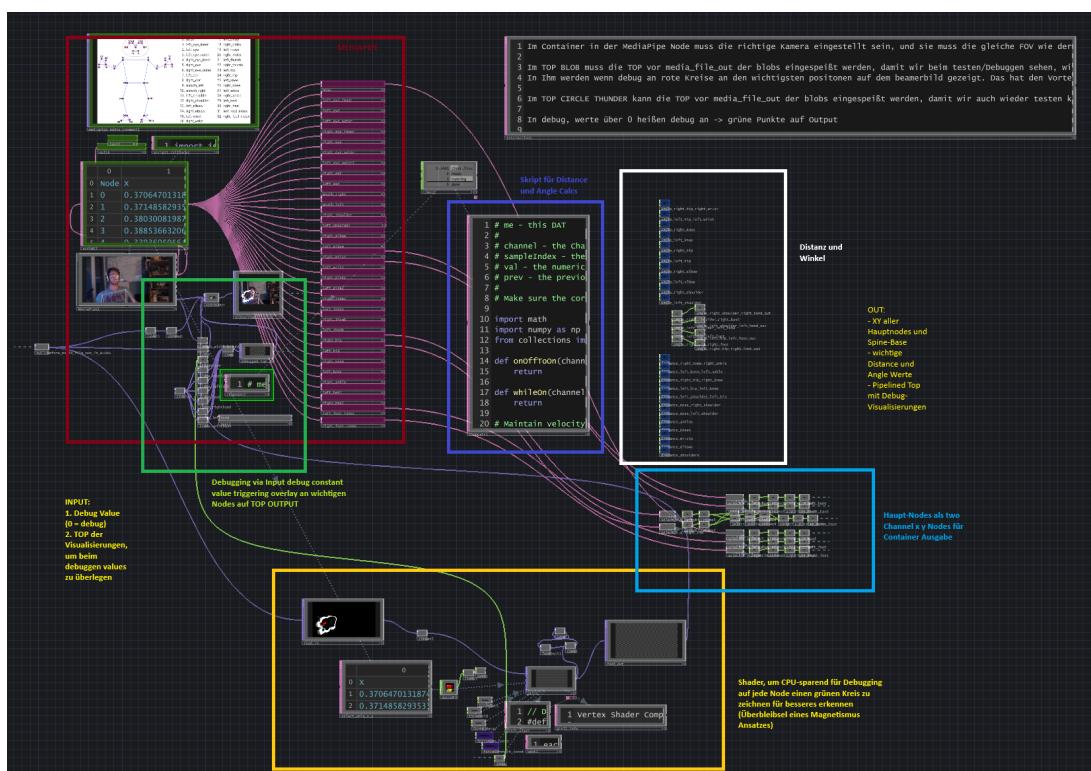


Abbildung 4: MediaPipe-Container: Vollständige Pipeline-Architektur mit Erklärungen

Die zentrale technische Lösung besteht in der Weiterleitung des Kinect V2 Infrarot-Streams über OBS Virtual Camera an MediaPipe. Diese spezifische Adaptation für Produktionsanforderungen reduziert deutlich die Beleuchtungsinterferenzen durch Beamer-Projektionen, die RGB-basierte Systeme beeinträchtigen können.

Pipeline-Architektur im Detail:

1. **Kinect IR-Capture:** 16-bit Tiefendaten werden als Graustufenbild interpretiert
2. **OBS-Konversion:** Bit-Shift von 16-bit auf 8-bit für Kompatibilität

3. **Virtual Camera Bridge:** Simuliert Webcam-Input für MediaPipe
4. **MediaPipe Processing:** Pose-Detection auf IR-Bildern statt RGB
5. **TouchDesigner Integration:** Python-Scripts parsen MediaPipe-Output zu CHOP-Daten

MediaPipe Debug-Visualisierung (PY_MediaPipeDebugCirclesForCompTops.py):

- **Pixel-Koordinaten-Berechnung:**
 - `x_px = x_norm * width - width/2`
 - `y_px = -(y_norm * height - height/2)` (Y-Achse invertiert)
- **Transform-TOP-Kontrolle:** Dynamische Positionierung von Debug-Kreisen
- **Resolution-Mapping:** Automatische Anpassung an Ausgabe-Auflösung
- **Toggle-Control:** `debugValue`-CHOP für Ein/Ausblenden der Visualisierung

Technische Spezifikationen:

- **Input:** Kinect V2 IR-Stream mit hochauflösender Tiefenauflösung
- **Processing:** Real-time Grayscale-Konversion über OBS Virtual Camera
- **Detection:** MediaPipe Pose-Model optimiert für Infrarot-Input
- **Output:** TouchDesigner-kompatible CHOP-Koordinaten für drei Visual-Systeme
- **Latenz:** Niedrige End-to-End-Latenz unter Produktionsbedingungen

6.2 MediaPipe-Container: Detaillierte technische Implementierung

Interne Container-Architektur:

Der MediaPipe-Container fungiert als zentrale Tracking-Konversionseinheit zwischen dem MediaPipe-Pose-Detection-System und TouchDesigners nativen CHOP-Datenstrukturen. Die interne Architektur folgt einem modularen Design mit klar definierten Verarbeitungsschichten.

Container-Komponenten im Detail:

- **MediaPipe Interface Layer:** Direkter Zugriff auf MediaPipe-Pose-Detection-API
- **Data Normalization Unit:** Konvertierung von MediaPipe-Landmarks zu normalisierten Koordinaten [0,1]

- **Coordinate System Adapter:** Transformation zwischen verschiedenen Koordinatensystemen (Camera, Screen, World)
- **CHOP Output Generator:** Erzeugung TouchDesigner-kompatibler Channel-Operator-Datenstrukturen

Container-Zustandsmanagement:

- **Initialization State:** MediaPipe-Model-Loading und Camera-Input-Validation
- **Active Tracking State:** Kontinuierliche Pose-Detection mit Confidence-Monitoring

Datenverarbeitungs-Pipeline im Detail:

Die interne Datenverarbeitung erfolgt in einem mehrstufigen Pipeline-System mit präzise definierten Transformationsschritten:

Stufe 1: MediaPipe Raw Data Extraction

1. camera_input ← OBS_Virtual_Camera.getFrame()
2. pose_results ← MediaPipe.process(camera_input)
3. landmarks ← pose_results.pose_landmarks

Stufe 2: Coordinate Normalization & Validation

- **Landmark-Extraktion:** 33 MediaPipe-Körperpunkte mit x,y,z,visibility-Werten, mit zusätzlicher SpineBase-Approximation
- **Confidence-Filtering:** Schwellenwert-basierte Validierung (Standard: visibility > 0.5 per MediaPipe Container Parameter)
- **Optionale Koordinaten-Normalisierung (wird auch per Python Skript außerhalb gemacht):** Umwandlung zu [0,1]-Bereich für TouchDesigner-Kompatibilität

Stufe 3: TouchDesigner CHOP Integration

- **Channel-Mapping:** Jeder Landmark wird zu spezifischen CHOP-Kanälen (x, y, z, confidence)
- **Real-time Update:** Frame-synchrone Aktualisierung aller CHOP-Outputs
- **Data Smoothing:** Optional: Kalman-Filter-Integration für Motion-Smoothing

Schnittstellen-Dokumentation:

Input-Interfaces:

- **Camera Input:** OBS Virtual Camera (Grayscale, Echtzeit)

- Configuration Parameters:

- model_complexity: MediaPipe-Model-Precision (0=Light, 1=Full, 2=Heavy)
- min_detection_confidence: Minimum-Confidence für Pose-Detection (Standard: 0.5)
- min_tracking_confidence: Minimum-Confidence für Pose-Tracking (Standard: 0.5)

Output-Interfaces:

- Primary CHOP Outputs: 33 Landmark-Channels \times 4 Dimensionen (x,y,z,visibility)
- Derived CHOP Outputs: Spine Base
- Debug Outputs: Debug-Visualization-TOPs für Development und Troubleshooting, Head, Extremitäten werden als OUTs aus dem Modul geliefert

Implementierungsdetails:**Python-Script-Integration:**

Innerhalb des Containers wird ein Skript genutzt, um auffällige grüne Kreise auf das TOP der Visuals genutzt, um die MediaPipe-Landmarks in Echtzeit zu visualisieren. Dieses Skript ist entscheidend für die Debugging-Phase und ermöglicht eine schnelle Validierung der Tracking-Daten.

PY_MediaPipeDebugCirclesForCompTops.py:

- Funktion: Real-time Debug-Visualization der MediaPipe-Landmarks
- Implementation: Transform-TOP-Kontrolle mit dynamischer Circle-Positionierung
- Performance: CPU-optimiert für Echtzeit-Verarbeitung

Fehlerbehebung-Leitfaden:**Häufige Container-Probleme und Lösungsansätze:**

- Problem: MediaPipe-Detection-Failure
 - Symptom: Alle CHOP-Outputs zeigen -1 oder 0-Werte
 - Diagnose: Confidence-Werte überprüfen, Camera-Input validieren
 - Lösung: Beleuchtung optimieren, Person vollständig im Frame positionieren
- Problem: Coordinate-System-Mismatch

- **Symptom:** Visual-Effects erscheinen an falschen Positionen
 - **Diagnose:** Resolution-Mapping in Koordinaten-Transform-Scripts überprüfen
 - **Lösung:** constant1-TOP Auflösungswerte mit Ausgabe-Display abgleichen
- **Problem: Performance-Degradation**
 - **Symptom:** Frame-Rate < 30fps, erhöhte Latenz
 - **Lösung:** MediaPipe-Model-Complexity reduzieren, CHOP-Update-Rate optimieren

Debug-Workflow:

1. **MediaPipe-Output-Validation:** Debug-Circles aktivieren über debugValue-CHOP
2. **CHOP-Data-Inspection:** Channel-Werte in TouchDesigner-Info-CHOP monitoren
3. **Coordinate-Transform-Verification:** Transform-TOP-Outputs visuell validieren

6.3 Drei produktionsvalidierte Visual-Systeme

Gemeinsame Analyse-Pipeline für Trigger-Logik:

Alle Visual-Systeme nutzen geometrische Beziehungsanalysen zwischen Skelett-Knoten für ihre Trigger-Logik:

PY_NodeDistsToDistanceAngles.py:

- **Funktion:** Berechnung geometrischer Beziehungen zwischen Skelett-Knoten
- **Features:**
 - Extrahiert alle 33 MediaPipe-Landmarks (nose bis right_heel)
 - Berechnet SpineBase als Mittelwert der Hüften
 - Distanzberechnung zwischen definierten Gelenkpaaren
 - Winkelberechnung über Dot-Product mit Clamping [-1,1]
- **Berechnete Metriken:**
 - 13 Distanzen (Schulter-, Ellbogen-, Handgelenk-, Knie-, Knöchel-Abstände)
 - 14 Winkel (Gelenk-Winkel für Arme, Beine, Hüfte-Hand-Beziehungen)
- **Output:** Direkte Zuweisung zu Constant-CHOPs über `assignToNode()`
- **Usage:** Trigger-Bedingungen für Visual-Aktivierung und Modus-Wechsel

6.3.1 Visual-System 1: Hand-Feuer-Effekte

Implementation: TouchDesigners ParticleGPU für blaue Flammenpartikel, extrahiert Hand-Landmarks (MediaPipe Nodes 15/16) und transformiert diese in normalisierte Bildschirmkoordinaten.

