

Morphing Displays: Fenster in einem virtuellen Interaktionsraum

Einleitung

Die Einbindung interaktiver Widgets in den virtuellen Raum eines dynamischen Displays stellt einen innovativen Ansatz zur Weiterentwicklung moderner Mensch-Maschine-Schnittstellen (HMI) dar. Displays, die sich in Form und im Raum bewegen können, ermöglichen durch ihre dynamische und adaptive Struktur eine neuartige Interaktion zwischen Nutzer und System, die sowohl technologische als kognitive Aspekte berücksichtigt. Wir haben Potenziale und Herausforderungen dieser Technologie untersucht und sind dabei auf folgende Schlussfolgerungen gekommen.

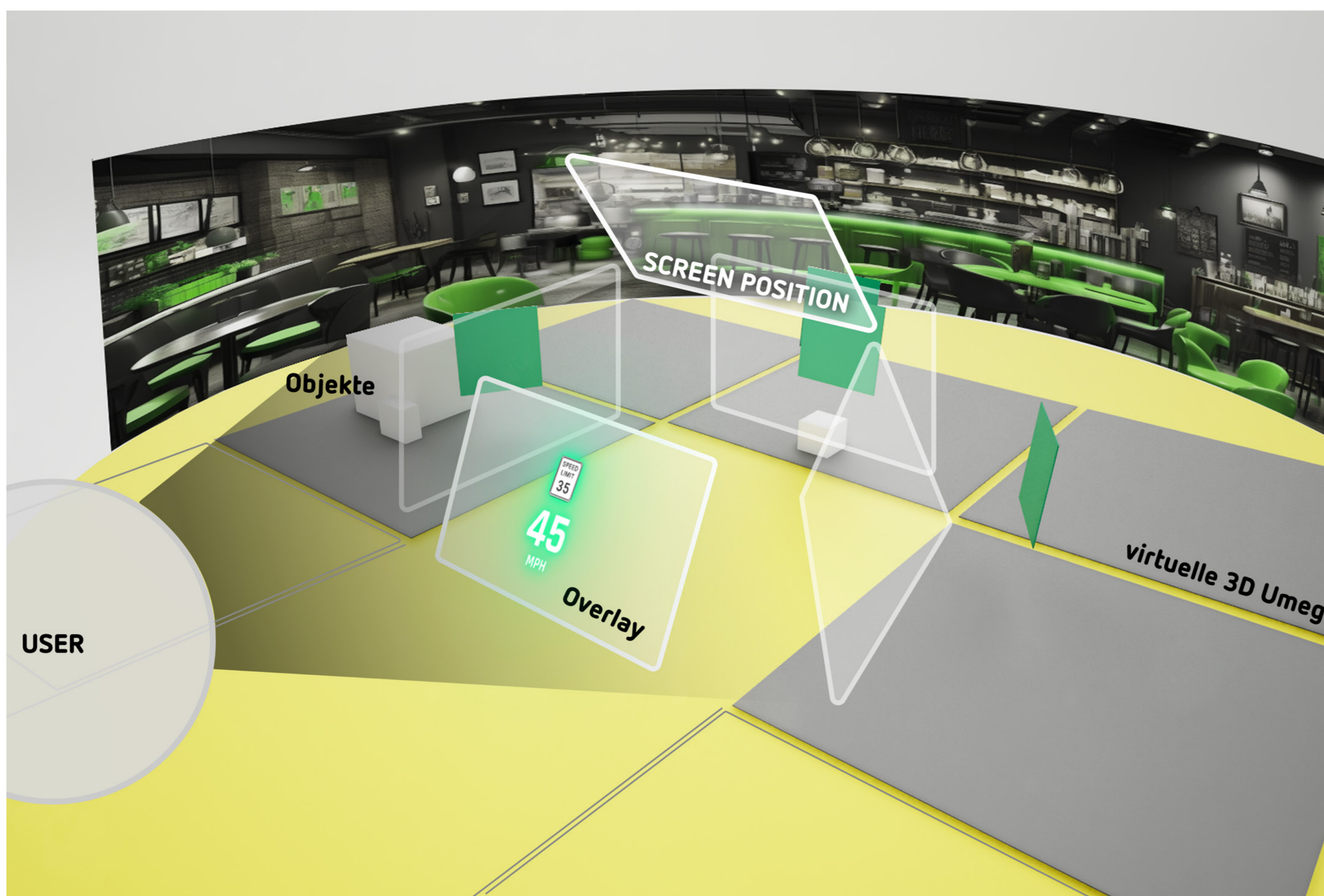
Zwischen Immersion und Beobachtung: Fließende adaptive Grenzen zwischen immersiver und non-immersiver VR

Unter klassischer VR versteht man VR-Brillen, die vollständig immersiv sind und bei denen die virtuelle Welt strikt an die Kopfbewegungen des Nutzers gekoppelt ist. Dies erzeugt ein starkes Präsenzgefühl, bringt aber auch physische Einschränkungen mit sich, wie Isolation, Motion Sickness oder Ermüdungserscheinungen bei längerer Nutzung.

