

# Ich habe etwas vorbereitet – 3D-Modell-basierte AR/VR-Kopplung

Markus Kiesel<sup>1</sup>, Alexander Kögel<sup>2</sup>

<sup>1</sup> CMC-Kiesel GmbH, <sup>2</sup> TU Chemnitz – Professur Arbeitswissenschaft und Innovationsmanagement

## Zusammenfassung

Eine Möglichkeit der Bereitstellung von Objektdaten für die Kollaboration zwischen Augmented-Reality (AR)-Anwendern im Feld und Experten, die in einer Virtuellen Umgebung (VR) agieren, sind vorab aus CAD-Daten erstellte 3D-Modelle. Der Beitrag beschreibt eine Vorgehensweise zur Modellerstellung sowie Anforderungen an den Backend-Server und skizziert einen Ansatz, um Abweichungen zwischen Modell und Realobjekt zu identifizieren.

## 1 Einleitung

Die naheliegende Grundlage für die Bereitstellung von 3D-Objektmodellen für AR/VR-Kollaborationen bilden vorhandene CAD-Daten, welche typischerweise in der Entwurfs- und Konstruktionsphase einer Maschine oder Anlage mit etablierten CAD-Programmen erstellt werden. CAD-Modelle dienen als Digitaler Zwilling bereits als zentrale Referenz für sämtliche nachfolgenden Produktionsschritte – angefangen bei der Fertigung einzelner Komponenten bis hin zur finalen Montage.

Vorteil der CAD-Daten bzw. Digitalen Zwillinge ist, dass sie in der Regel auf Basis vollständiger und präziser Stücklisten alle Komponenten einer Maschine, Anlage oder – allgemein gesprochen – eines Objekts umfassen. Auch die AR/VR-basierte Fernunterstützung von Fehleranalysen, Wartungen und Instandsetzungen profitiert von solchen detailgetreuen Abbildern.

Allerdings können CAD-Daten nicht direkt und im ursprünglichen Format für die Echtzeitvisualisierung in AR und VR genutzt werden. Im Folgenden werden die Gründe genauer erläutert und anschließend Lösungen aufgezeigt.

## 2 BREP und Mesh-Modelle. Was ist was?

CAD-Daten liegen grundsätzlich als mathematische Modelle vor, sogenannte Boundary Representation (BREP)-Daten. Hierbei wird die Geometrie eines Modells durch **exakte mathematische Beschreibungen seiner Oberflächen** dargestellt. Die BREP-Modelle entstehen mittels verschiedener Modellierungsalgorithmen, etwa durch Extrusions- und Subtraktionsverfahren (CSG-Algorithmen, Constructive Solid Geometry). Dabei werden eine oder mehrere zweidimensionale Kurven zu komplexeren 3D-Modellen verarbeitet.

Diese Vorgehensweise ist für Anwendungen geeignet, die eine hohe Genauigkeit, Maßhaltigkeit und numerische Stabilität erfordern – beispielsweise in der Konstruktion, Simulation oder Fertigung. Für die Darstellung in AR- und VR-Umgebungen ist dieses Verfahren jedoch ungeeignet, da die mathematische Beschreibung der Oberflächen in Echtzeit berechnet werden müsste, was zu erheblichen Performanceeinbußen führt.

Ein weiteres Hindernis für die direkte Nutzung von CAD-Daten in AR und VR ist die hohe **Komplexität typischer CAD-Gesamtmodelle**. Maschinen oder Anlagen bestehen häufig aus einer großen Anzahl einzelner Bauteile, die schnell 1.000 bis 10.000 erreicht, und die in unterschiedlicher Form miteinander verbunden sind. Auch diese Komplexität führt zu einem massiven Ressourcenbedarf bzw. Performanceproblemen bei Echtzeitvisualisierungen.

Um eine performante Echtzeitvisualisierung von CAD-Modellen zu ermöglichen, sind die BREP-Daten zwingend in effizient verarbeitbare, polygonale Repräsentationen – sogenannte Mesh-Daten – umzuwandeln. Dieser Prozess wird **Triangulierung** genannt. Dabei werden alle Oberflächen des ursprünglichen mathematisch definierten Modells durch Dreiecke angenähert. Dadurch werden beispielsweise Kreisflächen zu N-Ecken. Die Genauigkeit der Mesh-Daten lässt sich durch die Anzahl der verwendeten Dreiecke variieren: Eine hohe Anzahl an Dreiecken bietet eine bessere visuelle Annäherung an das ursprüngliche CAD-Modell.