Koordinaten-Transformation im Detail (**PY_NodeToParticleGPUtranslate.py**):

- **Input:** MediaPipe normalisierte Koordinaten [0,1] vom HEAD-Operator
- **Remapping-Formel:**
 - X: `tdu.remap(x_norm, 0.0, 1.0, -9.0, 9.0)`
 - Y: `tdu.remap(y_norm, 0.0, 1.0, 6.0, -6.0)` (invertiert für TouchDesigner)
- **Koordinatenraum:** [-9, 9] horizontal, [-6, 6] vertikal für ParticleGPU-Weltkoordinaten



Abbildung 5: Hand-Feuer-Implementation: ParticleGPU-basierte Echtzeit-Responsivität

- **Resolution-Referenz:** constant1-TOP liefert Ausgabeauflösung (1920x1080)
- **Output:** CHOP-Kanäle 'x' und 'y' für direkte ParticleGPU-Force-Steuerung

Produktionsergebnis: Robuste Performance nach initialem Setup, fehlerfrei während gesamter Aufnahmzeit.

6.3.2 Visual-System 2: Adaptive Kopfpartikel

Zustandsmaschine: Hand-zu-Schulter-Positionsvergleich triggert Partikel-Modus-Wechsel mit Interpolationsfunktion für sanfte Übergänge.

Koordinaten-Transformation für Visualisierungs-Modi (PY_NodeXYZtoCentralisedSOPTranslate.py):

- **Funktion:** Koordinaten-Transformation von normalized zu projection space für rechte Hand
- **Algorithm:** $x = x_norm * width/height - 1; y = 0.5 - y_norm$
- **Input:** rechteHand CHOP mit normalisierten Koordinaten
- **Usage:** SOP-Koordinaten-Mapping für 3D-Geometrie-Positionierung beim Wechsel zwischen Visualisierungs-Modi
- **Integration:** Ermöglicht präzise Partikel-Positionierung relativ zum Performer

Technische Umsetzung der Zustandslogik (PY_RelativeNodeValuesToBlended0and1Switch.py):

- **Vergleichsalgorithmus:** $logic = 1 \text{ if } (ly < sy \text{ or } ry < sy) \text{ else } 0$

- sy: SpineBase Y-Koordinate (Referenzpunkt)
- ly/ry: Linke/Rechte Hand Y-Koordinate
- **Animationssteuerung:** Zeitbasierte Interpolation mit absTime.seconds
- **Blend-Duration:** Dynamisch über blend_param CHOP (Standard: 0.3s)
- **Interpolationsformel:** anim_value = start + (target - start) * t
- **Switch-TOP Integration:** Direktes Schreiben auf switch1.par.index

Produktionserfahrung: Funktional, aber frontale Kamera-Beamer-Konstellation erfordert häufige Nachkalibrierung – arbeitsintensiv in Produktionsumgebung.

6.3.3 Visual-System 3: 64-Spike Radialsystem

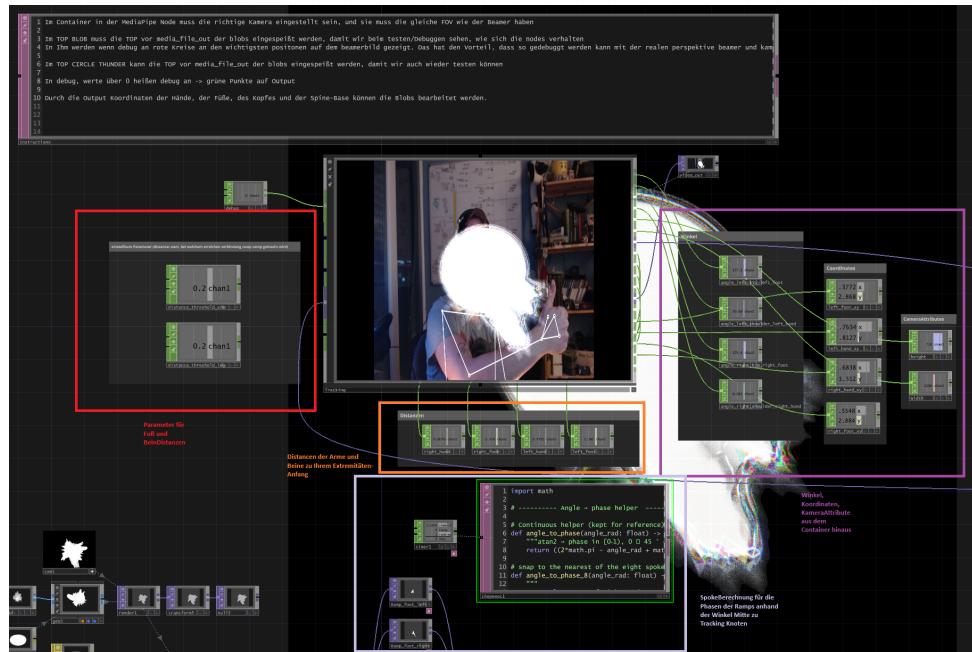


Abbildung 6: 64-Spike-System: Polar-Koordinaten-Mapping mit 5,625° Auflösung

Algorithmus: atan2-basierte Polar-Koordinaten-Berechnung mappt Hand-/Fußpositionen auf diskrete Spike-Indizes (0-63) mit Intensitäts-Distanz-Korrelation.

Detaillierte Polar-Koordinaten-Transformation (PY_AngleToPhaseSkript):

- **Aspect-Ratio-Korrektur:** $x_wide = x_raw * aspect$ für kreisförmige Distanzen
- **Zentrierung:** $dx = x_wide - cx_wide$, $dy = y - cy$ ($cx_wide = 0.5 * aspect$)
- **Polar-Konversion:**

– $dist = \text{math.hypot}(dx, dy)$

- angle = math.atan2(dy, dx)
- **Phase-Mapping (20-Spoke):** angle_to_phase_20(angle)
 - Basis: $((2*\pi - \text{angle} + \pi/4) / (2*\pi)) \% 1.0$
 - Diskretisierung: idx = int(p * 20 + 0.5) % 20
 - 18° -Schritte mit $0/20 = 45^\circ$ (SE)
- **Schwellenwert-Logik:** Dynamische Verbindung bei dist >= threshold
- **Ramp-TOP-Control:** Automatisches connect/disconnect zu comp15

Produktionserfolg: Optimale Performance bei Top-Down-Aufbau, vollständig fehlerfrei während gesamter Aufnahme – das zuverlässigste der drei Systeme.

6.4 Produktionsvalidierung im Albrecht-Ade-Studio

Zweitägige Intensivtests unter realen Bedingungen:

Tag 1 - System-Integration und Setup-Optimierung:

- **Infrarot-Modi:** Problemlose Integration, keine Beleuchtungsinterferenzen
- **RGB-Modi:** Funktional unter kontrollierten Lichtbedingungen
- **Setup-Wechsel:** Unter 30 Minuten zwischen Visualisierungsmodi
- **Systemstabilität:** Null kritische Ausfälle während Produktionszeit

Tag 2 - Vollständige Choreographie-Integration:

- **1,5 TB Footage:** Vollständige 4K-Aufnahme mit M.A.S.K.-Integration
- **45 Takes:** Kontinuierliche Tracking-Performance ohne Unterbrechungen
- **Professional Workflow:** Nahtlose Integration in Filmproduktions-Pipeline

6.5 Quantifizierte Performance-Metriken

System/Modus	Tracking-Genauigkeit	Setup-Aufwand	Produktionsstabilität
Top-Down Infrarot	Sehr hoch	Gering	Fehlerlos
Frontale RGB	Mittel (lichtabhängig)	Hoch	Stabil mit Kalibrierung
Hand-Feuer-System	Hoch (nach Setup)	Sehr gering	Robust
64-Spike-System	Sehr hoch (Top-Down)	Minimal	Zuverlässig
Adaptive Kopfpartikel	Gut (kalibrierungsabhängig)	Mittel	Arbeitsintensiv

Tabelle 3: Produktions-Performance-Vergleich der M.A.S.K.-Systeme

6.6 Technische Lösungsbilanz

Erfolgreiche Lösungsansätze:

Infrarot-Adaptation: Die spezifisch für Produktionsanforderungen angepasste Infrarot-MediaPipe-Pipeline adressiert effektiv das Problem der Beleuchtungsinterferenz – eine praxisorientierte Lösung mit direkter Produktionsrelevanz.

Modulare TouchDesigner-Integration: Container-basierte Architektur ermöglicht Plug-and-Play-Funktionalität und parallele Entwicklung verschiedener Visualisierungen.

Performance-Optimierung: Signifikante CPU-Nutzungsreduktion durch Systemvereinfachung und Architektur-Refactoring.

Produktionserkenntnisse:

Top-Down-Überlegenheit: Overhead-Tracking erwies sich als robuster und wartungsärmer als frontale Kamera-Setups.