Unsere Displaylösung vereint zwei Modi in einem System: den immersiven Modus „Ich bin in der Welt“ und den non-immersiven Modus „Ich schaue auf eine Welt“. Dadurch ermöglichen wir eine stufenlos anpassbare Immersion und fördern die natürliche Integration in die reale Umgebung. Unser Ansatz ergänzt klassische VR, indem er die Immersion kontrollierbar macht und die Grenzen zwischen immersiver und non-immersiver VR fließend gestaltet.

Obwohl unsere 3D-Umgebung und klassische VR denselben virtuellen Raum mit derselben Engine nutzen, unterscheiden sie sich in Nutzung und Wahrnehmung deutlich. Klassische VR setzt auf ein weites Sichtfeld und ein Head-Mounted Display, das den Nutzer vollständig von der realen Umgebung isoliert. Unser System arbeitet mit einem physisch begrenzten Display, das im immersiven Modus den Nutzer in die virtuelle Welt eintauchen lässt, aber jederzeit in den non-immersiven Modus wechseln kann, wodurch das Display zu einem „Fenster“ in die virtuelle Welt wird.

Dies ermöglicht flexible Interaktionen und bewahrt die Verbindung zur realen Umgebung. Der Nutzer kann jederzeit zwischen Eintauchen und Beobachten wechseln – was neue Anwendungs- und Interaktionsmöglichkeiten vor dem Display sowie innerhalb der 3D-Umgebung eröffnet.