Abbildung 1 veranschaulicht dies anhand eines Robotermodells. Beide Modelle basieren auf denselben CAD-Daten und wirken auf den ersten Blick sehr ähnlich. Das linke Modell besitzt jedoch fast doppelt so viele Dreiecke wie das rechte und erscheint bei näherer Betrachtung schärfer und realitätstreuer.

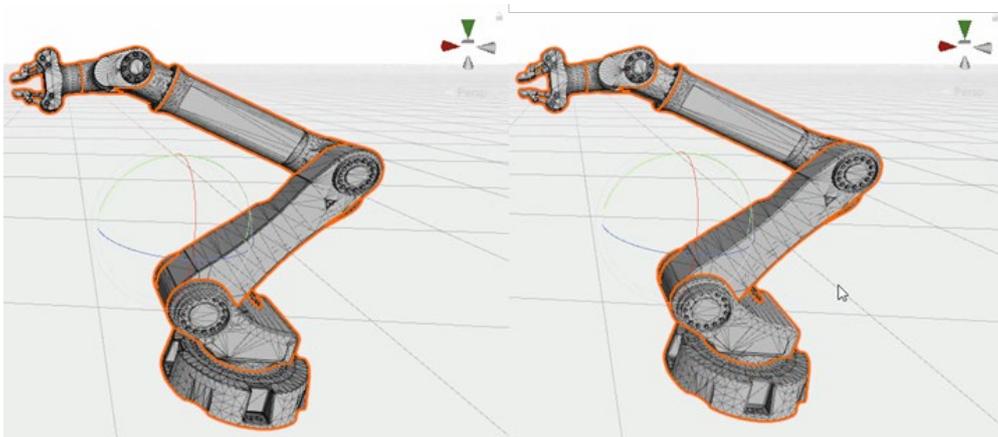


Abbildung 1: Beispielhafte Unterschiede bei der Triangulierung: Linkes Modell mit 50.000 und rechtes Modell mit 26.000 Dreiecken (eigene Darstellung)

Eine zu hohe **Anzahl von Dreiecken pro Szene** führt wiederum zu Performanceeinbußen. Tabelle 1 zeigt empfohlene Grenzwerte für deren maximale Anzahl bei ausgewählten Augmented- und Mixed-Reality-Geräten. Dabei wird für Praktiker schnell deutlich, dass diese Grenzwerte bei komplexen Anlagenmodellen schnell erreicht oder überschritten werden. Zudem sind Reservekapazitäten für weitere Berechnungen wie komplexe Shader, Beleuchtung oder Materialeigenschaften einzukalkulieren.

Tabelle 1: Übersicht maximale Anzahl an Dreiecken

Gerät	Maximale Anzahl an Dreiecken	Geräte-Art
<b>Microsoft HoloLens 1</b>	50.000 – 100.000	AR
<b>Microsoft HoloLens 2</b>	100.000 – 300.000	AR
<b>Meta Quest 2</b>	750.000 – 1.000.000	XR
<b>Meta Quest 3</b>	1.500.000 – 2.500.000	XR
<b>Magic Leap 1</b>	300.000 – 500.000	AR
<b>Magic Leap 2</b>	500.000 – 1.000.000	AR
<b>HTC Vive XR Elite</b>	1.500.000 – 3.000.000	XR
<b>Apple Vision Pro</b>	3.000.000 – 5.000.000	XR

Im Kontext von VR/AR-Anwendungen für die Fernunterstützung in der Industrie reduziert sich die effektiv für Maschinenmodelle nutzbare Anzahl von Dreiecken weiter, da meist noch zusätzliche Objekte wie Avatare, Hilfsgeometrie zur Bedienung sowie grafische Overlays dargestellt werden müssen.

Daher ist zur Gewährleistung ausreichender Performance in der Regel eine umfangreiche, meist manuelle Optimierung von Mesh-Daten für Maschinen und Anlagen notwendig. Dies erfolgt typischerweise in folgenden Schritten:

- Initiale Konvertierung des gesamten Maschinenmodells in Mesh-Daten unter Berücksichtigung einer definierten Genauigkeit. Diese hängt von der Größe und Komplexität der Anlage ab.
- Reduzierung der Dreiecksanzahl einzelner Elemente durch vereinfachende Verfahren wie Decimation und Mesh-Optimierung.
- Optionales Entfernen sehr kleiner, nicht relevanter Bauteile, beispielsweise Schrauben, Muttern oder Unterlegscheiben, die visuell kaum wahrnehmbar sind.
- Optionales Entfernen irrelevanter Geometrie-Features, etwa innerer Strukturen, die für die AR/VR-Darstellung irrelevant sind.