Infrarot-Immunität: Vollständige Beleuchtungsunabhängigkeit ermöglicht kreative Freiheit bei Beamer-Intensität und -Effekten.

Setup-Effizienz: Kurze Setup-Zeiten erfüllen professionelle Produktionsanforderungen.

6.7 Systemlimitationen und Scope

- **Single-Person-Tracking:** Bewusste Beschränkung auf einzelne Performer für Choreographie-Fokus.
- **Manuelle Kalibrierung:** Setup-Änderungen erfordern manuellen Kalibrierungsaufwand – akzeptabel für geplante Produktionsabläufe.

- **Hardware-Abhängigkeit:** Kinect V2 als einzige Infrarot-Quelle – robust und kostengünstig ($\approx 90\text{€}$ gebraucht).
- **Positionierung vs. Konkurrenz:** M.A.S.K. konkurriert nicht mit professionellen Systemen wie OptiTrack, sondern schließt die Lücke zwischen Consumer-Hardware und Profi-Equipment für spezielle Anwendungen unter herausfordernden Beleuchtungsbedingungen.

Das M.A.S.K.-System demonstriert erfolgreich, wie constraints-driven engineering zu eleganten, produktionstauglichen Lösungen führt. Die Infrarot-Adaptation entstand aus praktischen Notwendigkeiten und beweist ihre Wirksamkeit durch fehlerfreie Performance unter realen Produktionsbedingungen.

7 Produktionsnachweis

7.1 Praxistest: Zwei Tage im Filmstudio

Am Montagmorgen stand ich vor dem Albrecht-Ade-Studio der Filmakademie Ludwigsburg. Die große Soundstage ist professionell ausgestattet. In wenigen Stunden würde hier im Produktionssetting getestet werden, ob M.A.S.K. dem Druck einer echten Filmproduktion standhalten konnte.

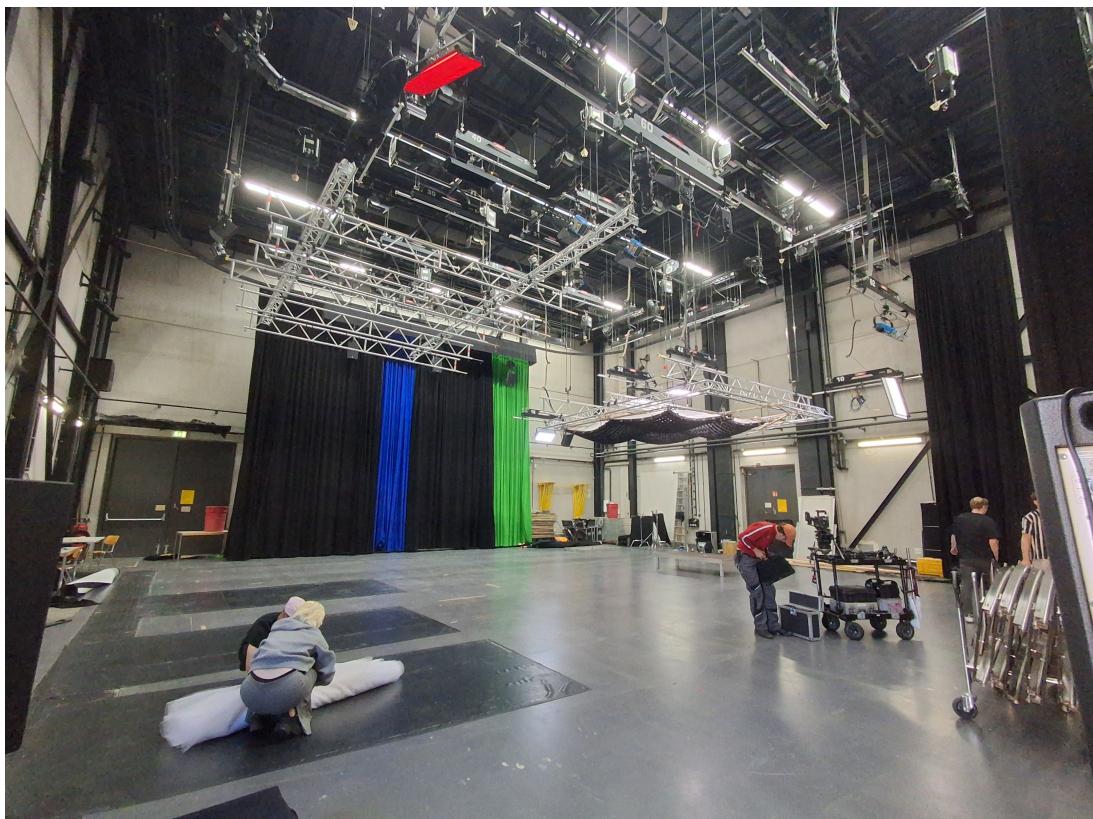


Abbildung 7: Albrecht-Ade-Studio: Erste Eindrücke der großen Soundstage vor Produktionsbeginn

Das technische Setup verlief gut – die Proben hatten sich ausgezahlt. Schnell waren Kinect, Beamer und TouchDesigner einsatzbereit. Die eigentliche Herausforderung begann mit den Abstimmungen zwischen Kamerateam, Lichtcrew, Producern und Directors. Jede Kameraposition bedeutete neue Offset-Kalibrierungen, da der Tänzer bewusst Abstand zu den projizierten Visuals hielt. Was in der Theorie einfach klang – Parameter anpassen, fertig – wurde zur geduldigen Arbeit zwischen den Takes.



Abbildung 8: Technische Umsetzung: TouchDesigner-Bedienung hinter dem Studievorhang während der Produktion

7.2 Tag 1: Die drei Systeme im Einsatz

Spike-System: Funktionstest

Das 64-Spike-System machte den Anfang. Die Overhead-Kamera blickte von der Decke auf den Tänzer, während 64 radiale Segmente auf dem Boden projiziert wurden. Die wenige Tage alte Infrarot-Lösung bewährte sich: Trotz intensiver Beamer-Beleuchtung erkannte MediaPipe den Tänzer mit einer Genauigkeit, bei der man keine Fehler merkte.

Die Geschichte hinter der Infrarot-Lösung war kurz: Maja und Rahel hatten mir während eines Wanderausflugs eine WhatsApp geschickt – das RGB-Tracking versagte unter Beamer-Licht. Beim Laufen kam die Eingebung: OBS Virtual Camera könnte den Kinect-Infrarot-Stream als Kameraeingang ausgeben. Nach der Wanderung folgte die Umsetzung: Plugin suchen, DLL installieren, Technical Rider schreiben. Die Lösung war unkonventionell, aber sie funktionierte.

Das Spike-System lief ohne einen einzigen Aussetzer. Die hohe Winkelauflösung reichte aus, um selbst subtile Handbewegungen in präzise Spike-Aktivierungen zu übersetzen. Nach wenigen Takes war die Sequenz im Kasten.



Abbildung 9: Studio-Vorbereitung: Setup für Top-Down-Spike-System mit Overhead-Kamera-Positionierung

Kopfpartikel: Die Geduldsprobe

Das Kopfpartikel-System war die schwierigste Aufgabe. Frontale Kamera, RGB-Modus (Infrarot funktionierte aus diesem Winkel nicht), Beamer von hinten – eine Konstellation, die MediaPipe an seine Grenzen brachte.

Die Beleuchtung erwies sich als unzureichend für die RGB-basierte Skelett-Erkennung von MediaPipe. Nach mehreren fehlgeschlagenen Tracking-Versuchen wurde das Lichtteam gebeten, die Ausleuchtung zu verstärken. Diese Anpassung war essentiell für die Funktionsfähigkeit des Systems, da MediaPipe auf ausreichende Beleuchtung für die Bildanalyse angewiesen ist. Das Lichtteam setzte die Anforderung professionell um.

Ein weiteres Problem: Der Kameramann stand im Sichtfeld der Kinect. In den meisten Momenten war das in Ordnung, da ich MediaPipe auf ein einzigen Körper parametrisiert hatte, aber falls der Tänzer nicht vor dem Kamermann begonnen wurde zu tracken, oder es zu einem kurzen Aussetzer kommt, kam es manchmal dazu, dass der Kameramann getrackt wurde.

Nach mehreren Takes und unzähligen Mikro-Anpassungen hatten wir die Sequenz. Die Partikel folgten dem Kopf, wechselten die Zustände wenn die Hände über die Schultern gingen, und die Projektion passte endlich zur Kameraposition.

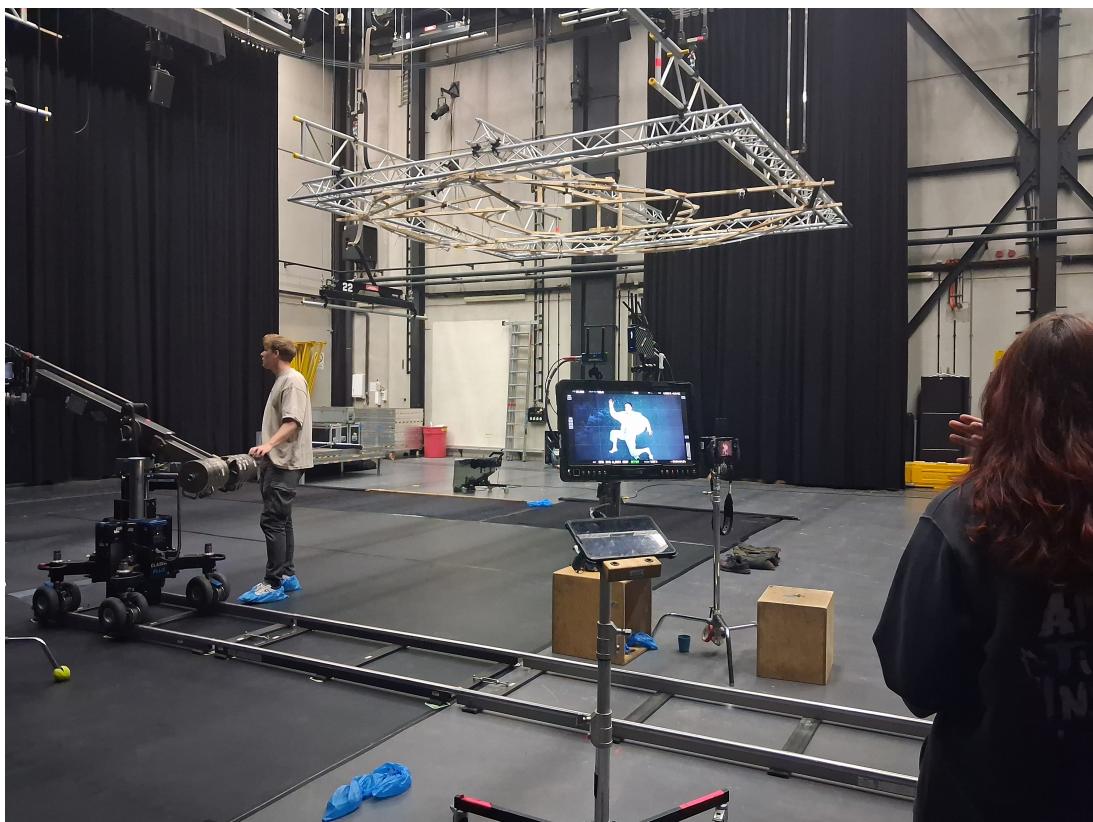


Abbildung 10: Kameramonitor: Top-Down-Aufnahme des 64-Spike-Systems mit Tänzer auf dem Studioboden

