De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen

School voor Informatietechnologie

master in de informatica

Masterthesis

ModulARboard: Exploring AR Dashboards for Immersive Analytics

Lise Verbeeck

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Gustavo Alberto ROVELO RUIZ

COPROMOTOR :
Prof. dr. Davy VANACKEN

BEGELEIDER :
De heer Jarne THYS

2024
2025

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

ModulARboard: Exploring AR Dashboards for Immersive Analytics

Lise Verbeeck

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Gustavo Alberto ROVELO RUIZ

COPROMOTOR :

Prof. dr. Davy VANACKEN

BEGELEIDER :

De heer Jarne THYS

Acknowledgments

First of all, I want to thank my promoter, Prof. dr. Gustavo Rovelo Ruiz and my co-promotor, Prof. dr. Davy Vanacken, for allowing me to work on this topic with them. I also want to thank my supervisor, Jarne Thys, for always being so quick to respond when I requested feedback. They always provided me with their constructive feedback, allowing me to present a thesis of a higher quality.

I want to thank my boyfriend, Jan, for always supporting me, for being my rubber ducky when things weren't working out, for listening to me when I came up with yet another definition and for philosophising with me about the specifics of AR. I also want to thank him for proofreading my thesis, despite not being interested in the topic. I really appreciate the effort and the feedback he provided me with.

Lastly, I want to thank my parents for being so supportive and keeping me motivated during the course of my thesis. They also had to bear with me when I was philosophising about AR yet again. I appreciate that they kept listening to me and tried to help, even if they weren't really familiar with the topic.

I want to thank everyone for being so patient and understanding with me during the course of this thesis when life was sending many difficulties my way. I really appreciate all the support that I got in order to finish this thesis and deliver work that I can be proud of.

This research was made possible with support from the MAXVR-INFRA project, a scalable and flexible infrastructure that facilitates the transition to digital-physical work environments. The MAXVR-INFRA project is funded by the European Union - NextGenerationEU and the Flemish Government.

The icons in the dashboard, as seen in the screenshots, are made by Freepik, wanicon, berkahicon, Iconic Artisan and Nikita Golubev from `www.flaticon.com`.

Samenvatting

Deze thesis onderzoekt hoe **Augmented Reality (AR)** kan worden ingezet om traditionele datadashboards te herontwerpen en levert daarmee een bijdrage aan het opkomende onderzoeksveld van **Immersive Analytics (IA)**. IA, geïntroduceerd in 2015, heeft als doel data-analyse intuïtiever, interactiever en toegankelijker te maken door gebruik te maken van immersieve technologieën zoals AR en Virtual Reality (VR). Als jong onderzoeksgebied kent IA nog veel open vragen en actieve onderzoekslijnen, wat het een vruchtbaar domein maakt voor verdere verkenning. Deze thesis draagt hieraan bij door het potentieel van AR-dashboards binnen IA te onderzoeken.

Om dit te doen, is het eerst noodzakelijk om te begrijpen wat AR precies is. Zonder een duidelijke definitie is het moeilijk te bepalen of een systeem echt tot het AR-domein behoort. Daarom begint dit werk met een kritische analyse van bestaande AR-definities, waarbij de tekortkomingen worden blootgelegd en een meer inclusieve en precieze definitie wordt voorgesteld om de ontwikkeling van immersieve dashboards te ondersteunen. Vervolgens onderzoekt deze thesis hoe AR geïmplementeerd is in bestaande systemen door te analyseren hoe virtuele elementen typisch gepositioneerd worden in de fysieke ruimte. Deze ruimtelijke positionering is essentieel voor het creëren van betekenisvolle en bruikbare immersieve interfaces. Het onderzoek naar verankeringsmechanismen vormt dan ook een kernelement van deze studie.

Het doel is echter niet om traditionele dashboards simpelweg te kopiëren in AR. De ambitie is om een unieke en innovatieve AR-native ervaring te bieden die optimaal gebruik maakt van de mogelijkheden van het medium. In dit kader wordt ook onderzocht hoe personalisatie kan worden voorzien in immersieve dashboards, zodat gebruikers hun data-omgeving kunnen aanpassen aan hun specifieke behoeften. Deze verkenning vormt de basis voor de ontwikkeling van het prototype ModulARboard. Dit leidt tot de volgende onderzoeksvraag:
Hoe kan Augmented Reality worden ingezet om traditionele datadashboards te herontwerpen voor Immersive Analytics?

1. Wat valt onder Augmented Reality?
2. Welke technieken bestaan er of kunnen worden toegepast om visuele elementen in AR te positioneren?
3. Hoe kan personalisatie worden geïntroduceerd bij het herontwerpen van traditionele dashboards?

Gerelateerd werk

Begrijpen wat kwalificeert als Augmented Reality (AR) is een fundamenteel aspect van deze thesis. Traditioneel wordt AR gedefinieerd als het real-time projecteren van virtuele elementen op de echte wereld, geregistreerd in een 3D-ruimte. Een van de meest algemeen geaccepteerde definities, voorgesteld door Azuma [Azu97], beschrijft drie kernkenmerken: een combinatie van echt en virtueel, real-time interactiviteit en registratie in 3D. Daarnaast beschrijft de definitie van Milgram and Kishino [MK94] het Reality-Virtuality Continuüm en wat wordt verstaan

onder ‘echt’ of ‘virtueel’. Zij stellen dat iets ‘echt’ is wanneer het digitaal kan worden opgenomen en opnieuw gesynthetiseerd. Iets is ‘virtueel’ wanneer het gesimuleerd moet worden.

Hoewel deze definities de basis legden voor Augmented Reality, zijn ze enigszins verouderd geraakt door de snelle opkomst van nieuwe technologieën. Verschillende systemen bevinden zich in grijze zones omdat deze definities ze niet volledig adresseren. Bijvoorbeeld: een echt persoon die op afstand wordt geaugmenteerd in een andere kamer, kilometers verderop, wordt niet als AR beschouwd omdat de persoon wordt opgenomen en opnieuw gesynthetiseerd. Hierdoor is er geen combinatie meer van ‘echt’ en ‘virtueel’, wat betekent dat holografische telepresence op afstand geen AR is. Bovendien worden head-up displays die slechts gedeeltelijk reageren op gebruikersinput of 2D-interface-elementen die vastzitten aan het gezichtsveld van de gebruiker, mogelijk niet als AR beschouwd. Dit is afhankelijk van hoe strikt men de vereisten voor real-time interactiviteit of ruimtelijke registratie van Azuma [Azu97] interpreteert.

Als gevolg hiervan stelt deze thesis een verfijnde definitie van AR voor om duidelijkheid te verschaffen en aan te sluiten bij moderne gebruiksscenario’s. Deze behoudt de combinatie van echte en virtuele elementen, maar schrapt de strikte vereiste van 3D-registratie. Daarnaast introduceert de nieuwe definitie het begrip ruimtelijke nabijheid in het opname- en resyntheseproces om beter te bepalen wat kwalificeert als ‘echt’. Ter illustratie van deze nieuwe eis, neem het scenario waarin iemand aanwezig is bij het opnameproces en iemand anders bij het resyntheseproces. Als zij elkaar kunnen zien, ruiken, aanraken of proeven, of elkaar kunnen horen roepen, worden zij beschouwd als zijnde ongeveer op dezelfde locatie. Want zelfs zonder augmentatie zouden zij nog steeds in real-time met elkaar kunnen interageren. Wanneer zij elkaar echter niet kunnen waarnemen vanaf de opname- of resyntheselocatie, wordt dit nu als ‘virtueel’ beschouwd, omdat augmentatie nodig is om in real-time te kunnen interageren. Dit kan ook uitgebreid worden naar interactie tussen mensen en levenloze objecten.

Samenvattend omvat deze nieuwe definitie nu ook de eerder genoemde grijze-zone toepassingen en beantwoordt het het eerste deel van de onderzoeksvraag: *Wat valt onder AR?*. Deze bredere en meer onderbouwde definitie helpt randgevallen te integreren en ondersteunt de verschillende zintuiglijke en ruimtelijke behoeften van toepassingen binnen immersive analytics.

Gerelateerd werk: Design space

Om beter te begrijpen hoe AR doorgaans wordt geïmplementeerd, is er een design space geconstrueerd op basis van meer dan 140 beoordeelde AR-systemen. Dit bracht terugkerende patronen en overeenkomsten aan het licht, maar ook waar de systemen in verschillen van elkaar. Deze overeenkomsten werden geformaliseerd in zeven categorieën, waaronder het toepassingsgebied, de geaugmenteerde zintuigen, de displaytechniek, de mate van aanwezigheid, samenwerking, systeemtype en de verankeringsmechanismen. In totaal zijn er 22 subcategorieën die aangeven waarin de implementaties licht verschillen.

De eerste categorie, **toepassingsgebied (application area)**, verwijst naar de context waarin het AR-systeem wordt gebruikt, zoals in de industrie, het onderwijs, entertainment of wetenschappelijk onderzoek. Dit helpt bij het classificeren van het doel en domeinspecifieke eisen van een bepaalde toepassing. Ten tweede beschrijft de categorie **zintuigen (senses)** de soorten sensorische modaliteiten die betrokken zijn bij de AR-ervaring. Hoewel visuele augmentatie het meest voorkomt, is het ook mogelijk om andere zintuigen te versterken (gehoor, tast, reuk en smaak). Bijvoorbeeld, aanraking kan worden geaugmenteerd via haptische feedback en gehoor via geluiden. Ten derde identificeert de categorie **displaytechniek (display technique)** de hardware-setup die wordt gebruikt om de AR-inhoud weer te geven. Dit omvat ‘handheld’ apparaten zoals smartphones en tablets, head-mounted displays (HMD’s) zoals de Magic Leap 2 of HoloLens, en projectie-gebaseerde systemen.