Auch die Anzahl der einzelnen Bauteile beeinflusst die Darstellungsleistung einer Echtzeitvisualisierung erheblich. Es böte sich daher an, Bauteile nach gemeinsamen Eigenschaften – beispielsweise gleicher Materialfarbe oder Zugehörigkeit zu einer Baugruppe – zusammenzufassen. Das schränkt allerdings die Flexibilität ein, da so einzelne Komponenten nicht mehr separat bewegt oder selektiv bearbeitet werden können. Für die Fernunterstützung via AR/VR-Kopplung sollte daher auf diese Form der Optimierung verzichtet werden, um eine hohe Interaktivität auf Komponentenebene zu gewährleisten.

In den Anwendungsfällen [1] des Projekts TeleInteraction-XR wurden Geräte, Maschinen und Anlagen mit einer moderaten Komplexität betrachtet. Deshalb konnte dort auf die oben genannten optionalen Optimierungsschritte verzichtet werden.

### 3 Datenkommunikation – Backend-Server als Drehscheibe

Um eine nahtlose Echtzeit-Kollaboration zwischen verschiedenen örtlich verteilten Systemen zu ermöglichen, ist ein möglichst flexibles und skalierbares Backend-System nötig. Das Backend soll heterogene Lösungen unter-

schiedlichster Art anbinden und mit wechselnden Zahlen von Nutzenden effizient umgehen können.

Im Zentrum des im Projekt TeleInteraction-XR entwickelten Backend-Systems steht ein leistungsfähiger, event-basierter NodeJS-Server, der die Kommunikation zwischen sämtlichen Teilnehmenden (Clients) koordiniert. Dessen primäre Aufgabe besteht darin, von den einzelnen Clients empfangene Datenpakete entgegenzunehmen und umgehend an alle anderen verbundenen Systeme weiterzuleiten. Darüber hinaus speichert der Server die Historie sämtlicher Nachrichten, um nachträglich hinzukommenden Clients ein vollständiges Abbild der Session bereitstellen zu können.

Die Kommunikation zwischen Backend und Clients erfolgt über zwei Hauptkanäle, die sich aufgrund ihrer unterschiedlichen technologischen Stärken gegenseitigen ergänzen:

**gRPC** (Google Remote Procedure Call) ist ein leistungsstarkes, plattformübergreifendes Open-Source-Framework zur Kommunikation zwischen verteilten Systemen. Es basiert auf Protocol Buffers (protobuf), einer standardisierten Methode zur kompakten und maschinenlesbaren Definition von Schnittstellen. Durch den Einsatz von HTTP/2 bietet gRPC eine besonders schnelle, zuverlässige und ressourcenschonende Datenübertragung. Ein weiterer Vorteil liegt darin, dass definierte protobuf-Schnittstellen automatisch und mit minimalem Aufwand in zahlreiche Programmiersprachen kompiliert werden können, was die Integration in diverse Applikationen erleichtert.

**WebSockets** wiederum stellen ein bidirektionales, vollduplexfähiges Kommunikationsprotokoll dar, welches über eine persistente TCP-Verbindung arbeitet. Im Gegensatz zu klassischen HTTP-Verbindungen müssen Clients dabei nicht kontinuierlich neue Anfragen senden, um aktuelle Informationen zu erhalten. Vielmehr ermöglicht eine einmal etablierte Verbindung einen kontinuierlichen, nahezu verzögerungsfreien Datenaustausch zwischen Server und Client in Echtzeit. Aufgrund der breiten Unterstützung in modernen Browsern und vielfältigen Anwendungen stellen WebSockets eine äußerst universelle Lösung für die Echtzeitkommunikation dar.

Das entwickelte Backend zeichnet sich insbesondere durch seine Fähigkeit aus, Informationen dynamisch zwischen diesen beiden Kommunikationskanälen zu übersetzen: Es ist möglich, eine Seite einer Kommunikationsverbindung via WebSockets anzubinden, während auf der Gegenseite die Kommunikation über gRPC erfolgt. Damit kann nahezu jedes System einfach und effizient an das Kollaborations-Backend angekoppelt werden.