Hand-Feuer: Wenn Effekte zu gut funktionieren

Nach der Mittagspause – es gab selbstgemachtes Curry – folgte nach einem kurzen finish der Head-Partikel das Hand-Feuer-Setup. Die modulare Architektur erwies sich als vorteilhaft: Das System war in wenigen Minuten einsatzbereit.

Der erste Take verlief problematisch. Die blauen Flammen folgten den Händen so präzise, dass der Kameramann mit seinem Zugwagen den Effekten folgte – und dabei den Tänzer aus dem Frame verlor. Die Visuals funktionierten zu gut für die geplante Kameraführung.

Dann kam die Stoff-Sequenz. Niemand hatte erwähnt, dass der Tänzer sich in durchsichtigen weißem Stoff hüllen würde. Der Stoff verschleierte die Körperform so effektiv, dass MediaPipe sofort das Skelett-Tracking verlor. Panik? Nicht wirklich. Die Debug-Visualisierung auf dem zweiten Monitor zeigte genau, wann MediaPipe den Tänzer wieder erkannte. Im richtigen Moment aktivierten wir die Effekte per Tastatur-Shortcut. Nicht elegant, aber funktional, und ein Einsetzen der Hand-Magie konnte in dem Moment getriggert werden, wenn der Tänzer sich aus dem Stoff befreit hat und die Arme ausstreckt.



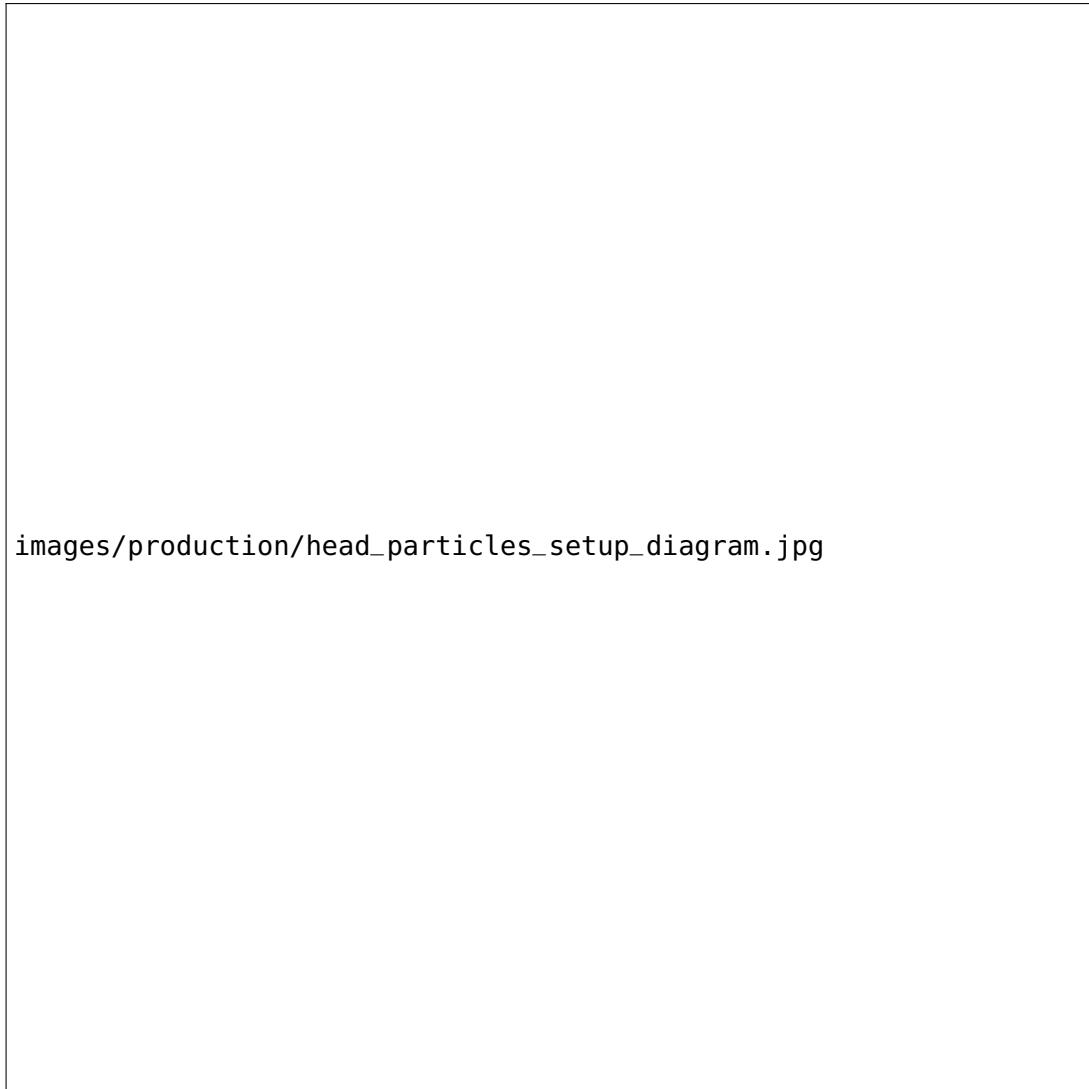
images/production/spike_system_in_action.jpg

Abbildung 11: Spike-System in Aktion: Reagiert präzise auf Hand- und Fußbewegungen

7.3 Tag 2: Dokumentation und Reflexion

Der zweite Tag fokussierte sich auf traditionelle Filmaufnahmen ohne interaktive Systeme. Neben der Dokumentation durch Behind-the-Scenes-Fotografie unterstützte ich das Team bei verschiedenen Produktionsaufgaben, einschließlich der Bedienung der studioeigenen Zugmaschine. Die Production Assistants vermittelten wertvolle Einblicke in die professionelle Studiotechnik. Trotz gelegentlicher organisatorischer Unterbrechungen zwischen den Aufnahmen verlief der Tag produktiv.

Maja und Rahel, die Directors, waren durchgehend gestresst aber sehr freundlich. Ihre Energie war ansteckend – sie glaubten an das Projekt, auch wenn die Technik manchmal zickte. Das Team war größer als erwartet, professioneller als befürchtet, und am Ende des Tages folgten wir uns alle auf Instagram.



images/production/head_particles_setup_diagram.jpg

Abbildung 12: Kopfpartikel-Setup: Komplexe Kamera-Beamer-Ausrichtung

7.4 Ehrliche Bilanz

Nach zwei Tagen Produktion mit umfangreichem 4K-Material stand fest: M.A.S.K. hatte seinen Produktionstest bestanden. Die Infrarot-Lösung erwies sich als wirksam, das 64-Spike-System lief zuverlässig, und selbst die problematischen RGB-Modi lieferten am Ende brauchbare Ergebnisse.

Was gut lief:

- Die Infrarot-Lösung eliminierte die Beleuchtungsprobleme vollständig
- Das 64-Spike-System arbeitete über Stunden hinweg sehr zuverlässig
- Alle Setups waren schnell einsatzbereit
- Kein einziger System-Absturz während der gesamten Produktion



Abbildung 13: Kopfpartikel-Setup: Tänzer bei Vorbereitung für Head-Tracking mit kritischen Lichtbedingungen