De vierde categorie, **mate van aanwezigheid (Extent of Presence)**, onderscheidt egocentrische en exocentrische systemen. Egocentrische systemen hebben een perspectief waarbij de gebruiker zich binnen de visualisaties bevindt. De gebruiker voelt zich omringd door virtuele inhoud. Exocentrische systemen presenteren AR-inhoud vanuit een vaste ruimtelijke positie in de omgeving. De gebruiker kan de visualisaties van buitenaf bekijken. Dit is echter een lastig onderwerp om te categoriseren, vermits het vooral een gevoelswaarde is. Ten vijfde kijkt de categorie **samenwerking (collaboration)** of het systeem ontworpen is voor een enkele gebruiker of meerdere gebruikers, die met dezelfde AR-inhoud interageren, lokaal of op afstand. De zesde categorie, **systeem (system)**, categoriseert of de AR-ervaring losstaand is of gecombineerd met andere instanties of systemen. Bijvoorbeeld, sommige AR-systemen voorzien een tablet bovenop een HMD om extra of meer gedetailleerde informatie aan te bieden dan alleen op de HMD mogelijk zou zijn.

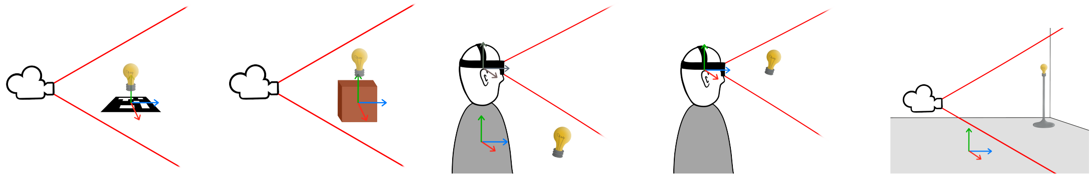
De laatste en belangrijkste categorie voor deze thesis betreft de **verankeringsmechanismen (anchoring mechanisms)**. Deze definiëren hoe virtuele elementen in de ruimte worden gepositioneerd. Dit omvat technieken zoals marker-based anchoring, object-relative anchoring, observer-relative anchoring, view frustum anchoring en spatial anchoring. Deze categorie is vooral belangrijk omdat deze direct aansluit bij de tweede onderzoeksvraag van de thesis: *Welke technieken bestaan er/kunnen worden toegepast om visuele elementen in AR te positioneren?*

Marker-based anchoring koppelt virtuele elementen aan een fysieke marker in de omgeving, waarbij het coördinatensysteem van de marker wordt gebruikt voor de positionering (Figuur 1a). Object-relative anchoring verbindt virtuele inhoud aan een fysiek object. Hierbij vormt het lokale coördinatensysteem van het object het referentiekader (Figuur 1b). Dit verschilt van marker-based anchoring omdat er geen eenvoudig te herkennen marker nodig is. In de plaats daarvan kan het aan elk object worden verankerd. Vervolgens koppelt observer-relative anchoring virtuele objecten aan de positie of het lichaam van de gebruiker. Hierbij wordt een constante afstand aangehouden ten opzicht van het coördinatensysteem van de gebruiker, onafhankelijk van de positie of oriëntatie van het hoofd van de gebruiker (Figuur 1c). Wanneer de gebruiker met het hele lichaam draait, draait het virtuele item mee om op dezelfde positie te blijven ten opzichte van de gebruiker. Draait de gebruiker alleen het hoofd, dan draait het item niet mee, omdat het coördinatensysteem wordt bepaald door het middelpunt van de gebruiker. View frustum anchoring plaatst inhoud in het gezichtsveld van de gebruiker, op een vaste positie voor de camera/het hoofd (Figuur 1d). Bij dit type verankering volgt de visualisatie altijd de beweging van het hoofd/de camera. Tenslotte maakt spatial anchoring gebruik van het wereldcoördinatensysteem van het apparaat. Hierdoor wordt inhoud persistent op een fysieke locatie geplaatst (Figuur 1e).

Deze zeven categorieën vormen een uitgebreide design space die niet alleen helpt om het AR-landschap te begrijpen, maar ook de scope van onze proof-of-concept applicatie ModulARboard afbakent. Dit gebeurt door te focussen op een geselecteerde set subcategorieën. ModulARboard is een AR-applicatie die zich richt op de augmentatie van het zicht via een standalone HMD met een egocentrisch perspectief. Het is bedoeld voor individueel gebruik binnen het toepassingsgebied presentatie & visualisatie. Bovendien probeert het alle verankeringsmechanismen die besproken zijn, uit te testen. Deze selectie subcategorieën is onderaan te zien in de tabel in Figuur 2. Deze tabel toont veertien geselecteerde studies met een AR-systeem, met ModulARboard apart onderaan. De veertien studies werden geëvalueerd ten opzichte van de zeven categorieën en hun subcategorieën om de populairste implementatietechnieken te identificeren.

ModulARboard

De motivatie voor ModulARboard komt voort uit de beperkingen van traditionele dashboards, namelijk vaste layouts, beperkte schermruimte en een gebrek aan contextueel bewustzijn. AR biedt de mogelijkheid om te experimenteren met hoe data beter gerangschikt, genavigeerd en begrepen kan worden door deze te positioneren in de omgeving van de gebruiker. Hierdoor is ModulARboard ontworpen als een modulair en uitbreidbaar dashboardsysteem dat gebruikers in staat stelt om visualisaties te bouwen, personaliseren en te positioneren in de fysieke ruimte.



(a) Marker-based (b) Object-relative (c) Observer-relative (d) View frustum (e) Spatial

Figure 1: Illustratie van de verankeringsmechanismen in functie van hun coördinatensysteem, zoals geïllustreerd met de driekleuren-assen.
(a) Marker-based anchoring: het coördinatensysteem wordt bepaald door de pose (positie & oriëntatie) van de marker. Wanneer de marker beweegt of roteert, bewegen en roteren alle eraan gekoppelde items mee.
(b) Object-relative anchoring: het coördinatensysteem wordt bepaald door de pose van het object.
(c) Observer-relative anchoring: het coördinatensysteem wordt bepaald door de pose van de waarnemer, onafhankelijk van het hoofd. Wanneer de camera/het hoofd roteert, blijft het item op dezelfde plek. Wanneer de waarnemer echter het lichaam beweegt of roteert, bewegen en roteren de verankerde items mee.
(d) View frustum anchoring: het coördinatensysteem wordt bepaald door de pose van de camera/het hoofd.
(e) Spatial anchoring: het coördinatensysteem wordt bepaald door de fysieke ruimte en zal niet bewegen of roteren.

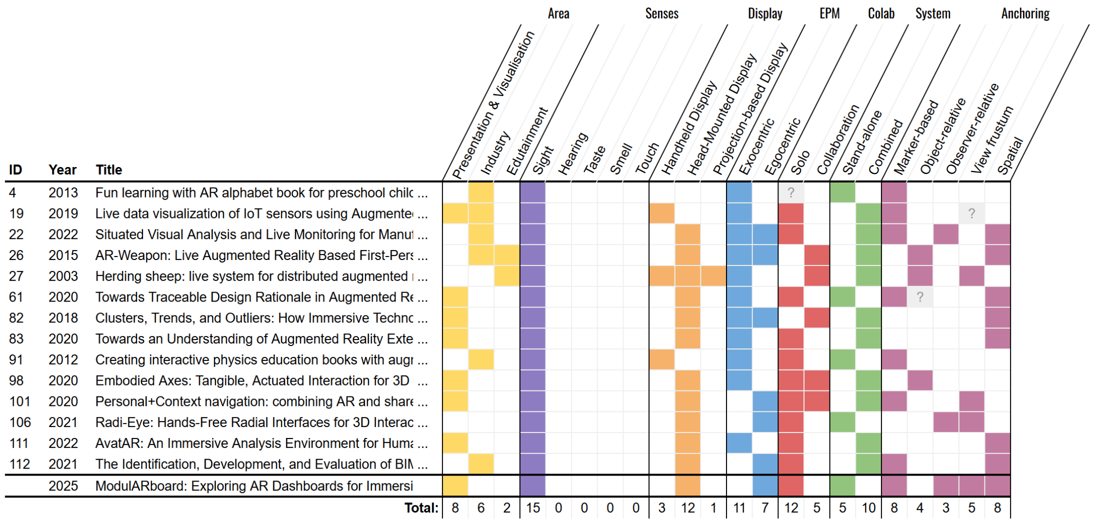


Figure 2: Deze tabel geeft een samenvatting van de evaluatie van veertien AR-studies ten opzicht van de zeven categorieën van de design space. Dit zijn de beoordeelde bronnen:
4: [RMS13], 19: [NM19], 22: [Bec+22], 26: [Zhu+15], 27: [Mac+03], 61: [CCV20], 82: [But+18], 83: [Wan+20], 91: [Dün+12], 98: [Cor+20], 101: [Jam+20], 106: [Sid+21], 111: [Rei+22] & 112: [May+22]

De hoofdapplicatiestroom begint met het importeren van een dashboardbestand in JSON-formaat. Bij het inladen wordt elk onderdeel weergegeven met een modulair pictogram in het hoofd-dashboard. De gebruiker kan deze componenten vrij selecteren en groeperen om het visualisatieproces te starten. Dit kan op verschillende manieren gebeuren. Een visualisatie kan direct worden opgevraagd via de component-tegel in het hoofd-dashboard, of door eerst het label van de component te verankeren en vervolgens het visualisatiemenu te openen. Het is ook mogelijk om het samengestelde visualisatiemenu (voor meerdere componenten) te openen door het menu direct in het hoofd-dashboard te openen, of eerst componenten te groeperen en het groepslabel te verankeren. Het samengestelde visualisatiemenu kan dan geopend worden via het groepslabel en wordt automatisch geïnitialiseerd door alle componenten van de groep te selecteren. Dit maakt het eenvoudiger om meerdere visualisaties te vernieuwen of op te vragen.