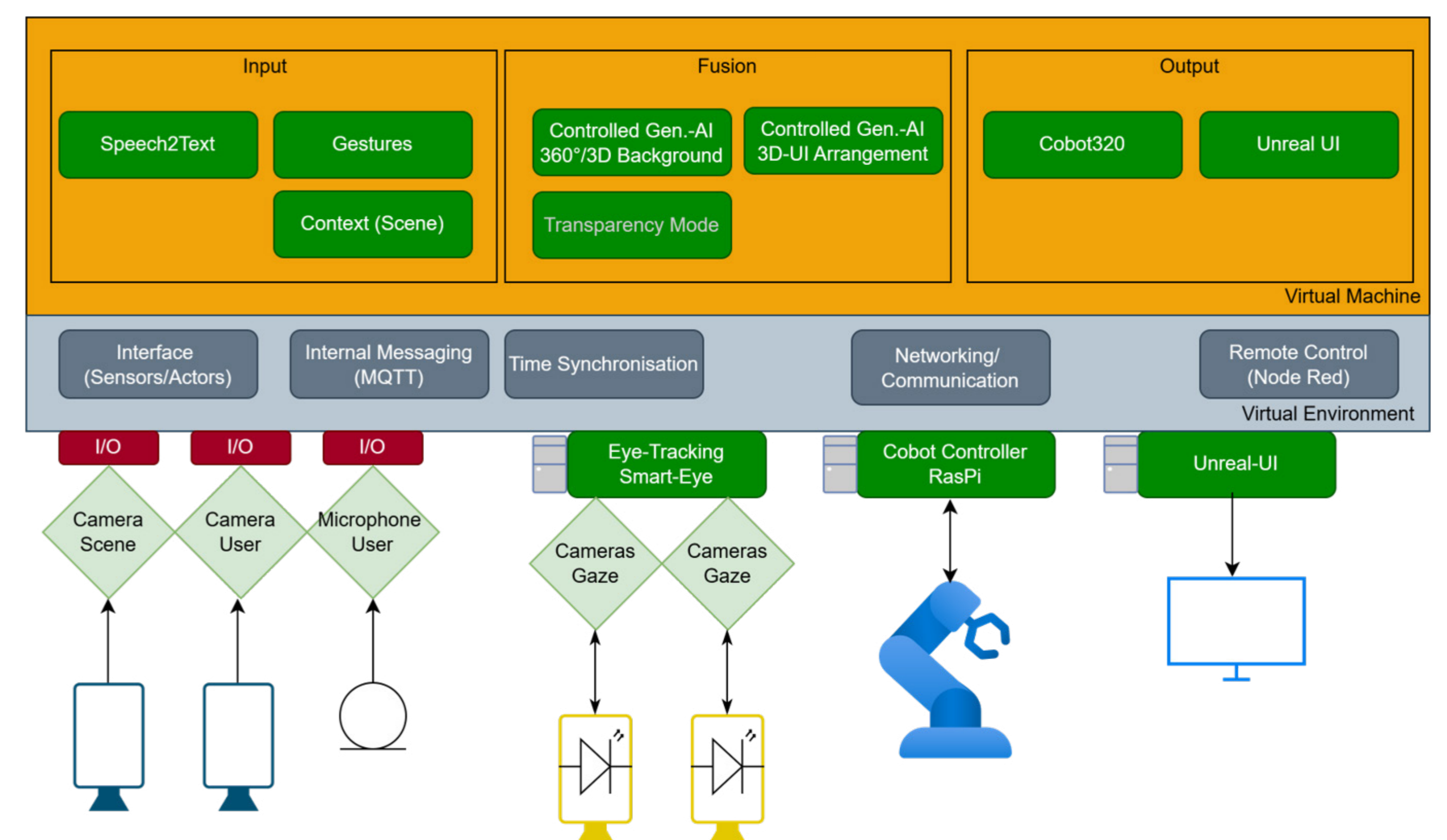


Szenengraph-basierte Architektur zur Steuerung virtueller 3D-Umgebungen in Unreal Engine

In Unreal, wird typischerweise ein Szenegraph für virtuelle 3D-Umgebungen verwendet. Dieser Szenegraph dient als zentraler Knotenpunkt, an dem die Display-Position in Form einer virtuellen Kamera und alle Widgets hinzugefügt werden. Darüber hinaus können verschiedene andere Elemente wie 3D-Hintergründe, 3D-Objekte, Videostreams, Lichtquellen und Animationen in den Szenegraph integriert werden.

Die KI kann die Umgebung über eine MQTT-Schnittstelle erstellen und anpassen, wobei die Kameraposition ebenfalls über MQTT angebunden ist und sich der Display-Position anpasst. Durch die gemeinsame Nutzung des Szenegraphs für die Kamera und Widgets führen Display-Bewegungen gleichzeitig immersive Widget-Animationen aus. Um eine flüssige Animation zu gewährleisten, werden Kamera-Transformationen durch lineare Interpolation berechnet.

Systemüberblick



Das System ist in eine virtuelle Verwaltungsumgebung sowie eine separate virtuelle Maschine für spezifische Teilaufgaben gegliedert. Alle Softwarekomponenten sind containerisiert, um Modularität, Skalierbarkeit und Wartbarkeit sicherzustellen. Rechenintensive und latenzkritische Funktionen (u. a. Eye-Tracking, Robotersteuerung, Bild- und Anzeigeverarbeitung) sind auf spezialisierte Rechner ausgelagert. Das Gesamtsystem gliedert sich in drei Funktionsbereiche:

1. Eingabe

Die Eingabeebene erfasst Nutzerinteraktionen und Kontextinformationen. Primäre Interaktionsmodalität ist Sprache, ergänzt durch Eye-Tracking und Gestenerkennung zur Disambiguierung und Validierung. Eine Szenenkamera liefert zusätzliche Kontextdaten, wie die Erfassung einzelner Objekte, z.B. Auto oder der Gesamtsituation, z.B. Stau.

2. Fusion

In der Fusionsschicht werden multimodale Eingaben ergänzt mit Domänenwissen (z. B. Markenrichtlinien, Designvorgaben) integriert. Die resultierenden Informationen steuern sowohl die Cobot-Bewegungsplanung als auch die generative Bildsynthese in der Unreal Engine. Details zur Fusionslogik sind in Poster zum 'Generatives HMI' beschrieben.

3. Ausgabe

Die Ausgabe erfolgt über ein Display, das entweder durch einen Cobot oder manuell vom Nutzer positioniert und ausgerichtet wird. Position und Orientierung werden in Echtzeit erfasst und zurückgemeldet und der Bildschirminhalt entsprechend dynamisch und adaptiv angepasst.