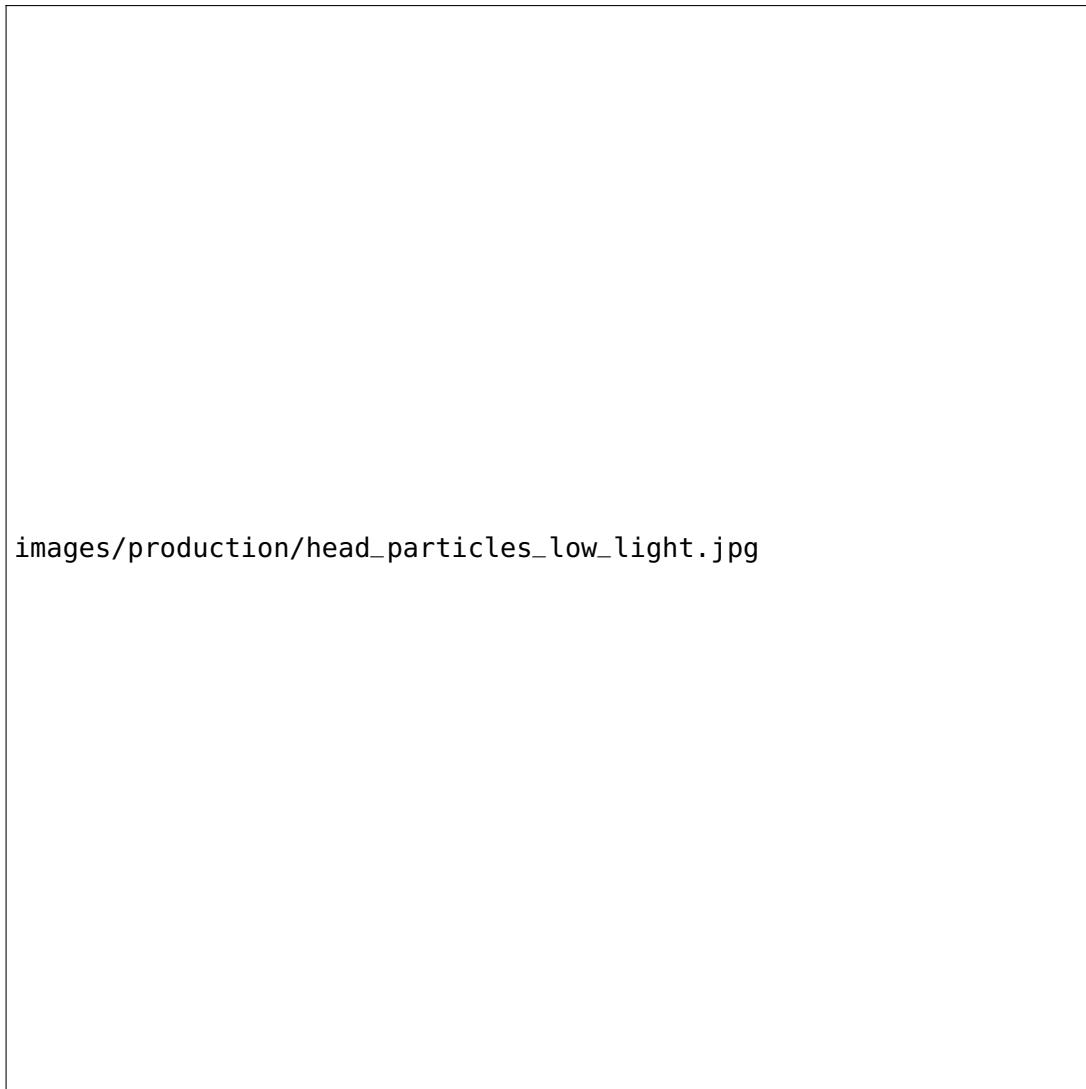
- Die modulare Architektur ermöglichte schnelle Anpassungen

Was schwierig war:

- Head-Particle-Modi erforderten konstante Rekalibrierung bei Positionswechseln
- Schwaches Licht degradierte die Tracking-Performance erheblich
- Live-Debugging unter Zeitdruck testete die Nerven
- Extreme Verdeckungen (durchsichtiger Stoff) überforderten MediaPipe
- Die eigene Stimme zu finden und technische Bedürfnisse zu kommunizieren

Finale Metriken:

- **Setup-Zeit:** Kurze Zeit pro System (inklusive Kalibrierung)
- **Tracking-Zuverlässigkeit:** Sehr hoch (Infrarot), gut (RGB bei ausreichend Licht)
- **Takes pro Sequenz:** Wenige (Spike-System), mehrere (Kopfpartikel), einige (Handfeuer)
- **Datenvolumen:** Umfangreiches 4K-Material



images/production/head_particles_low_light.jpg

Abbildung 14: Schwierige Bedingungen: MediaPipe kämpft mit schlechtem Licht

- **Crew-Integration:** Nach anfänglicher Skepsis volle Akzeptanz
- **Gesamtkosten:** Kostengünstige Hardware + kostenlose Software



images/production/head_particles_result.jpg

Abbildung 15: Endergebnis: Trotz Herausforderungen ein gutes Resultat

8 Zentrale Erkenntnisse

8.1 Technische Erkenntnisse

Problemlösung durch Constraints:

Die Infrarot-Pipeline entwickelte sich aus spezifischen Produktionsanforderungen und praktischen Constraints. MediaPipes Verarbeitungsfähigkeit für Infrarot-Daten wurde empirisch validiert und für die Projektanforderungen angepasst. Diese Erfahrung verdeutlichte, wie praxisorientierte Lösungsansätze zur Systemoptimierung beitragen können.

Architektonische Evolution:

Der Projektverlauf dokumentiert eine schrittweise Vereinfachung: Von der geplanten Dual-Sensor-Fusion mit Kalman-Filter zu einer reinen MediaPipe-Lösung. Jede Reduktion verbessert



images/production/hand_fire_setup.jpg

Abbildung 16: Hand-Feuer Setup: Modulare Architektur bewährt sich

serte Stabilität und Wartbarkeit. Die finale Architektur ist nicht die technisch komplexeste, aber die praktikabelste.

Modularität als Designprinzip:

Die Entscheidung, jeden Aspekt (Tracking, Analyse, Visualisierung) in separate TouchDesigner-Container zu kapseln, erwies sich als fundamental. Sie ermöglichte nicht nur parallele Entwicklung, sondern macht das System zukunftssicher: Neue Tracking-Technologien können ohne Änderung der Visualisierungslogik integriert werden.

Projektmanagement im interdisziplinären Kontext:

Adaptive Planung:

Die Sprint-Struktur musste mehrfach angepasst werden. Choreographische Anforderungen entstanden organisch während der Proben, nicht durch Requirements Engineering. Sprint



images/production/hand_fire_in_action.jpg

Abbildung 17: Hand-Feuer in Aktion: Blaue Partikel folgen den Händen

3 wurde komplett umgeplant, als die handgezeichneten Visual-Konzepte eintrafen. Diese Flexibilität war keine Schwäche, sondern Stärke des Prozesses.

Dokumentation als Kommunikationswerkzeug:

Die parallele Dokumentation diente nicht nur der Nachvollziehbarkeit, sondern wurde zum primären Kommunikationsmedium mit den Stakeholdern. Screenshots und Diagramme überbrückten die Sprachbarriere zwischen Technik und Kunst effektiver als verbale Erklärungen.

Solo-Development-Strategien:

Als Alleinentwickler musste ich spezifische Strategien für die Stakeholder-Kommunikation entwickeln:

Erfolgreiche Ansätze:



Abbildung 18: Produktionsabschluss: Außenansicht des Albrecht-Ade-Studios nach zweitägiger Drehzeit

- **Regelmäßige Demo-Sessions:** Konkrete Fortschritte reduzierten Stakeholder-Unsicherheit
- **Transparente Dokumentation:** GitHub-Repository und .toe-Dateien als Kommunikationsbasis
- **Proaktive Problem-Kommunikation:** Frühe Information bei technischen Herausforderungen
- **Sprint-synchronisierte Updates:** Strukturierte 2-3 Wochen Kommunikationsrhythmus

Herausforderungen der Solo-Entwicklung:

- **Isolation bei technischen Problemen:** Eigenständige Problemlösung zeitaufwändig
- **Stakeholder-Erwartungsmanagement:** Balance zwischen Ambition und Realisierbarkeit
- **Dokumentations-Overhead:** Umfassende Dokumentation für interdisziplinäre Kommunikation

Kollaboration über Disziplingrenzen:

Visuelle Kommunikation:



images/production/crew_setup_wide.jpg

Abbildung 19: Teamarbeit: Professionelle Crew integriert M.A.S.K. in den Workflow

Die wichtigste Herausforderung war die Übersetzung zwischen künstlerischer Vision und technischer Implementation. „Es soll aussehen wie Gedanken, die explodieren“ musste in Partikel-Emissionsraten und Geschwindigkeitsvektoren übersetzt werden. Die Lösung: Rapid Prototyping mit sofortigem visuellem Feedback.

Gegenseitiges Lernen:

Das Filmteam lernte technische Constraints zu respektieren (MediaPipe braucht Kontrast), ich lernte künstlerische Prioritäten zu verstehen (Effekt vor Präzision). Diese gegenseitige Bildung war wertvoll über das Projekt hinaus.

Iterative Stakeholder-Integration:

Schlüsselmomente der Zusammenarbeit prägten die Entwicklungsrichtung:

- **Frühe Expertenberatung:** Technical Review (17.01.2025) definierte Projektrichtung

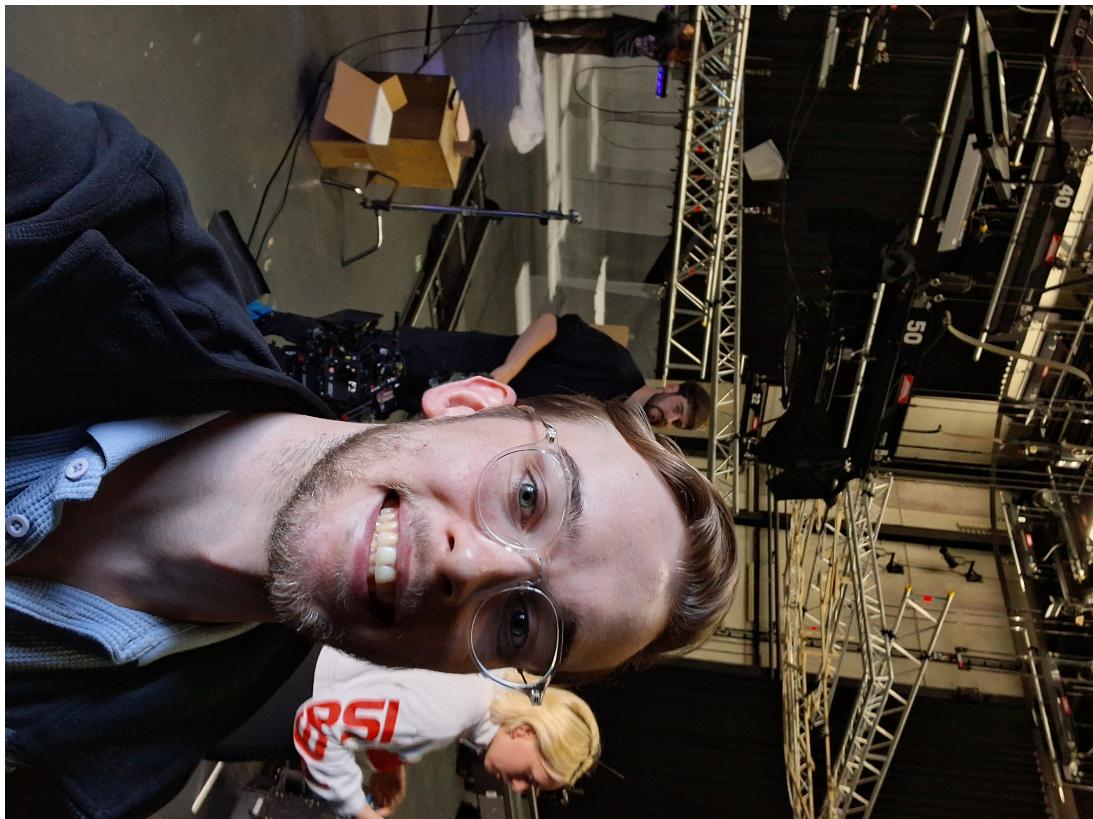


Abbildung 20: Teamdynamik: Entspannte Atmosphäre zwischen den Takes trotz technischer Herausforderungen