In het samengestelde visualisatiemenu is het mogelijk om meerdere componenten te selecteren. Elke keer dat een component geselecteerd of gedeselecteerd wordt, evalueert het systeem de soorten data-velden binnen de selectie. Het systeem filtert en suggereert dan dynamisch relevante data-attributen voor de huidige groep. Deze stap wordt uitgevoerd door het ‘adaptive data loading system’ en zorgt ervoor dat de gebruiker alleen logische en compatibele opties krijgt voorgesteld, waardoor informatie-overload en mogelijke fouten worden verminderd. De gebruiker kan vervolgens een of meerdere data-opties selecteren om de aanbevolen visualisaties op te vragen die door het ‘visualisation recommendation system’ worden gegenereerd. Dit systeem houdt rekening met de structuur, kardinaliteit en semantiek van de data om visualisaties zoals staafdiagrammen, taartdiagrammen, radardiagrammen of zelfs gecombineerde visualisaties voor te stellen. Deze aanbeveling past zich aan zowel enkelvoudige dataseries als gegroepeerde data aan, wat een flexibele en responsieve workflow mogelijk maakt. Samen vormen deze systemen het ‘adaptive visualisation recommendation system’, dat verantwoordelijk is voor het genereren van visualisaties binnen ModulARboard. De functionaliteit kan eenvoudig worden uitgebreid door af te leiden van de meegeleverde basisklassen om extra datatypes en/of visualisaties toe te voegen.

Zodra de gebruiker tevreden is met een visualisatie, kan hij een type verankeringsmechanisme kiezen om deze in de fysieke omgeving te positioneren. ModulARboard ondersteunt vier van de vijf verankeringsmechanismen zoals besproken in de design space. Er is besloten om object-relative anchoring niet te implementeren, vermits deze sterk lijkt op marker-based anchoring. Daarnaast is observer-relative anchoring een benadering, aangezien er geen sensor beschikbaar was om de lichaamspositie onafhankelijk van de hoofdpositie te volgen. Wanneer de gebruiker een verankeringsmechanisme heeft gekozen, kan de visualisatie worden geschaald en verplaatst ten opzichte van het coördinatensysteem dat door het verankeringsmechanisme wordt gedefinieerd. Deze mogelijkheden bieden opties tot aanpassingen afhankelijk van het beoogde gebruik, of het nu gaat om statische datawanden of mobiele dashboards.

Figuur 3 toont ModulARboard in een reële situatie (printer-lab). In de screenshot is het samengestelde visualisatiemenu geopend vanuit het label van de 3D-printergroep, wat resulteert in de selectie van alle 3D-printers in het linker paneel. In het middelste paneel kan de gebruiker de dynamisch geladen data-opties selecteren, wat leidt tot het genereren van aanbevolen visualisaties in het rechter paneel. In dit scenario is de data-optie ‘times used’ geselecteerd voor negen printers in totaal, wat resulteert in vier verschillende visualisaties voor het discrete datatype. Er is ook een gecombineerde ringdiagramvisualisatie verankerd aan het view frustum van de gebruiker linksboven. Deze toont de filamentvoorraad van acht printers (met uitzondering van de printer met vier spoelen). Ten slotte is het label van de robotarm verankerd in de ruimte, samen met een numerieke visualisatie naast de fysieke robotarm.

ModulARboard is ontwikkeld voor de Magic Leap 2 in Unity, met aanvullende pakketten voor JSON-conversie, een XR-toetsenbord en 2D-visualisaties. De gebruikte data is gegenereerd door OpenAI’s ChatGPT en de pictogrammen in het dashboard zijn gedownload van Flaticon.

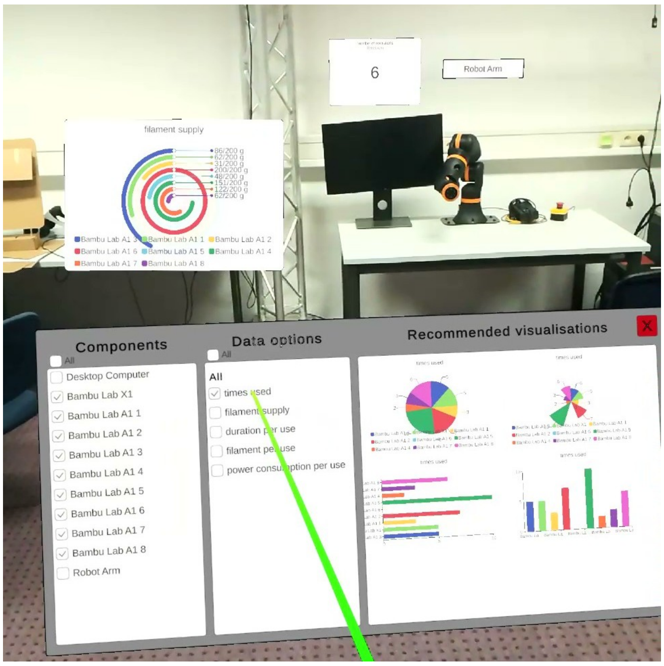


Figure 3: Dit is een screenshot van ModulARboard in een reële situatie. Het illustreert het samengestelde visualisatiemenu met alle 3D-printers geselecteerd en de data-optie ‘times used’ in het middelste paneel, wat resulteert in een samengestelde visualisatie aan de rechterkant. Er is ook een samengestelde ringdiagramvisualisatie verankerd aan het view frustum linksboven. Tot slot wordt een label en een numerieke visualisatie geïllustreerd die naast de robotarm in de ruimte verankerd zijn.

Use cases

ModulARboard toont flexibiliteit in verschillende contexten door zowel traditionele als geavanceerde use cases te ondersteunen. Het stelt gebruikers in staat om datavisualisaties en labels direct te verankeren aan apparaten of omgevingen, zoals in ‘smart homes’, waardoor intuïtieve ruimtelijke associaties ontstaan. Het marker-gebaseerde verankeringssysteem maakt dynamische opstellingen mogelijk. Het uitbreidbare framework van ModulARboard maakt het eenvoudig om aangepaste datatypes en visualisaties toe te voegen, afgestemd op specifieke organisatorische behoeften. Bij grootschalige implementaties, zoals fabrieken of industriële omgevingen, maakt de groepsfunctionaliteit het mogelijk om volledige sets visualisaties tegelijkertijd bij te werken. Op deze manier worden complexe monitoringtaken ondersteund met minimale inspanning van de gebruiker. Tot slot kunnen de verankeringsmechanismen ook op een meer abstracte manier gebruikt worden om ook andere soorten virtuele content te organiseren en ruimtelijk te rangschikken. Op deze manier kunnen de verankeringsmechanismen dienen voor meer doeleinden, ook buiten ModulARboard.

Toekomstig werk

Het uitvoeren van een gebruikerstudie zou empirische validatie bieden en aanvullende patronen onthullen op het gebied van bruikbaarheid, effectiviteit en gebruikersvoorkeuren. Verder onderzoek naar visualisatietechnieken kan ook de personalisatie verbeteren, bijvoorbeeld door gebruikers in staat te stellen alternatieve aggregaties zoals gemiddelden of medianen op te vragen, of door histogrammen toe te voegen voor distributieanalyse. Naast individuele aanpassing biedt het ondersteunen van collaboratieve workflows in modulaire AR-dashboards een veelbelovende verdere richting.

Conclusie

Deze thesis onderzocht hoe Augmented Reality kan worden ingezet om traditionele datadashboards te herontwerpen voor Immersive Analytics. Hiervoor was het eerst belangrijk om te definiëren wat kwalificeert als AR. Dit werd aangepakt door een analyse van bestaande AR-definities, waarbij ambiguïteiten en beperkingen in oudere kaders aan het licht kwamen. Er werd een nieuwe, meer inclusieve definitie voorgesteld, waarbij de vereiste voor 3D-registratie werd verwijderd en ruimtelijke nabijheid als criterium werd geïntroduceerd. Deze herdefinitie maakt bredere toepassingen mogelijk binnen immersive analytics-contexten.

Voor de tweede deelvraag, met betrekking tot de verschillende mogelijkheden om virtuele elementen in fysieke ruimte te positioneren, werd een uitgebreide analyse van de design space uitgevoerd. Dit leidde tot de identificatie van vijf verankeringsmechanismen die in AR-systemen worden gebruikt. De implementatie van ModulARboard demonstreerde vier daarvan: marker-based, spatial, view frustum en observer-relative anchoring, waarbij object-relative anchoring werd uitgesloten. Elk mechanisme heeft zijn eigen sterke en zwakke punten. Marker-based en object-relative anchoring zijn het meest geschikt voor dynamische opstellingen. Object-relative anchoring kan echter een schonere implementatie bieden zonder de noodzaak van markers. View frustum en observer-relative anchoring kunnen nuttig zijn om belangrijke informatie in het oog te houden, hoewel deze alleen gebruikt zouden moeten worden voor een select aantal visualisaties. Dit is omdat het het zicht van de gebruiker kan belemmeren en gevaarlijke situaties kan veroorzaken. Tot slot is spatial anchoring ideaal voor statische opstellingen in tegenstelling tot marker-based en object-relative anchoring. Spatial anchoring vereist echter meshing-algoritmen om zich te oriënteren in de fysieke ruimte. Zonder deze algoritmen werkt spatial anchoring niet over meerdere sessies heen. Over het algemeen is het aan te raden om spaarzaam om te gaan met het aantal verankerde items, omdat het anders rommelig en overweldigend kan worden in de augmented omgeving.