## 4 Besonderheiten bei 3D-basierte AR/VR-Kollaboration

Um spezielle Anforderungen der AR/VR-Kollaboration mittels vorbereiteter 3D-Modelle zu genügen, wurden dem Backend-Server folgende Funktionalitäten hinzugefügt:

**Session-Management:** Um mehrere Kollaborationssitzungen parallel durchführen zu können, wurde ein Mechanismus implementiert, der Clients und deren übertragene Daten eindeutig einzelnen Sessions zuordnen und Datenpakete adäquat verteilen kann. Eine Session wird dazu durch eine eindeutige alphanumerische Kennung identifiziert. So lässt sich mit nur einem Server eine Skalierung auf zahlreiche parallele Sessions erreichen.

**Transformationsübertragung:** Um sicherzustellen, dass sämtliche 3D-Modelle auf allen Plattformen konsistent und positionsgenau dargestellt werden, werden Positionsänderungen der einzelnen Komponenten kontinuierlich an alle Teilnehmer übertragen. Das entsprechende Datenpaket umfasst jeweils Informationen zur Position, Rotation sowie zur Skalierung eines spezifischen Bauteils im dreidimensionalen Raum.

**Avatar-Übertragung:** Für Avatare müssen neben Transformationsdaten (Position, Orientierung) zusätzliche Zustandsinformationen wie Benutzernamen, Statusinformationen (z. B. offene vs. geschlossene Hand) sowie weitere dynamische Eigenschaften übertragen werden.

**Highlight-Funktionalität:** Um die Aufmerksamkeit von Teilnehmenden während einer Kollaboration zu lenken, wurde eine zusätzliche Funktion implementiert, mit der einzelne oder mehrere Komponenten innerhalb eines komplexen Modells visuell hervorgehoben werden können.

**Identifikation:** Die eindeutige Identifikation sämtlicher Komponenten innerhalb des virtuellen Modells erfolgt über Global Unique Identifiers (GUIDs), die während der Datenaufbereitung und Optimierung automatisch für jedes einzelne Bauteil vergeben werden. Nach Abschluss dieses Prozesses wird das gesamte optimierte Modell als GLB-Datei exportiert.

GLB (Graphics Language Binary) ist ein von der Khronos Group entwickeltes offenes binäres Dateiformat für das standardisierte Grafikformat glTF (Graphics Language Transmission Format). Es wurde für die Speicherung und Übertragung komplexer 3D-Modelle optimiert und eignet sich hervorragend für Web-, AR- und VR-Anwendungen, da es kompakt und universell kompatibel ist. Außerdem können neben Geometrie- und Texturdaten auch beliebige nutzerdefinierte Metadaten integriert werden.

Die Verteilung der GLB-Dateien an die Teilnehmenden einer AR/VR-Kollaboration kann entweder direkt über das Backend oder über einen manuellen Datenaustausch erfolgen.

Da die Anwendungsfälle im Projekt TeleInteraction-XR auf spezifische, vorab bekannte Geräte, Maschinen bzw. Anlagen der beteiligten Industriepartner fokussiert waren, wurden die vorbereiteten 3D-Objektmodelle insbesondere auf der AR-Seite bereits im Vorfeld der Kollaboration auf der jeweiligen lokalen Applikation integriert. Dies gewährleistet eine hohe Performance und minimierte die initialen Ladezeiten beim Start der Kollaborationssitzung.

## 5 Verortung in Koordinatensystemen

Eine exakte und zuverlässige Ausrichtung zwischen realem Objekt und seinem virtuellen Abbild in der AR-Darstellung ist schon für allein betriebene AR-Anwendungen essenziell. Für die AR/VR-Kopplung sind zusätzlich noch die unterschiedlichen Weltkoordinaten der AR- und der VR-Seite zu synchronisieren (Grundlagen und Begriffe siehe [2]). Dafür stehen mehrere Methoden zur Verfügung, von denen zwei näher betrachtet werden:

Beim **image-basierten Verfahren** erfolgt die Verortung des virtuellen Modells durch Erkennung und räumliche Zuordnung eines zuvor definierten Referenzbildes oder Markers. Dieser Marker muss speziell für den Einsatzzweck konzipiert sein und über eindeutig erkennbare Muster und Kontraste verfügen. Nach erfolgreicher Bilderkennung durch die AR-Anwendung wird automatisch die räumliche Position und Orientierung des virtuellen 3D-Modells ermittelt und entsprechend an die reale Anlage angelegt.