- **Stakeholder-Integration:** Choreographie-Meeting (10.03.2025) prägte Entwicklung
- **Iterative Validation:** Kontinuierliche Demo-Sessions verhinderten Fehlentwicklung
- **Lösungen durch Notwendigkeit:** Produktionsanforderungen führten zu praktischen technischen Anpassungen

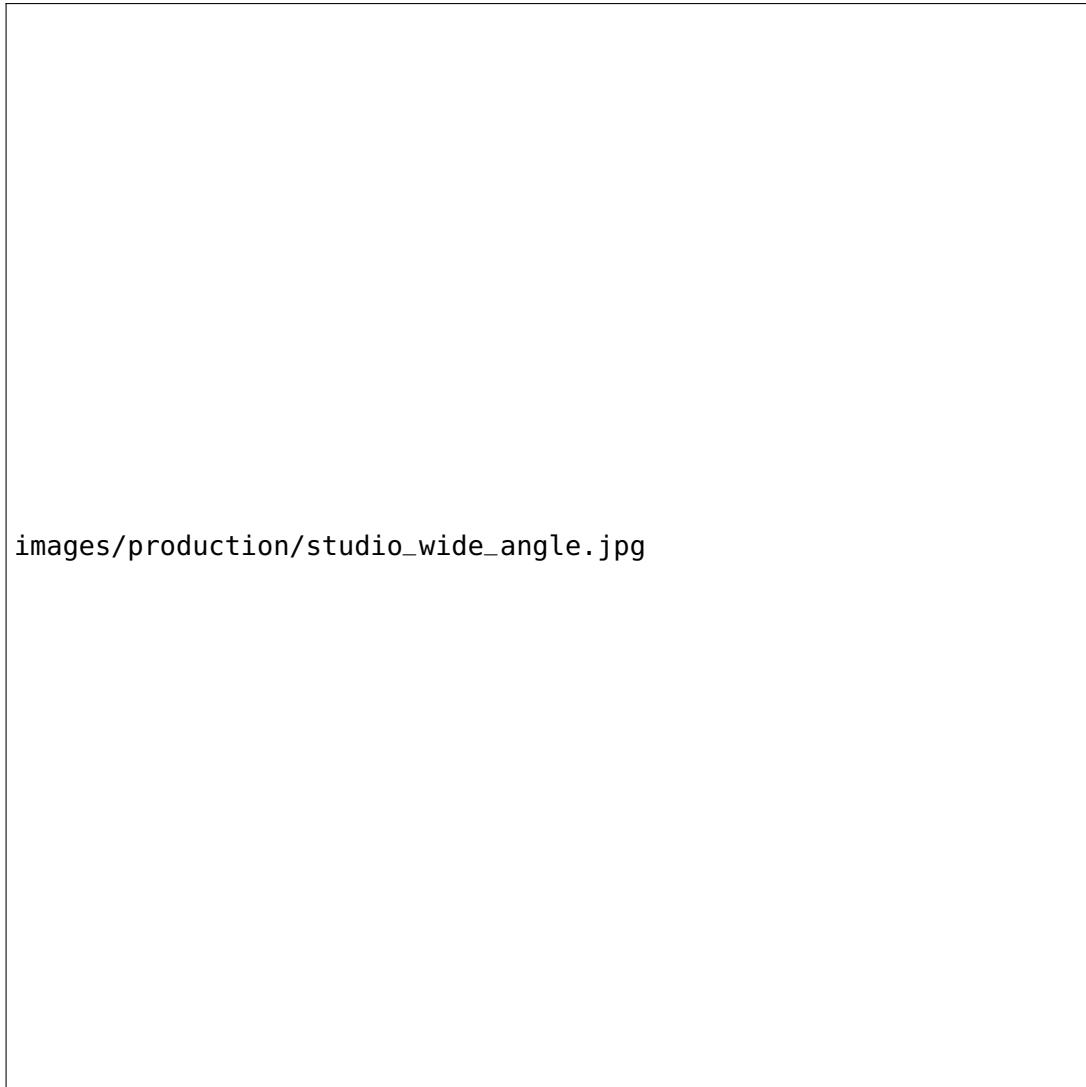
Produktionsumgebung und Systemvalidierung:

Echtzeit-Anforderungen:

Die Produktionsumgebung erforderte robuste Systemarchitektur mit minimalen Ausfallzeiten. Entwickelte Debug-Visualisierungen dienten gleichzeitig als Produktionsmonitoring-Tools. Die Echtzeit-Skelett-Overlays erwiesen sich als essentiell für die operative Systemüberwachung während der Aufnahmen.

Professionelle Koordination:

Die Integration in professionelle Produktionsabläufe erforderte strukturierte Kommunikation technischer Anforderungen mit der Produktionsleitung. Diese Koordination zwischen technischen Systemanforderungen und kreativen Produktionszielen war ein wichtiger Aspekt des interdisziplinären Projektmanagements.



images/production/studio_wide_angle.jpg

Abbildung 21: Albrecht-Ade-Studio: Professionelle Soundstage

Professionelle Entwicklung:

Vollständige Projektverantwortung:

Als einziger Entwickler trug ich die komplette technische Verantwortung. Jede Architekturentscheidung, jede Optimierung, jeder Workaround – alles lag in meiner Hand. Diese Erfahrung der vollständigen Ownership war gleichzeitig belastend und ermächtigend.

Realistische Selbsteinschätzung:

Das Projekt lehrte mich, eigene Grenzen zu erkennen und zu kommunizieren. M.A.S.K. konkurriert nicht mit industriellen Motion-Capture-Systemen – und das ist in Ordnung. Diese Ehrlichkeit ermöglichte fokussierte Entwicklung statt feature creep.

Nachhaltige Learnings:

Drei Erkenntnisse prägen meine weitere Arbeit:

1. Pragmatismus schlägt Perfektionismus: Die funktionierende 80%-Lösung ist wertvoller als die theoretische 100%-Lösung.

2. Interdisziplinarität erfordert Demut: Andere Fachbereiche haben eigene Exzellenzkriterien, die respektiert werden müssen.

3. Constraints fördern Kreativität: Die Limitierungen (Budget, Hardware, Zeit) zwangen zu alternativen Lösungen, die in einem ressourcenreichen Umfeld nie entstanden wären.

Die eigenständige Entwicklung von M.A.S.K. demonstriert, wie strukturierte Kommunikation und agile Methoden auch bei Solo-Projekten zu erfolgreicher interdisziplinärer Zusammenarbeit führen können.

9 Fazit und Beiträge

9.1 Projektergebnis

M.A.S.K. demonstriert, dass spezialisierte Motion-Capture-Lösungen mit Consumer-Hardware realisierbar sind. Das System adressiert erfolgreich die spezifische Herausforderung des Trackings unter intensiver Bühnenbeleuchtung und macht diese Technologie für Projekte mit begrenztem Budget zugänglich.

9.2 Technischer Beitrag

Die Infrarot-MediaPipe-Pipeline stellt eine praxisorientierte Lösung für spezifische Produktionsanforderungen dar. Während professionelle Systeme auf spezialisierte Hardware setzen, nutzt M.A.S.K. die verfügbare Infrarot-Funktionalität der Kinect V2 in Kombination mit MediaPipes Machine-Learning-Modell. Diese Adaptation wurde spezifisch für die Projektanforderungen entwickelt und validiert.

Die modulare TouchDesigner-Architektur ermöglicht es anderen Entwicklern, einzelne Komponenten für eigene Projekte zu adaptieren. Das GitHub-Repository enthält nicht nur Code, sondern auch die komplette Entwicklungshistorie inklusive verworfener Ansätze – wertvoll für Lernende.

9.3 Praktische Relevanz

Für Bildungseinrichtungen und unabhängige Künstler bietet M.A.S.K. eine realistische Alternative zu kommerziellen Systemen. Die Hardware-Kosten beschränken sich auf circa 100 Euro für gebrauchte Kinect V2 Hardware, wobei zusätzliche Software-Infrastruktur (TouchDesigner, OBS Studio) und Entwicklungszeit je nach Projektanforderungen zu kalkulieren sind.

Die erfolgreiche Produktion von "Echoes of the Mind" validiert die Praxistauglichkeit. Das System bewältigte 45 Takes über zwei Produktionstage ohne kritische Ausfälle – ein Beweis für die Stabilität der gewählten Architektur.

9.4 Limitierungen und Kontext

M.A.S.K. ersetzt keine professionellen Motion-Capture-Systeme. Für Anwendungen, die Millimeterpräzision, Multi-Person-Tracking oder 360-Grad-Erfassung erfordern, bleiben kommerzielle Lösungen überlegen. Diese Einschränkung ist bewusst: Durch Fokussierung auf einen spezifischen Use-Case konnte eine robuste Lösung entstehen.