De derde deelvraag richtte zich op personalisatie. ModulARboard biedt verschillende aanpassingsmogelijkheden aan: visualisaties kunnen van grootte worden veranderd en gegroepeerd, data-elementen kunnen geselecteerd worden om adaptieve grafieken te genereren, en gebruikers kunnen content positioneren met behulp van verschillende verankeringsstrategieën. Daarnaast ondersteunt het framework de integratie van aangepaste visualisaties en nieuwe datatypes, waardoor het zeer uitbreidbaar is. Deze functionaliteiten stellen gebruikers in staat dashboards te bouwen die aansluiten bij hun workflow, omgeving en visuele voorkeuren. Samenvattend kan Augmented Reality worden ingezet om traditionele datadashboards te herontwerpen door de implementatie van verankeringsmechanismen en het aanbieden van personalisatiemogelijkheden aan de gebruiker.

Summary

This thesis explores how **Augmented Reality (AR)** can be leveraged to reimagine traditional data dashboards, contributing to the emerging research field of **Immersive Analytics (IA)**. IA, introduced in 2015, aims to make data analysis more intuitive, interactive, and accessible by utilising immersive technologies such as AR and Virtual Reality (VR). As a young research field, IA still holds many unanswered questions and active research avenues, which makes it a fertile domain for exploration. This thesis contributes to this field by examining the potential of AR dashboards within IA.

To do so, it is first necessary to understand what AR is. Without a clear definition, it is difficult to determine whether a system truly belongs to the AR domain. Therefore, this work begins by critically examining existing definitions of AR, identifying their shortcomings, and proposing a more inclusive and precise definition to guide the development of immersive dashboards. Subsequently, this thesis investigates how AR is implemented in existing systems by analysing how virtual elements are typically positioned within physical space. This spatial arrangement is crucial to creating meaningful and usable immersive interfaces. The study of anchoring mechanisms and positioning techniques forms a core part of the research.

However, the goal is not simply to replicate traditional dashboards in AR. Instead, the ambition is to provide a unique and innovative AR-native experience that leverages the medium's full potential. To this end, the thesis also explores how customisation can be introduced into immersive dashboards, allowing users to personalise their data environments and adapt visualisation layouts to their specific needs. This exploration forms the foundation for the development of the proof-of-concept application ModulARboard. This results in the following research questions:
How can Augmented Reality be leveraged to reimagine traditional data dashboards for Immersive Analytics?

1. What qualifies as Augmented Reality?
2. What techniques exist/can be applied to position visual elements in AR?
3. How can customisability be introduced when reimagining traditional dashboards?

Related work

Understanding what qualifies as Augmented Reality (AR) is a foundational aspect of this thesis. Traditionally, AR has been defined as the superimposition of virtual elements onto the real world in real time and registered in 3D space. One of the most widely accepted definitions, proposed by Azuma [Azu97], outlines three core characteristics: a combination of real and virtual, real-time interactivity, and registration in 3D. On top of that, the definition of Milgram and Kishino [MK94] describes the Reality-Virtuality Continuum and what is defined as 'real' or 'virtual'. They claim that something is considered 'real' when it can be sampled and resynthesised. On the other hand, something is 'virtual' when it has to be simulated.

While these definitions laid the foundation of Augmented Reality, they have become slightly outdated with the rapid rise of new technologies. Several systems fall into grey zones, as these

definitions do not fully address them. For example, a real person who is augmented in another room, kilometres away, is not considered AR since the person is sampled and resynthesised. Therefore, there is no longer a combination of ‘real’ and ‘virtual’, which means that remote holographic telepresence is not AR. Moreover, head-up displays that only partially respond to user input or 2D interface elements fixed to the user’s field of view might not be considered AR depending on how strictly one interprets the real-time interactivity or spatial registration requirements of Azuma [Azu97].

As a result, this thesis proposes a refined definition of AR to provide clarity and align with modern use cases. It maintains the combination of real and virtual elements but removes the strict requirement for 3D registration. Additionally, it introduces the notion of spatial proximity in the sampling and resynthesis process to determine better what qualifies as ‘real’. To illustrate this new requirement, consider the scenario where someone is present at the sampling process and someone at the resynthesisation process. If they can see, smell, touch or taste each other or can hear one another yell, they are considered to be in approximately the same location. Because even without augmentation, they would still be able to interact with each other in real-time. Consequently, when they cannot perceive one another from the sampling or resynthesisation location, it is now considered ‘virtual’. Because augmentation is required in order to be able to interact with each other in real-time. This can be extended for interaction between people and inanimate objects as well. In conclusion, this new definition now includes the grey-zone applications mentioned previously and answers the first part of the research question: *What qualifies as AR?*. This broader and more grounded definition helps incorporate edge cases and supports the varied sensory and spatial needs of immersive analytics applications.

Related work: Design space

To gain a better understanding of how AR is typically implemented, a design space was constructed based on over 140 reviewed AR systems. This revealed recurring patterns and similarities, but also highlighted where key differences emerge. These similarities were formalised into seven categories, including the application area, augmented senses, display technique, extent of presence, collaboration, system type and the anchoring mechanisms. In total, there are 22 subcategories that define where they slightly differ in implementation.

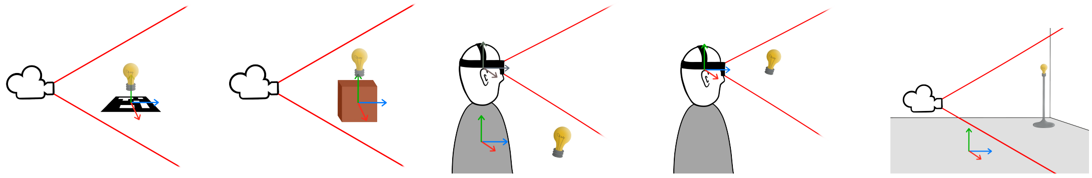
The first category, **application area**, refers to the context in which the AR system is used, such as in industry, education, entertainment, or scientific research. It helps classify the purpose and domain-specific requirements of a given application. Secondly, the **senses** category describes the types of sensory modalities involved in the AR experience. While visual augmentation is most common, it is also possible to augment the other senses (hearing, touch, smell and taste). For example, touch can be augmented through haptic feedback and hearing by sounds. Thirdly, the **display technique** identifies the hardware setup used to present the AR content. This includes handheld devices like smartphones and tablets, head-mounted displays (HMDs) such as the Magic Leap 2 or HoloLens, and projection-based systems.

The fourth category, **Extent of Presence**, distinguishes between egocentric and exocentric systems. Egocentric systems feature viewpoints where the user is located within the visualisations. The user feels surrounded by virtual content. Whereas exocentric systems present AR content from a fixed spatial position in the environment. The user can look at the visualisations from the outside. However, this is a difficult subject to categorise, since it is mostly a subjective measure. Fifthly, the **collaboration** category considers whether the system is designed for a single user or supports multiple users interacting with the same AR content, either locally or remotely. The sixth category, **system** categorises whether the AR experience is standalone or combined with other modalities or data systems. For example, some AR systems implement a tablet on top of an HMD in order to provide additional or more detailed information than would be possible on the HMD alone.

The final and most central category for this thesis concerns the **anchoring mechanisms**.

These define how virtual elements are positioned in space, including techniques like marker-based anchoring, object-relative anchoring, observer-relative anchoring, view frustum anchoring and spatial anchoring. This category is especially significant because it directly addresses the second research question of the thesis: *What techniques exist/can be applied to position visual elements in AR?*

Marker-based anchoring ties virtual content to a physical marker in the environment, using the marker's coordinate system for positioning (Figure 4a). Object-relative anchoring links virtual content to a tracked physical object, making the object's local coordinate system the reference frame (Figure 4b). It differs from marker-based anchoring in the sense that it does not need a recognisable marker, but can anchor it to any object. Subsequently, observer-relative anchoring attaches virtual objects to the user's position or body, maintaining a constant offset in the user coordinate system independent of the user's head (Figure 4c). When the user turns around with their full body, the virtual item will turn along to remain in the same position relative to the user. However, when the user only turns their head around, the item will not turn along, since the coordinate system is defined by the middle of the user. View frustum anchoring aligns content with the user's field of view, often placing it at a fixed position in front of the camera (Figure 4d). With this type of anchoring, the visualisations will always follow along with the movement of the head/camera. Lastly, spatial anchoring uses the device's world coordinate system, allowing content to be placed persistently in a physical location (Figure 4e).



(a) Marker-based **(b)** Object-relative **(c)** Observer-relative **(d)** View frustum **(e)** Spatial

Figure 4: Illustration of the anchoring mechanisms in function of their coordinate system, as illustrated by the three-coloured axes.
(a) Marker-based anchoring: the coordinate system is defined by the pose (position & orientation) of the marker. When the marker moves or rotates, all items anchored to it move and rotate accordingly.
(b) Object-relative anchoring: the coordinate system is defined by the pose of the object.
(c) Observer-relative anchoring: the coordinate system is defined by the pose of the observer, independent of the head. When the camera/head rotates around, the item will remain in the same spot. However, when the observer fully moves or rotates around, the anchored items will move and rotate accordingly.
(d) View frustum anchoring: the coordinate system is defined by the pose of the camera/head.
(e) Spatial anchoring: the coordinate system is defined by the physical space and will not move or rotate around.

Altogether, these seven categories form a comprehensive design space that not only aids in understanding the AR landscape but also helps define the scope of our proof-of-concept application ModulARboard. This is done by focusing on a chosen set of subcategories. ModulARboard is an AR application that focuses on the augmentation of sight through a stand-alone HMD with an egocentric view. It is intended for individual usage in the application area of presentation & visualisation. Moreover, it attempts to test out all the anchoring mechanisms as previously discussed. This selection of subcategories can be seen at the bottom of the table in Figure 5. This table illustrates fourteen selected studies that feature an AR system, along with ModulARboard separately at the bottom. The fourteen studies were evaluated against the seven categories and their subcategories in order to find the most popular implementation techniques.