Damit dieses Verfahren korrekt funktioniert, müssen sämtliche 3D-Daten innerhalb eines verschiebbaren, lokalen Koordinatensystems (Local Space) eines Hilfsobjekts positioniert sein, dessen Ursprung (Origin) exakt auf dem Referenzmarker liegt. Nach der initialen Ausrichtung auf Basis des Bildes erfolgt häufig ein Übergang zum sogenannten Sensorfusion-basierten Tracking. Hierbei werden zusätzliche Sensoren wie Kameras, Gyroskope oder Beschleunigungsmesser eingesetzt, um auch bei Bewegung eine kontinuierliche Positionsverfolgung zu ermöglichen. Allerdings besteht bei zunehmender Entfernung vom ursprünglichen Marker die Gefahr, dass Abweichungen auftreten und somit Ungenauigkeiten bei der Zuordnung zwischen virtueller und realer Welt entstehen.

Eine Alternative ist die **semi-manuelle Verortung** (Anchoring). Hierbei bestimmt der Nutzer einen Punkt, der sowohl im digitalen Modell als auch in der realen Anlage leicht identifizier- und erreichbar ist. Besonders geeignet

sind dafür definierte Eckpunkte von Geometrien. Abbildung 2 zeigt dafür ein Beispiel. Links ist das virtuelle 3D-Modell der Maschine zu sehen, während rechts die Platzierung des Ankerpunktes an der realen Anlage gezeigt wird.

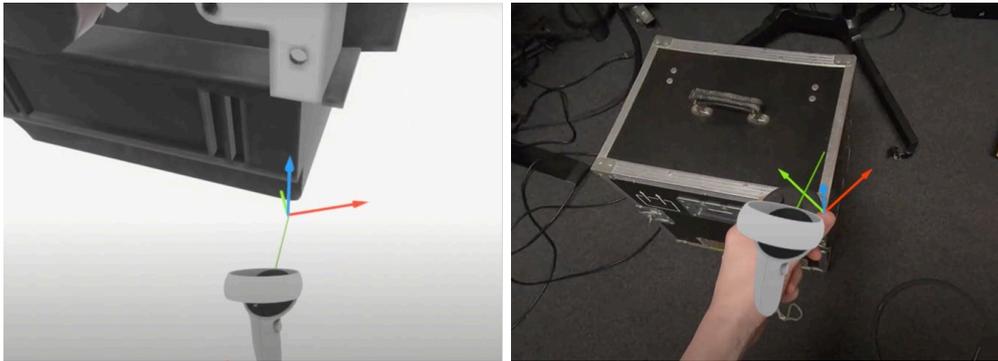


Abbildung 2: Ankerung, links am digitalen Modell im VR Modus, rechts an einem realen Modell in XR Modus

## 6 Digitaler Schatten – über den Produktlebensweg veränderliche 3D-Modelle

Nach erfolgreicher Inbetriebnahme einer Maschine oder Anlage folgt typischerweise eine lang andauernde Abfolge von Betriebs-, Wartungs- und ggf. Reparatur-, Umbau- bzw. Modernisierungsphasen. Dabei erfolgte Modifikationen werden nicht automatisch im ursprünglichen CAD-Modell berücksichtigt, wodurch es über die Zeit zu Abweichungen zwischen der real existierenden Anlage und ihrem virtuellen, digitalen Zwilling kommt. Wird das digitale Modell nicht regelmäßig aktualisiert, wachsen diese Abweichungen mit der Zeit immer weiter an, was die Präzision und Aussagekraft der AR/VR-basierten Kollaboration zunehmend beeinträchtigt.

Um trotz dieser – nicht immer vermeidbaren – Diskrepanzen zwischen Realität und virtuellem Modell eine akzeptable AR/VR-Kollaboration zu gewährleisten, kann es helfen, diese Abweichungen sicht- und nachvollziehbar zu machen.