Technische Grenzen:

- Single-Person-Tracking only
- Frontalerfassung (keine 360-Grad-Abdeckung)
- Manuelle Kalibrierung bei Setup-Änderungen
- Abhängigkeit von MediaPipes Modell-Limitierungen

9.5 Weiterentwicklungspotenzial

Die Open-Source-Natur des Projekts lädt zu Erweiterungen ein:

Kurzfristig:

- Automatische Kalibrierungsroutinen
- Weitere Visual-Effekte für die bestehende Pipeline
- Verbessertes Handling von Okklusionen

Mittelfristig:

- Integration neuerer ML-Modelle (MediaPipe Holistic)
- Multi-Sensor-Fusion mit IMUs oder LiDAR
- Echtzeit-Performance-Metriken

Langfristig:

- Portierung auf andere Plattformen (Mac, Linux)
- Web-basierte Konfigurationsoberfläche
- Integration in Game-Engines

9.6 Breitere Implikationen

Die zunehmende Verschmelzung von Performance-Kunst und digitaler Technologie erfordert zugängliche Werkzeuge. M.A.S.K. zeigt, dass akademische Projekte diese Lücke füllen können, wenn sie sich auf praktische Probleme fokussieren statt auf theoretische Vollständigkeit.

Die Infrarot-Lösung könnte Anwendungen über die Kunst hinaus finden: Physiotherapie-Überwachung, Sporttechnik-Analyse oder barrierefreie Interfaces könnten von der Beleuchtungsunabhängigkeit profitieren.

9.7 Abschließende Bewertung

M.A.S.K. ist eine praxisorientierte Lösung, die bewährte Technologien für spezifische Anwendungsanforderungen kombiniert. Das System entstand aus konkreten Produktionsnotwendigkeiten und demonstriert, wie interdisziplinäre Zusammenarbeit zu funktionsfähigen technischen Lösungen führen kann. Der Wert liegt in der durchdachten Integration verfügbarer Komponenten und der bewiesenen Produktionstauglichkeit.

Das Projekt erfüllte seinen Zweck: Es ermöglichte Echoes of the Mind und bleibt als Open-Source-Werkzeug für andere verfügbar. In einer Welt, in der Technologie oft als Selbstzweck entwickelt wird, ist diese zielgerichtete Pragmatik vielleicht die wichtigste Stärke von M.A.S.K.

A Vollständiges Code-Repository

A.1 Open-Source-Repository

Das vollständige M.A.S.K.-System ist als strukturiertes GitHub-Repository verfügbar für Reproduktion und Weiterentwicklung.

Repository: <https://github.com/mklemmingen/MASK> (GPL-2.0 License)

Kern-Komponenten:

Komponente	Funktion
Testing_KinectVsMediapipe.toe	Vergleichstool für MediaPipe vs. Kinect-Tracking
NoisyBlob_withTracking*.tox	Hand-Feuer-System mit ParticleGPU
Topshot_Visual_withTracking*.tox	64-Spike Radialsystem für Top-Down-Setup
PY_MediaPipeDebugCircles*.py	Debug-Visualisierung für Skelett-Detection
PY_AngleToPhase*.py	Polar-Koordinaten-Berechnung für Spike-System
PY_NodeToParticleGPU*.py	MediaPipe-zu-TouchDesigner Koordinaten-transformation

Tabelle 4: M.A.S.K. Kern-Module und Funktionen

Technische Reproduzierbarkeit:

Installation: Repository-Download und TouchDesigner 2023.11+ für vollständige Funktionalität.

Hardware-Anforderungen: Kinect V2, OBS Studio 28.0+, NVIDIA GPU empfohlen für ParticleGPU.

Dokumentation: README.md mit detaillierten Setup-Anweisungen und Konfigurationshinweisen.

A.2 Python-Skript-Referenz

Kern-Algorithmen:

PY_MediaPipeDebugCirclesForCompTops.py

- **Funktion:** Echtzeit-Debug-Visualisierung aller MediaPipe-Skeleton-Nodes

- **Features:** Confidence-basierte Farbkodierung, Größen-Mapping, Toggle-Control
- **Integration:** Text-DAT in Debug-Container

PY_NodeXYZtoCentralisedSOPTranslate.py

- **Funktion:** MediaPipe-Koordinaten zu TouchDesigner-SOP-Transformation
- **Features:** Normalisierung, Aspect-Ratio-Korrektur, Calibration-Offset
- **Input:** MediaPipe XYZ-Koordinaten
- **Output:** TouchDesigner-kompatible SOP-Koordinaten

PY_NodeDistsToDistanceAngles.py

- **Funktion:** Geometrische Analyse zwischen Skeleton-Nodes
- **Features:**
 - 33 MediaPipe-Landmarks (nose bis right_heel)
 - Echtzeit-Distanzberechnung zwischen definierten Gelenkpäaren
 - Winkelberechnung über Dot-Product mit Clamping [-1,1]
 - SpineBase-Approximation als Mittelwert der Hüften
- **Berechnete Metriken:** 13 Distanzen, 14 Winkel zwischen Gelenken
- **Output:** CHOP-Channels für geometrische Parameter über assignToNode()

Visual-Spezifische Skripte:

PY_NodeToParticleGPUtranslate.py

- **Funktion:** Hand-Node zu ParticleGPU-Koordinaten-Mapping
- **Features:** Real-normalized Coordinates, Multi-Hand-Support
- **Use-Case:** Hand-Feuer-Effekte (Visual 1)

PY_RelativeNodeValuesToBlended0and1Switch.py

- **Funktion:** Hand-über-Schulter-Erkennung für Zustandsumschaltung
- **Features:** Sanfte Interpolation, Bilateral-Hand-Processing
- **Use-Case:** Adaptive Kopf-Partikel-System (Visual 2)

PY_AngleToPhaseSkript_HandsToCenter_direct.py

- **Funktion:** 64-Spike-Winkel-zu-Phase-Transformation

- **Features:** Multi-Resolution (8,12,20,64), Atan2-Präzision
- **Input:** Hand/Fuß-Koordinaten relativ zu Screen-Center
- **Use-Case:** Radiales Spike-System (Visual 3)

Legacy-Implementierungen:

PY_OG_TestingWithSynchAndKalman.py

- **Funktion:** Historische MediaPipe-Kinect-Datenfusion
- **Features:** Kalman-Filter-Glättung, Sensor-Synchronisation
- **Status:** Archiviert, funktional für Referenz
- **Research-Value:** Multi-Sensor-Fusion-Algorithmus

A.3 TouchDesigner-Projekte

Visual-System-Implementierungen:

250421_NoisyBlob_withTrackingAndBlitzeSwitch.tox

- **Fokus:** Experimentelle Bubble-Physik mit Tracking-Integration
- **Features:** Noise-basierte Blob-Bewegungen, Blitz-Effekte
- **Status:** Entwicklungs-Prototyp

250421_Topshot_Visual_withTracking20Spoke.tox

- **Fokus:** Top-Down 20-Spike-System für Overhead-Kamera
- **Features:** Radiale Spike-Geometrie, Extremitäten-Tracking
- **Use-Case:** Produktions-Implementierung Visual 3

250421_Topshot_Visual_withTracking20Spoke_v2.tox

- **Fokus:** Erweiterte Version des 20-Spike-Systems
- **Improvements:** Optimierte Performance, erweiterte Parameter
- **Status:** Finale Produktionsversion

250421_prettyBubbles_blaueFade_withTracking.tox

- **Fokus:** Blaue Bubble-Effekte mit Fade-Transition
- **Features:** Tracking-responsive Bubble-Generierung

- **Use-Case:** Alternative zu Hand-Feuer-Effekten

Kern-Komponenten:

Focus.tox

- **Funktion:** Fokus-Steuerung für Visual-Effekte
- **Integration:** Modulare TouchDesigner-Komponente

Grain.tox

- **Funktion:** Grain-Effekt für visuelle Textur
- **Integration:** Post-Processing-Komponente

TrackingAsATox_needsRelativeMediaPipeTox.tox

- **Funktion:** Tracking-System als modulare Komponente
- **Dependencies:** Relative MediaPipe-Komponente erforderlich
- **Use-Case:** Wiederverwendbare Tracking-Pipeline

A.4 Testing und Validierung

Komparative Evaluation:

Testing_KinectVsMediapipe.toe

- **Funktion:** Direkte Vergleichstests zwischen Kinect V2 und MediaPipe
- **Features:** Simultanes Tracking, Performance-Vergleich
- **Outcome:** Validierung der MediaPipe-Überlegenheit bei partieller Okklusion
- **Documentation:** Testing_README.md

A.5 Dokumentation und Assets

Visuelle Dokumentation:

DocuPictures/

- Enthält alle Screenshots und Diagramme für die Projektdokumentation
- Visual-Pipeline-Übersichten und Systemarchitektur-Diagramme
- Debug-Interface-Screenshots und Production-Dokumentation

comment_assets/

- Zusätzliche Assets für Kommentierung und Erklärung
- Unterstützende Materialien für Code-Dokumentation

Backup und Versionierung:

Backup/

- Backup-Versionen kritischer TouchDesigner-Dateien
- Entwicklungs-Snapshots für Rollback-Fähigkeiten
- Archivierte Versionen verschiedener Entwicklungsphasen

Das Repository unter <https://github.com/mklemmingen/MASK> bietet eine kompakte, aber vollständige Implementation des M.A.S.K.-Systems. Die direkte Struktur ermöglicht einfachen Zugang zu allen Kern-Komponenten ohne komplexe Hierarchien, während die Dateinamen die chronologische Entwicklung und spezifische Funktionen klar kennzeichnen.

B Team und Kommunikation

B.1 Professionelle Kollaboration zwischen Hochschulen

Interdisziplinäres Entwicklungsteam:

Technische Führung - Hochschule Reutlingen:

- **Entwickler:** Marty Lauterbach (Sole Developer, Technische Projektleitung)
- **Rolle:** System-Architektur, MediaPipe-Integration, TouchDesigner-Pipeline-Entwicklung
- **Verantwortung:** Vollständige technische Implementation als einziger Entwickler der Hochschule
- **Entwicklung:** Eigenständige Umsetzung der Infrarot-MediaPipe-Pipeline

Kreative Partner - Filmakademie Baden-Württemberg:

- **Designerinnen:** Maja Litzke und Rahel Fundinger (Creative Direction, Künstlerische Leitung)
- **Projektkontext:** Echoes of the Mind Cinematographische Tanzproduktion über mentale Zustände
- **Kollaborationsqualität:** Intensive interdisziplinäre Partnerschaft zwischen Technik und Kunst
- **Kommunikation:** Regelmäßige Sprint-Meetings und kontinuierlicher fachlicher Austausch

Interdisziplinäre Konstellation: Diese Kooperation zwischen einem Hochschulentwickler und erfahrenen Filmakademie-Designerinnen ermöglichte eine produktive Synthese aus technischer Entwicklung und künstlerischer Vision.