			Area			Senses					Display			EPM		Colab		System		Anchoring				
ID	Year	Title	Presentation & Visualisation	Industry	Edutainment	Sight	Hearing	Taste	Smell	Touch	Handheld Display	Head-Mounted Display	Projection-based Display	Exocentric	Egocentric	Solo	Collaboration	Stand-alone	Combined	Marker-based	Object-relative	Observer-relative	View frustum	Spatial
4	2013	Fun learning with AR alphabet book for preschool chilc ...		X		X								X		?		X		X				
19	2019	Live data visualization of IoT sensors using Augmente ...	X	X		X					X			X		X			X	X			?	
22	2022	Situated Visual Analysis and Live Monitoring for Manut ...		X		X						X		X	X	X			X	X		X		X
26	2015	AR-Weapon: Live Augmented Reality Based First-Pers ...		X	X	X						X		X	X		X		X		X			X
27	2003	Herding sheep: live system for distributed augmented ...			X	X					X	X	X	X			X		X		X		X	
61	2020	Towards Traceable Design Rationale in Augmented Re ...	X			X						X		X		X		X		X	?			X
82	2018	Clusters, Trends, and Outliers: How Immersive Technc ...	X			X						X		X	X		X		X					X
83	2020	Towards an Understanding of Augmented Reality Exte ...	X			X						X		X		X			X					X
91	2012	Creating interactive physics education books with augı ...		X		X					X			X		X		X		X				
98	2020	Embodied Axes: Tangible, Actuated Interaction for 3D ...	X			X						X		X		X	X		X		X			
101	2020	Personal+Context navigation: combining AR and share ...	X			X						X			X	X	X		X	X			X	
106	2021	Radi-Eye: Hands-Free Radial Interfaces for 3D Interac ...				X						X			X	X		X				X	X	
111	2022	AvatAR: An Immersive Analysis Environment for Humı ...	X			X						X		X		X			X					X
112	2021	The Identification, Development, and Evaluation of BIN ...		X		X						X			X	X			X	X				X
	2025	ModulARboard: Exploring AR Dashboards for Immersi	X			X						X			X	X		X		X		X	X	X
		Total:	8	6	2	15	0	0	0	0	3	12	1	11	7	12	5	5	10	8	4	3	5	8

Figure 5: This table summarises the evaluation of fourteen AR studies against the seven categories of the design space. These are the evaluated sources:
4: [RMS13], 19: [NM19], 22: [Bec+22], 26: [Zhu+15], 27: [Mac+03], 61: [CCV20], 82: [But+18], 83: [Wan+20], 91: [Dün+12], 98: [Cor+20], 101: [Jam+20], 106: [Sid+21], 111: [Rei+22] & 112: [May+22]

ModulARboard

The motivation for ModulARboard stems from the limitations observed in traditional dashboards, namely, fixed layouts, constrained screen space, and a lack of contextual awareness. With AR, there is an opportunity to rethink how data can be arranged, navigated, and understood by placing it directly in the user's environment. ModulARboard was thus designed as a modular and extensible dashboard system that empowers users to construct, personalise, and position visualisations in physical space.

The main application flow begins with the user importing a dashboard file in JSON format. Upon loading, each component is represented as a modular icon in the main dashboard. The user can select and group these components freely to initiate the visualisation process. This can be achieved throughout various stages. A visualisation can be requested immediately through the component tile in the main dashboard or by first anchoring the component's label and then accessing its visualisation menu. It is also possible to access the composite visualisation menu (for multiple components) by opening the menu directly in the main dashboard or by first grouping components together and anchoring the label. The composite visualisation menu can then be accessed through the group label and will be initialised by selecting all the components of the group. This makes it easier to renew or request multiple visualisations.

In the composite visualisation menu, it is possible to select multiple components. Every time a component is selected or deselected, the system evaluates the types of data fields within the selection and dynamically filters and suggests data attributes that are relevant for the given grouping. This step is performed by the 'adaptive data loading system' and ensures that the user is only presented with logical and compatible options, reducing information overload and potential errors. The user can then select one or multiple data options again in order to request the recommended visualisations generated by the 'visualisation recommendation system'. It considers the structure, cardinality, and semantics of the data to suggest visualisations such as bar charts, pie charts, radar charts, or combined visualisations. This recommendation step adapts to both single data series and grouped data, enabling a flexible and responsive workflow. Both systems combined form the 'adaptive visualisation recommendation system', which is responsible for the visualisation generation within ModulARboard. Their functionality can be extended by deriving from the provided base classes in order to provide additional data types and/or visualisations.

Once the user is satisfied with a visualisation, they can choose a type of anchoring mechanism in order to position it in their physical environment. ModulARboard supports four of the five anchoring mechanisms as discussed in the design space. It was decided not to implement object-relative anchoring, since it highly resembles marker-based anchoring. On top of that, observer-relative anchoring is an approximation, since there was no sensor available to track the body's pose independently of the head's pose. After the selection of the anchoring mechanism, the visualisation can be rescaled and moved around relative to the coordinate system defined by the anchoring mechanism. These options provide adaptability depending on the intended use case, whether static data walls, mobile dashboards, or user-guided exploration.

Figure 6 showcases ModulARboard in a real-life scenario (printer lab). In the screenshot, the composite visualisation menu is opened from the 3D printer group label, resulting in the selection of all the 3D printers in the left panel. In the middle panel, the user can select the dynamically loaded data options, which results in the recommended visualisation generation in the right panel. In this scenario, the 'times used' data option is selected for nine printers in total, resulting in four different visualisations for the discrete data type. There is also a combined ring chart anchored to the user's view frustum at the top left, which displays the filament supply of eight printers (excluding the printer with four spools). Lastly, the label of the robot arm is anchored to space, along with a number visualisation alongside the physical robot arm.

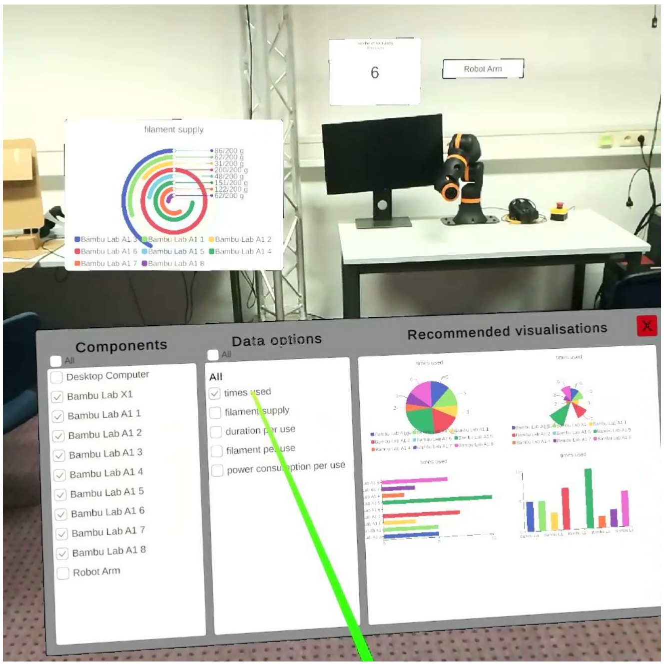


Figure 6: This is a screenshot of ModulARboard in a real-life scenario. It illustrates the composite visualisation menu with all 3D printers selected and the times used data option in the middle panel, resulting in composite visualisation on the right. There is also a composite ring chart visualisation anchored to the view frustum at the top left. Lastly, it illustrates a label and a single number visualisation anchored to space next to the robot arm.

ModulARboard was developed for the Magic Leap 2 in Unity with additional packages for JSON conversion, an XR Keyboard and 2D visualisations. The utilised data was generated by OpenAI's ChatGPT, and the icons in the dashboard were downloaded from Flaticon.

Use cases

ModulARboard demonstrates flexibility across a variety of contexts by supporting both traditional and advanced use cases. It enables users to anchor data visualisations and labels directly to appliances or environments, such as in smart homes, creating intuitive spatial associations. Its marker-based anchoring system allows for dynamic setups. The extendable framework of ModulARboard makes it easy to add custom data types and visualisations, tailored to specific organisational needs. In large-scale deployments such as factories or industrial environments, group functionality enables entire sets of visualisations to be updated simultaneously, supporting complex monitoring tasks with minimal user effort. Finally, the system's anchoring mechanisms can be used more abstractly to organise and spatially arrange any virtual content, stepping beyond the possibilities of ModulARboard.

Future work

Conducting a user study would offer empirical validation and reveal additional patterns in usability, effectiveness, and user preference. Further exploration of visualisation techniques could also enhance customisability, such as enabling users to request alternative aggregations like averages or medians, or introducing histograms for distribution analysis. Beyond individual customisation, supporting collaborative workflows in modular AR dashboards presents another promising direction.

Conclusion

This thesis explored how Augmented Reality can be leveraged to reimagine traditional data dashboards for Immersive Analytics. In order to do this, it was first important to define what qualifies as AR. This was addressed through an analysis of existing AR definitions, which revealed ambiguities and limitations in older frameworks. A new, more inclusive definition was proposed, removing the requirement for 3D registration and introducing spatial proximity as a criterion. This redefinition enables broader application in immersive analytics contexts.