Im Projekt TeleInteraction-XR wurde mit der **Integration von Punktwolken zur Visualisierung von Abweichungen** eine Methode zur Darstellung dieser Unterschiede beispielhaft erprobt. Die Punktwolke besteht dabei aus einer Vielzahl von räumlich erfassten Messpunkten, die mittels 3D-Scanning-Technologien (z. B. LiDAR-Scanner oder photogrammetrische Verfahren)

aus der realen Umgebung aufgenommen wurden. Diese Punktwolken erfüllen mehrere Aufgaben:

- **Visualisierung der tatsächlichen Umgebung:** Die aufgenommenen Punktwolken ermöglichen es, eine detaillierte und realitätsgetreue Abbildung der tatsächlichen Umgebung zu erzeugen. Dieser Kontext verbessert gerade auf der AR-Seite die Orientierung erheblich.
- **Identifikation von Abweichungen:** Durch den Vergleich der Punktwolke mit dem 3D-Modell können Abweichungen erkannt und analysiert werden.
- **Erleichterte Entscheidungsfindung für Experten:** Auf Basis der identifizierten Anforderungen kann entschieden werden, ob die Modelle und die zugrundeliegenden CAD-Daten nachzupflegen oder unzulässige Abweichungen in der Realität zu korrigieren sind.

Als beispielhafte Erprobung dieses Konzepts wurde durch einen 3D-Scan mit einem iPad eine Punktwolke eines realen Objekts aufgenommen und anschließend in das bestehende digitale Modell integriert. Aus der gemeinsamen Darstellung von digitalem Modell und real erfasster Punktwolke ließen sich erwartungsgemäß die Diskrepanzen erkennen.

## 7 Fazit

Die AR/VR-Kopplung auf Basis vorbereiteter 3D-Modelle, die aus CAD-Daten aufbereitet wurden, konnte in Verbindung mit einem spezifisch konfigurierten Backend-Server erfolgreich implementiert, getestet und praktisch validiert werden.

Die Vorgehensweise, Objektdaten für die AR/VR-Kollaboration mit vorbereiteten 3D-Modellen bereit zu stellen, verspricht insbesondere für neue Maschinen und Anlagen einen hohen Nutzwert. Für diese sind in der Regel vollständige und präzise CAD-Daten verfügbar, die zuverlässig und relativ aufwandsarm in hinreichend detailreiche 3D-Modelle gewandelt werden können.

Um Änderungen an Maschinen und Anlagen, die über den Lebenszyklus entstehen und die nicht ohne weiteres in den CAD-Modellen nachgepflegt werden, in der AR/VR-Kollaboration zu berücksichtigen, wurde beispielhaft das Erfassen einer Punktwolke vom realen Objekt und dessen Integration in die AR/VR-Umgebungen mit Erfolg erprobt. Es sind jedoch weitere Entwicklungsschritte nötig, um eine industrietaugliche Anwendung zu ermöglichen.

*Die Forschungs- und Entwicklungsarbeit wurde vom Bundesministerium für Bildung und Forschung im Fachprogramm „Zukunft der Wertschöpfung“ und der Fördermaßnahme „Innovative Arbeitswelten im Mittelstand“ im Projekt TeleInteraction-XR (Förderkennzeichen 02L21B550-4) gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Verantwortung für den Inhalt der Veröffentlichung liegt beim Autor.*

## 8 Literaturverzeichnis

- [1] Ruffert, D., Kögel, A., Löffler, T. (2025). So fern und doch so nah – Telepräsenes Arbeiten in der Industrie. In: Bullinger-Hoffmann, A.C. (2025). Telepräsenes Fernunterstützung in der Industrie. aw&I – Wissenschaft und Praxis.
- [2] Kögel, A. (2025). Wer und was ist wo? - Synchronisation von Virtueller und Erweiterter Realität. In: Bullinger-Hoffmann, A.C. (2025). Telepräsenes Fernunterstützung in der Industrie. aw&I – Wissenschaft und Praxis.

### Autoren



Markus Kiesel

Markus Kiesel ist studierter Maschinenbauingenieur (M.Eng.) Seit 2016 ist er als Unternehmer im Bereich der industriellen grafischen Simulation tätig. Mit seinem Team entwickelt er Anwendungen welche weltweite Verbreitung finden.



Kögel, Alexander

Alexander Kögel studierte Medientechnik (B.Eng.) und Medienmanagement (M.Eng) an der HTWK Leipzig. Seit 2017 ist er an der TU Chemnitz als wissenschaftlicher Mitarbeiter beschäftigt, mit dem Fokus auf VR-Anwendungen, Einflussgrößen auf das Presence-Empfinden in virtuellen Umgebungen und der Erschließung der technologischen und wirtschaftlichen Potenziale der Digitalisierung, Vernetzung und Anwendung von Industrie 4.0.