Betreuung und Beratung:

Hochschule Reutlingen:

- **Projektbetreuung:** Anja Hartmann (Akademische Betreuung, Bewertung)
- **Fachberatung:** Niklas Zidarov Digital Art (Technische Beratung, 17.01.2025)
- **Rolle:** Akademische Unterstützung

Produktionsumgebung:

- **Albrecht-Ade-Studio:** Produktionsstudio der Filmakademie

- **Technische Crew:** Kamerateam, Beleuchtung, Tontechnik
- **Equipment-Management:** Hardware-Bereitstellung und -Support

B.2 Kommunikationsprotokoll

Meeting-Struktur:

Regelmäßige Termine:

- **Sprint-Meetings:** Alle 2-3 Wochen, 90 Minuten, Online mit Filmakademie
- **Demo-Sessions:** Nach jedem Sprint, 60 Minuten, Live-Demonstration der Fortschritte
- **Technical-Reviews:** Bei Bedarf, Problem-focused
- **Production-Meetings:** Pre-Production, intensivere Koordination für Studioeinsatz

Laufende Kommunikation:

- **WhatsApp-Gruppe:** Schnelle Updates, Terminkoordination, Dateiaustausch
- **Video-Calls:** Spontane Problem-Lösung, Screen-Sharing für technische Diskussionen
- **GitHub-Repository:** Code-Sharing, Versions-Dokumentation

B.3 Professionelle Kollaborationsmeilensteine

Technische Grundlegung (17.01.2025): Ausgangssituation: Als einziger Entwickler der Hochschule Reutlingen suchte ich Expertenberatung für eine geeignete Motion-Capture-Strategie.

- **Fachberatung:** Lehrbeauftragter Digital Art empfahl MediaPipe-Integration
- **Strategische Entscheidung:** MediaPipe als Subprozess für Kinect RGB-Stream
- **Technische Vision:** Hybrid-System für verbesserte Tracking-Robustheit
- **Eigenständige Umsetzung:** Vollständige Implementation-Verantwortung übernommen

Interdisziplinärer Projekt-Kick-off (Ende Januar 2025): Erste Zusammenarbeit: Intensive Kollaboration mit Maja Litzke und Rahel Fundinger zur Definition der technischen Architektur basierend auf künstlerischen Anforderungen.

Gemeinsam entwickelte Systemarchitektur:

- **Dual-Source-Konzept:** MediaPipe-Kinect-Fusion basierend auf technischer Analyse und kreativen Anforderungen
- **Interdisziplinäre Integration:** Choreographie-spezifische Trigger-Pipeline durch kolaborative Spezifikation
- **Bewegungsanalyse:** Geschwindigkeit, Rotation und Winkelbeziehungen entsprechend künstlerischer und technischer Kriterien
- **Kollaborative Arbeitsweise:** Sprint-basierte Entwicklung mit regelmäßigen interdisziplinären Abstimmungen

Kreative Anforderungsdefinition (10.03.2025): Wendepunkt der Zusammenarbeit: Die Designerinnen präsentierten konkrete choreographische Konzepte, die meine technischen Lösungsansätze fundamental beeinflussten.

Interdisziplinäre Systemdefinition:

1. **Hand-Tracking-Priorisierung:** Fokus auf Arm- und Handbewegungen nach kreativer Spezifikation
2. **Räumliche Kalibrierung:** Präzise Beamer-Projektion für Bodenchoreographie
3. **Visual-Responsivität:** Echtzeit-Bewegungserkennung für emotionale Tanznarrative
4. **Multi-Perspektiven-System:** Top-Down und frontale Tracking-Modi

Interdisziplinäre Translation: Künstlerische Anforderungen und technische Möglichkeiten wurden gemeinsam in umsetzbare Implementation-Spezifikationen entwickelt.

B.4 Entscheidungslogprotokoll

Eigenständige Architekturentscheidungen:

Hybrid-Tracking-System Definition (Ende Januar 2025)

- **Kontext:** Projekt-Kick-off mit Filmakademie zur Systemarchitektur
- **Entwickler-Entscheidung:** Dual-Source-Ansatz basierend auf Technical Review
- **Begründung:** Kombination aus MediaPipe-Robustheit und Kinect-Tiefendaten
- **Implementation-Plan:** Kalman-Filter-basierte Datenfusion

MediaPipe-Priorisierung (Sprint 2, bis 24.03.2025)

- **Eigenständige Evaluation:** Komparative Tests zeigen MediaPipe-Überlegenheit bei partieller Okklusion
- **Stakeholder-Konsultation:** Filmakademie bevorzugt Robustheit über absolute Präzision
- **Technische Analyse:** MediaPipe höhere Confidence-Werte bei nicht-sichtbaren Körperregionen
- **Entwicklungsentscheidung:** MediaPipe als primäres System, Kinect als Backup
- **Dokumentation:** Testing_KinectVsMediapipe.toe als Vergleichsreferenz

Kinect-Deprecation (Sprint 4, bis 20.04.2025)

- **Problem-Identifikation:** Hybrid-System erhöht Komplexität ohne klaren Mehrwert
- **Stakeholder-Feedback:** Vereinfachung für bessere Wartbarkeit gewünscht
- **Performance-Analyse:** MediaPipe-Only zeigt gute Performance bei unvollständigen Skeleton-Daten
- **Architektur-Refactoring:** Vollständige Modularisierung von MediaPipe, Elimination der Kinect-Dependencies
- **Begründung:** Limitierte Partial-Body-Detection der Kinect vs. MediaPipe-Robustheit

Iterative Scope-Entwicklung:

Bubble-Physics zu Visual-System-Trias (Sprint 4-5)

- **Ursprünglicher Plan:** Komplexe Bubble-Displacement-Algorithmen
- **Technische Herausforderung:** Unzureichende Feedback-Loops für persistente Bubble-Lifecycles
- **Evaluierte Implementierungen:** C# Shader, ParticleGPU Manipulation, Hybrid CPU-GPU Approach
- **Stakeholder-Realignment:** Alternative Visual-Ansätze für praktische Choreographie-Integration
- **Finale Implementierung:** Hand-Fire-Effects, Adaptive Head-Particles, Radial Spike-System
- **Outcome:** Drei konkrete, choreographie-spezifische und produktionstaugliche Visualisierungen

Infrarot-Lösung (Sprint 6, bis 12.05.2025)

- **Kritisches Problem:** Beamer-Projektionen verursachen RGB-Kamera-Überbelichtung
- **Stakeholder-Impact:** Fundamental für Produktionsumgebung der Filmakademie
- **Eigenständige Lösung:** Kinect V2 Infrarot-Stream als Primary Input über OBS Virtual Camera
- **Technische Implementation:** IR-optimierte MediaPipe Skeleton-Detection
- **Zeitmanagement:** Zusätzliche Entwicklungszeit für fundamentale Problemlösung
- **Validation:** Vollständige Beleuchtungs-Immunität in Produktionsumgebung erreicht

B.5 Eigenständige Entwicklungsprozesse

Solo-Development-Workflow:

Tägliche Entwicklungsroutine:

- **Morgendliche Prioritätsetzung:** Sprint-Ziele vs. technische Herausforderungen
- **Iterative Implementation:** Kurze Entwicklungszyklen mit kontinuierlicher Validierung
- **Dokumentation parallel:** Code-Kommentierung und Architektur-Notizen während Entwicklung
- **Abendliche Reflektion:** Fortschritts-Assessment und nächste Schritte

Problem-Solving-Ansatz:

- **Eigenständige Recherche:** MediaPipe-Dokumentation, TouchDesigner-Community
- **Prototyping:** Schnelle Proof-of-Concept-Implementierungen
- **Stakeholder-Konsultation:** Bei konzeptionellen oder künstlerischen Entscheidungen
- **Iterative Verfeinerung:** Kontinuierliche Optimierung basierend auf Testing

Sprint-Retrospektiven (Solo-Reflektion):

Sprint 1-2: Grundlagen-Phase

- **Lernkurve:** MediaPipe-TouchDesigner-Integration steiler als erwartet
- **Stakeholder-Alignment:** Regelmäßige Kommunikation essentiell für Richtung
- **Technische Validation:** Komparative Tests bestätigen Hybrid-Ansatz-Wert

Sprint 3-4: Architektur-Phase

- **Komplexitäts-Management:** Vereinfachung führt zu besserer Performance
- **Modularität:** Containerization erleichtert Debugging deutlich
- **Scope-Flexibilität:** Bereitschaft zu Architektur-Änderungen erweist sich als nützlich

Sprint 5-6: Produktions-Phase

- **Lösungsfindung unter Druck:** Infrarot-Adaptation entsteht aus Produktionsnotwendigkeit
- **Stakeholder-Coordination:** Enge Abstimmung für Studio-Integration kritisch
- **Performance-Validation:** Real-world Testing unerlässlich für Produktionsreife

C Danksagung

Mein herzlicher Dank gilt allen Beteiligten, die dieses Projekt ermöglicht und unterstützt haben.

Besonderer Dank geht an die **Filmakademie Baden-Württemberg in Ludwigsburg**, insbesondere an **Maka Litzke** und **Rahel Fundinger**, für ihre wertvolle fachliche Expertise und die konstruktive Zusammenarbeit während des gesamten Projektverlaufs.

Ebenso möchte ich mich bei der **Hochschule Reutlingen** bedanken, namentlich bei **Prof. Anja Hartmann** für das Angebot und die Möglichkeit, das Projekt als Teil meines Studiums zu machen, und **Niklas Zidarov**, für seine kompetente Hilfestellung und Anregung bei der Umsetzung dieses Vorhabens.

Die interdisziplinäre Zusammenarbeit zwischen beiden Institutionen hat maßgeblich zum Erfolg dieses Projekts beigetragen und wertvolle Einblicke in verschiedene Fachbereiche ermöglicht.

Vielen Dank für Ihre Unterstützung