For the second subquestion regarding the different possibilities of positioning virtual elements in a physical space, a comprehensive design space analysis was conducted. This led to the identification of five anchoring mechanisms used across AR systems. The implementation of ModulARboard demonstrated four of these: marker-based, spatial, view frustum, and observer-relative anchoring, excluding object-relative anchoring. Each mechanism has distinct strengths and weaknesses. Marker-based and object-relative anchoring best suit dynamic setups. However, object-relative anchoring can provide a cleaner implementation without the need for markers. View frustum and observer-relative anchoring can prove useful for keeping an eye on important information. Although it should only be used for a select set of visualisations, since it can obstruct the user's view and cause hazardous situations. Lastly, spatial anchoring is ideal for static setups as opposed to marker-based and object-relative anchoring. However, spatial anchoring requires meshing algorithms in order to orient itself in a physical space. Without it, spatial anchoring does not function across multiple sessions. In general, it is best to be frugal with the number of anchored items, since it can become cluttered and overwhelming when too many items are present in the augmented environment.

The third subquestion focused on personalisation. ModulARboard offers several customisation options: visualisations can be resized and grouped, data elements can be selected to generate adaptive charts, and users can position content using different anchoring strategies. Additionally, the framework supports integration of custom visualisations and new data types, making it highly extensible. These features empower users to construct dashboards that reflect their workflow, environment, and visual preferences. In conclusion, Augmented Reality can be leveraged to reimagine traditional data dashboards through the implementation of anchoring mechanisms and provision of customisability towards the user.

Contents

Chapter 1

Introduction

The origin of Augmented Reality (AR) can be traced back to 1968, when Ivan Sutherland created the first Head-Mounted Display (HMD) [Sut98; Dwy+18]. Since then, the popularity of AR has slowly increased with the development of new technologies that are still in use today. Including the Cave Automatic Virtual Environment (CAVE), Head-Mounted Display and controllers [FP21; Dwy+18]. However, the technology took a giant leap in popularity, going from 1,439 publications in 2013 to 9,113 in 2023. The number of publications with 'Augmented Reality' in the title or abstract increased sixfold over the last ten years, as seen in Figure 1.1. This phenomenon is broader than just the research field since AR has also gained popularity in the industry, education, and entertainment sectors. The current value of the AR market is over 32 billion US dollars and is expected to double in value, reaching a revenue of over 50 billion US dollars by 2028 [How24]. Hence, AR is expected to gain even more popularity in the future.

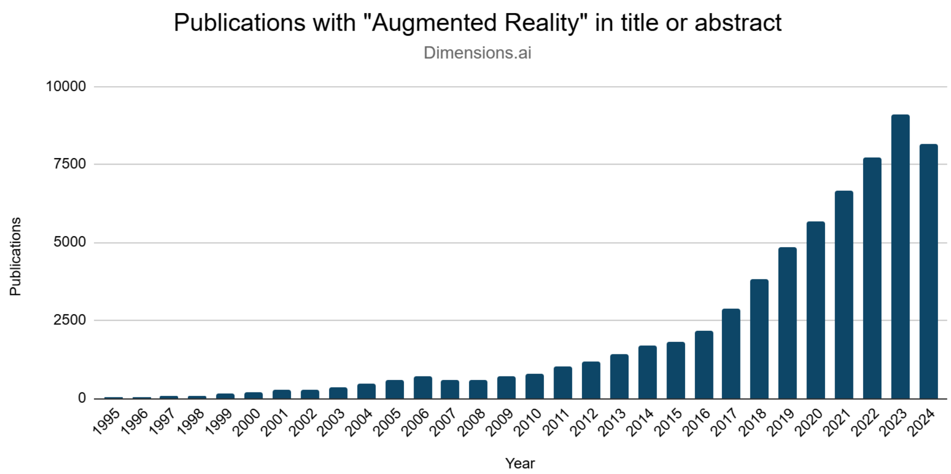


Figure 1.1: Publication trend of 'Augmented Reality' from 1995 till 2024 [Dim].

Augmented Reality extends the real world by superimposing computer-generated information through a computer device with a camera and a display [Cam25; YYJ11; DB11]. This device can be a tablet, head-mounted display (HMD), or a CAVE environment [DB11]. There is often a misconception that AR is limited to visual elements that augment sight. However, it can augment other senses like hearing and touch by implementing surround sound and haptic feedback [Azu97]. Moreover, AR is not limited to solely *adding* virtual elements; it can also *remove* them. Consequently, AR systems can also be used to visually cover up items in order to remove them from the scene. This can also be achieved for other senses, such as hearing. Masking signals can be sent out to cancel certain incoming sounds. This means that noise-cancelling headphones are also Augmented Reality. This opens the door to many possibilities.

Augmented Reality can be used to educate students in various domains in an innovative and interactive way [EDUa; EDUb], give people in training some hands-on experience through real-life simulation [Van], improve task performance on the work floor [Bec+22], provide eye-catching advertisements [Arp], or entertain children innovatively [EDUb; Qui]. However, these examples only scratch the surface of the vast range of possibilities that AR has to offer, since it can also be used to perform data analysis. This combination of immersive technologies and data analysis led to the rise of a new research field, called Immersive Analytics (IA), which dates back to 2015 [Dwy+18]. IA aims to make data analysis more accessible to the average person by making the data and the data analysis tools more accessible. Moreover, it builds on the fields of data visualisation, visual analytics, virtual reality, computer graphics and human-computer interaction. Yet, it is not tied to the implementation of specific techniques. As a result, Augmented Reality can also be utilised for data analysis in the field of Immersive Analytics.

As IA is a relatively young research field, it still has an extensive research agenda and many questions left to answer. Like when (if ever) the use of immersive technologies offers benefits over current desktop visual analytics tools? Or what guidelines are there in order to employ situated analytics in any process or situation? Is it possible that the usage of emotionally engaging and immersive data-driven narratives could lead to less objective and worse decisions by stakeholders? These and more questions form the research agenda of Immersive Analytics [Mar+18a]. Because there is still so much left to explore and study, we chose to conduct research in this field in order to contribute to the research agenda of Immersive Analytics. To contribute to this evolving field, this thesis focuses on one promising avenue: Augmented Reality dashboards. As a result, this thesis will analyse how Augmented Reality can be leveraged in order to reimagine traditional data dashboards.

Before this can be researched, it is important to first understand what Augmented Reality exactly means and stands for. Are existing definitions of AR still applicable, or do they fall short of describing modern systems? Or do they fall short? Is it necessary to compose a new definition? All these questions have to be answered before research can be conducted concerning AR. Otherwise, misconceptions can occur regarding the implemented techniques. For example, 2D overlays are often not considered to be AR [Azu97]. As a result, if solely 2D overlays were implemented, the application would not be AR and would not contribute to the field of Immersive Analytics. Hence, this will form the first part of the research question.

Once it is clear what constitutes AR, research can be conducted regarding already existing AR systems. How are they implemented? Are there similarities and/or differences between the various systems? What is it about them that makes them AR? And most importantly, how do they position the virtual elements in the physical space in order to augment the environment? This extensive research allows for the composition of a design space, which in turn allows the scope to be determined. This allows us to define the focus areas for the prototype application developed to explore how AR can be leveraged to reimagine traditional data dashboards. However, rather than directly replicating a traditional dashboard in AR, this thesis will aim to provide a more innovative AR dashboard. In order to do this, this thesis will explore how customisability can be introduced when reimagining traditional data dashboards.

In summary, this thesis focuses on the following research question and subquestions:
How can Augmented Reality be leveraged to reimagine traditional data dashboards for Immersive Analytics?

1. What qualifies as Augmented Reality?
2. What techniques exist/can be applied to position visual elements in AR?
3. How can customisability be introduced when reimagining traditional dashboards?

Chapter 2

Related work

This section discusses related works regarding Augmented Reality (AR) and Immersive Analytics (IA). First, it delves deeper into the specifics of Immersive Analytics and its applications. Secondly, the focus is on Augmented Reality. Throughout the years, there have been many works on AR aiming to find the true definition. This led to the rise of various definitions of what should be considered as Augmented Reality, resulting in incertitude and numerous grey zones. Hence, the second part of the related work focuses on the definition of AR and what qualifies as AR, answering the first part of the research question.

2.1 Immersive Analytics (IA)

Immersive Analytics is a relatively new research field that dates back to when it was first introduced at IEEE BDVA 2015 [Dwy+18]. The name comes from researchers who were experimenting with the combination of data visualisation and virtual/mixed reality. Data visualisation is traditionally two-dimensional, mostly because of the disapproval of 3D visualisations by researchers in the information visualisation community. However, the development of immersive technologies, like Virtual Reality (VR) and Augmented Reality (AR), has progressed a lot over the years. This makes way for novel and innovative ways in which people use their computers, including how they perform data analytics and decision-making. Immersive Analytics aims to make data understanding more accessible for everyone (individually or collaboratively) by removing barriers between them and their data, along with their analysis tools. This is because the amount and the complexity of the available data keep increasing, even surpassing our ability to understand or utilise it in decision-making. More specifically, "Immersive Analytics is the use of engaging, embodied analysis tools to support data understanding and decision making" [Dwy+18]. It builds upon various principles of data visualisation, visual analytics, virtual reality, computer graphics and human-computer interaction (Figure 2.1). However, Immersive Analytics is not tied to the use of specific techniques.

Because the hardware required for immersive data visualisation and exploration already exists, the subject of Immersive Analytics is to provide the knowledge needed to design appropriate human-computer interfaces and data visualisations, utilising these technologies. It offers several opportunities beyond the more traditional visual analytics, like situated analytics, embodied data exploration, collaboration, spatial immersion, multi-sensor presentation and engagement (Figure 2.1). However, this thesis will focus more on situated analytics and spatial immersion. For a more extensive explanation of the broad topic, we refer to the book of Immersive Analytics [Mar+18a].

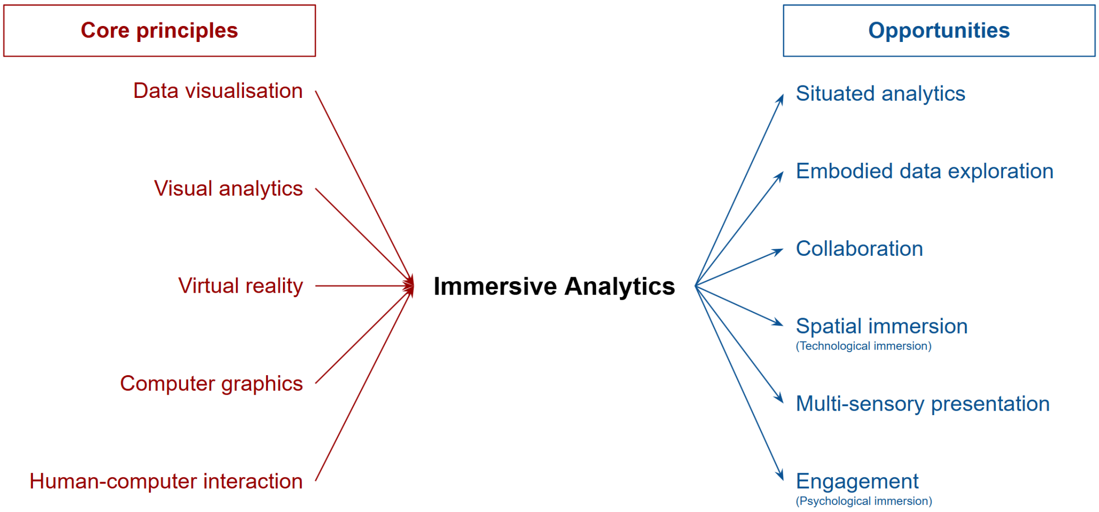


Figure 2.1: Illustration of the core principles of Immersive Analytics and the opportunities it has to offer [Dwy+18].

2.2 Augmented Reality (AR)

This section focuses on answering the first part of the research question: "What qualifies as Augmented Reality?". However, before attempting to define what qualifies as AR, it is inevitable to first understand what Augmented Reality is. To tackle this, the first subsection focuses on the Reality-Virtuality Continuum of Milgram [MK94] to better understand the AR setting. The second subsection focuses on some of the most well-known definitions of augmented reality and what they consider AR. The third subsection covers the shortcomings of the current definitions and associated grey areas to conclude what qualifies as Augmented Reality in this thesis in the last section.

2.2.1 The Reality-Virtuality Continuum

Milgram and Kishino [MK94] performed a taxonomy on Mixed Reality (MR) displays and depicted the Virtuality Continuum (VC) or Reality-Virtuality (RV) Continuum [Mil+95], as seen in Figure 2.2. The real-world aspect is on the continuum's left-hand side. This side represents the real world with physical objects, which can be perceived through any medium (e.g. through transparent or video displays). The other extreme is on the other end of the continuum, encompassing the entirely virtual world consisting of sole virtual artefacts. None of the perceived elements physically exist in the real world. So, in conclusion, Mixed Reality environments are everything that lies between both extremes of the RV Continuum.

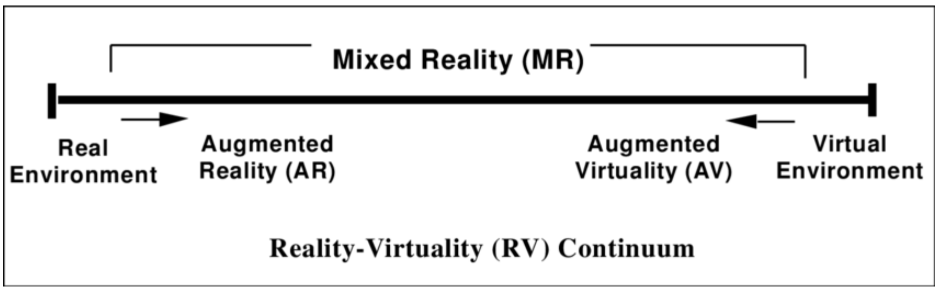


Figure 2.2: Reality-Virtuality Continuum (Figure from Milgram et al. [Mil+95]).

While MR covers everything between fully real- and virtual environments, it is possible to make an extra distinction between Augmented Reality (AR) and Augmented Virtuality (AV), as seen in Figure 2.2. As described by Milgram and Kishino [MK94], Augmented Reality is a term "to refer to any case in which an otherwise real environment is 'augmented' by means of virtual (computer graphic) objects". Augmented Virtuality, on the other hand, is the opposite of AR, where a virtual environment gets augmented with real objects. It can, for example, show the user's hands inside an entirely virtual environment. This way, the user can interact with the environment. Augmented Virtuality is, however, less common than Augmented Reality, with the term being exhibited in only 5,868 publications on Dimensions.ai[1] compared to 383,960 publications for AR.

2.2.2 Analysis of AR definitions

While the aforementioned definition of AR from Milgram and Kishino [MK94] is sufficient to form a basic understanding of AR, it does not provide explicit inclusion or exclusion criteria for what can be considered AR and what cannot. It merely states that the real world gets augmented with virtual artefacts, which is too broad a description to use for classification purposes. Following this definition, a movie that captures and augments the real world with Computer-Generated Imagery (CGI) is also classified as Augmented Reality. This is because Milgram and Kishino [MK94] do not specify the amount of time allowed between capturing the real world and perceiving it again. Hence, a more thorough description of what constitutes AR is required.

Azuma [Azu97] defines AR as "systems that have the following three characteristics: 1) Combines real and virtual, 2) Interactive in real-time, 3) Registered in 3-D". This definition ensures that movies augmented with CGI are not included as AR because they do not comply with the second characteristic, which states that they should be interactive in real-time. However, the definition does not specify the allowed geographical distance between the entity capturing the real world and the observer. The video could be captured by a drone and perceived by someone a few kilometres away. As long as the footage is augmented with virtual elements and viewed in real-time, it is considered AR. Azuma [Azu97] proceeds to claim that Augmented Reality is not limited to solely *adding* objects to a real environment, as it also has the potential to *remove* them. He states that "graphic overlays might also be used to remove or hide parts of the real environment from a user". This could, for example, be used to clear a city of its buildings and overlay it with a view of what the city used to look like in a specific year. Azuma [Azu97] additionally notes that AR is not limited to augmenting sight because it might apply to all senses, like touch, smell, hearing and taste. This characteristic of AR is often forgotten since most implementations are focused on visual elements.

The definition from Azuma [Azu97] is more specific than the one from Milgram and Kishino [MK94] and is generally perceived as the *true* definition of AR. However, it still has some shortcomings. It does not explicitly state that it encompasses all senses. This is mentioned in the paper but not directly in the definition (which consists of the three characteristics). It prioritises sight when examining the third requirement, which states that AR should be registered in 3D, implying it must occupy a position within the actual 3D environment. This raises concerns about how other senses, such as touch, smell, hearing, or taste, would fit within this framework. It is unclear how these senses would be registered in 3D and whether this is necessary.

Furthermore, the definition does not clarify what is meant by 'real' and 'virtual', leaving room for interpretation. However, the explanation of Milgram and Kishino [MK94] can be used for this clarification, as Azuma [Azu97] refers to their Reality-Virtuality continuum. These issues render the definition narrowly focused as it puts many techniques in a grey area, as illustrated in the following examples.

[1] https://www.dimensions.ai/

2.2.3 Examples and ambiguities

According to the definition, it is not AR when a real person who is not physically with the user is augmented into their environment. This example complies with the second and third characteristics of the definition since it is possible to interact with the other person in real-time, and they are projected somewhere in their environment (3D). However, it does not yield to the first characteristic. According to the definition of Milgram and Kishino [MK94], something is 'real' when "it can either be observed directly or it can be sampled and then resynthesised via some display device". This means that the other person projected into the environment is considered 'real' since they would be video-captured and resynthesised in the user's environment. On top of that, there is no limiting factor on the allowed distance between the sampling and resynthesisation process. This results in the absence of a virtual factor and, thus, not adhering to the first characteristic. It would, however, be considered AR when the person's sample is augmented by, for example, adding virtual lighting or changing their clothes.

Head-Up Displays (HUDs) do not classify as Augmented Reality either according to the definition. They display virtual objects and are aligned somewhere in the real world, thus adhering to the first and third requirements. Yet, HUDs meet only half of the second requirement, depending on how it is interpreted. If the 'interactive' part of the characteristic is interpreted as 'interactive with the user', then HUDs do not classify as AR since they only showcase information. However, suppose it is considered 'interactive in any way', for example, with the environment. In that case, HUDs are considered AR since they display real-time information from the immediate vicinity and update accordingly. This means that HUDs displaying only static information are not classified as AR.

2D overlays are another example of what is technically not considered Augmented Reality because it does not conform to the third characteristic. Azuma [Azu97] explicitly mentions that "2-D virtual overlays on top of live video can be done at interactive rates, but the overlays are not combined with the real world in 3-D". This means that when a 2D overlay showcases extra information, even when it is interactive, it is not considered AR because the overlay does not have a location in the 3D world. However, the definition could be interpreted slightly differently again to be able to consider it as AR nevertheless. People can perceive a particular area of their environment. This field of view can be compared to the bounding box of a virtual camera, which can capture a part of a 3D scene (e.g. Unity editor). This bounding box is outlined in red in Figure 2.3 and is called the 'view frustum' in 3D computer graphics [Tec]. While the 2D overlay is not registered in 3D in the real environment, it is, however, anchored in 3D to the view frustum relative to the camera's position or the user's head (Figure 2.4). No matter how the camera or head is turned, the position of the 2D overlay will remain the same. In conclusion, depending on how the third characteristic is interpreted, 2D overlays can be classified as AR.

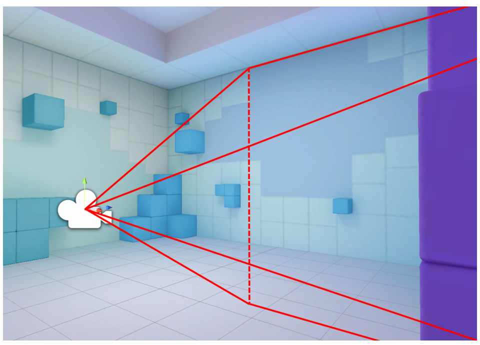


Figure 2.3: View frustum visualised in red as the volume a camera can capture in a 3D environment, illustrated with Unity.

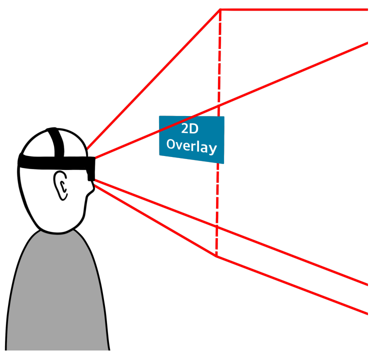


Figure 2.4: View frustum visualised in red for a Head-Mounted Display (HMD) with a 2D overlay in blue anchored to the view frustum.

2.2.4 What qualifies as Augmented Reality?

The question now remains: What qualifies as Augmented Reality? The definition of Azuma primarily focuses on strict criteria like 3D registration and real-time interaction, leaving many practical applications in a grey zone. However, Azuma [Azu97] also describes AR in the following way:

> Why is Augmented Reality an interesting topic? Why is combining real and virtual objects in 3-D useful? Augmented Reality enhances a user's perception of and interaction with the real world. The virtual objects display information that the user cannot directly detect with their own senses. The information conveyed by the virtual objects helps a user perform real-world tasks. [Azu97]

This perspective emphasises the primary goal of AR, which is to enhance perception and interaction through virtual artefacts that integrate with the real world. From this perspective, the systems described in the previous sections fit perfectly within Azuma's description. So, to answer this section's question, this thesis will consider an application to be AR when there is a combination of the real world and virtual artefacts (not limited to visual elements). It is no longer required that the virtual elements have to be registered in 3D, since this criterion is difficult to extend to the other senses. However, a new criterion will be added that states that the sampling and resynthesisation process have to happen in approximately the same location in order for the sampled subject to be considered 'real'. More specifically, to clarify when it is no longer considered approximately the same location, the requirement can be illustrated by placing people at both locations. When someone is present where the sampling happens and someone at the resynthesisation process, the question is whether they can perceive one another through any of their senses. If they can see, smell, touch or taste each other or can hear one another yell, they are considered to be in approximately the same location. Because even without augmentation, they would still be able to interact with each other in real-time. Consequently, when they cannot perceive one another from the sampling or resynthesisation location, it is now considered 'virtual'. Because augmentation is required in order to be able to interact with each other in real-time. This can be extended for interaction between people and inanimate objects as well.

In summary, the new definition utilises the description of Milgram and Kishino [MK94] to form the distinction between 'real' and 'virtual' with the extra requirement that the sampling and resynthesisation have to happen in the same approximate location in order to be 'real'. Furthermore, it adopts two of Azuma's [Azu97] characteristics, (1) combination of real and virtual and (2) interactive in real-time, in order to define what AR is. In conclusion, this new definition now includes the grey-zone applications mentioned in the previous section. This inclusive interpretation aligns with Azuma's vision and extends Milgram's by including the other senses as well. This allows for a richer exploration of Immersive Analytics without the constraints of strict categorisation.

Chapter 3

Related work: Design Space

This chapter will construct the design space. This is done by analysing and comparing many prior AR literature studies. These studies enable the deduction of categories, indicating similarities across various studies, while the subcategories indicate slight differentiations in adaptation of these trends. This resulted in seven categories including the application area, the augmented senses, the display technique, the extent of presence, whether collaboration is supported, the type of system and the anchoring mechanisms. Then, fourteen studies were selected to be evaluated against the categories of this design space in order to find the most popular implementations. Then, all findings and conclusions about the studied sources are summarised. Lastly, this entire research process is explained in the methodology section.

3.1 Application area

This first category covers the various purposes for which AR applications can be developed. It is possible to go into grand detail, as many researchers try to encompass every use case in a distinct category. This results in categories including AR applications for manufacturing, maintenance, healthcare, gaming, advertising, e-commerce, education, data analysis and information visualisation. However, many applications can have a similar foundation while deployed in different industries. This means that the implementation of AR systems is more closely tied to the goal they are designed for than the industry in which they are situated. Similar systems will likely be put in distinct categories when defining many detailed categories, while they belong together. This indicates that the categories should be broader depending on the required level of detail. In this case, employing the higher-level categories deducted by Ludwig and Reimann [LR05] as subcategories is sufficient. They define three broad categories for the application areas of AR systems. First, they define 'Presentation & Visualisation', which includes sales, marketing, geographic visualisation, navigation, architecture and interior design applications. The second category is 'Industry', encompassing AR training, education, maintenance/repair, remote support, product development and production planning. Lastly, the third category is called 'Edutainment' and contains systems for museum guides, tourism, and gaming.

3.2 Senses

Augmented Reality is often used to augment the physical world by embedding virtual visual elements. However, as is often forgotten, AR is not limited to sight only, but can augment the other senses (hearing, touch, smell and taste) as well, as explained by Azuma [Azu97]. As discussed in section 2, the definitions of Milgram and Kishino [MK94] and Azuma [Azu97] fall short because they don't include the other senses and have a limited definition of what is considered 'real' and 'virtual'. Because of this, section 2.2.4 defined a new combined definition in order to be more inclusive. This means that a colleague from another building who gets augmented in your meeting room in real-time is now considered AR, since the sampling happens in another location from the resynthesis.

This new definition of what is 'real' and 'virtual' is clear for visual elements meant to augment sight. However, the question remains how the other senses can be augmented to comply with this definition. First of all, sound is considered 'real' when it is captured via a microphone and replayed in the same location and in real-time. Once it is replayed at a later time or in another location, it is considered virtual. Similarly to when the sound is completely fabricated and simulated. For example, a nature exhibition augmented with recorded sounds of birds and crickets is considered AR since the sounds are played at a later time and another location. Secondly, touch is 'real' when you touch something that is physically there. A touch is virtual when the sensation of touch is simulated with haptic feedback, like vibrations or air pressure. For example, the controller vibrates when the user clicks on something. Thirdly, a smell is 'real' when it is naturally there without any intentional interference, like the smell of a restaurant or a flower field. A smell is 'virtual' when it is intentionally put there to provide augmentation to the experience. For example, when going to a 4D cinema that uses different fragrances to simulate different environments throughout the movie. Lastly, taste is difficult to define. However, taste could be considered 'real' when it can be tasted directly. While it can be considered 'virtual' when the taste is fabricated through a combination of scent and visuals. For example, like the claim that all Skittles actually taste the same but use different scents and food colourings to mimic different flavours [Rus18].

In conclusion, the requirement that the sampling has to happen in the same location as the resynthesis mostly applies to sight and hearing, since these are the only two senses that can be sampled and resynthesised in another location. The other senses cannot be digitally sampled, meaning they have to be simulated by physically fabricating something similar.

Moreover, as mentioned in section 2.2.2, Azuma [Azu97] mentions that Augmented Reality is not limited to solely *adding* objects, but can also be used to *remove* them. In general, this is achieved through cancelling out the elements in the physical space through virtual elements, depending on the sense. For sight, this can be done through overlaying the real world with an image that covers the item to be removed. For hearing, this is achieved through playing masking signals against the incoming sounds to cancel them out. As a result, noise-cancelling headphones are also considered Augmented Reality [Azu97]. For smell and taste, things can be 'removed' by trying to cancel them out. For smell, this can be attempted through releasing a complementary smell. For example, a good sweet smell to cover up a bad smell. Or in the case of the sense of taste, a very bitter flavour can be removed by adding a sweet flavour to cancel each other out. Touch, on the other hand, is difficult to cancel out, since this would simply result in the removal of the physical object. It could also be attempted by overlaying it with another texture.

3.3 Display technique

This category describes the medium through which we perceive the generated artefacts superimposed onto the real world. We refer to this concept as a 'display technique', as done by Azuma et al. [Azu+01] and Zhou, Duh, and Billinghurst [ZDB08], but it has also been called a 'viewing option' by Yuen, Yaoyuneyong, and Johnson [YYJ11]. This literature study found that most AR systems are handheld, head-mounted or projection-based, as confirmed by Azuma et al. [Azu+01] and Zhou, Duh, and Billinghurst [ZDB08]. Some systems use stationary displays like laptops or desktops and a separate camera to form an AR setup. However, since these setups are not prevalent, this thesis will focus on the three categories devised by Azuma et al. [Azu+01].

3.3.1 Handheld display

This type of display consists of a screen that users can hold in their hands, accompanied by a camera to capture the real world. This way, the device acts as a window through which the user can perceive the real world augmented with virtual artefacts [Azu+01]. Various AR devices can be handheld displays, like tablets and smartphones [ZDB08].

3.3.2 Head-Mounted Display (HMD)

Both optical see-through (OST) and video see-through (VST) displays exist. With optical see-through displays, the user can see the real world through the transparent display as they would through regular glasses. The real world is then augmented by overlaying virtual artefacts on the transparent display [Azu+01], as seen in Figure 3.1 on the left. This technology is safer since users can still see the world around them during a power outage [Yad22]. With video-see-through, a camera captures the real world from near the position of the user's eyes and then overlays the virtual artefacts on top of the captured video. The combined content is then shown on the opaque display in front of the user's eyes [Azu+01], as seen in Figure 3.1 on the right. The distinction between OST and VST is usually made in the context of HMDs, but could also be extended to other types of displays.

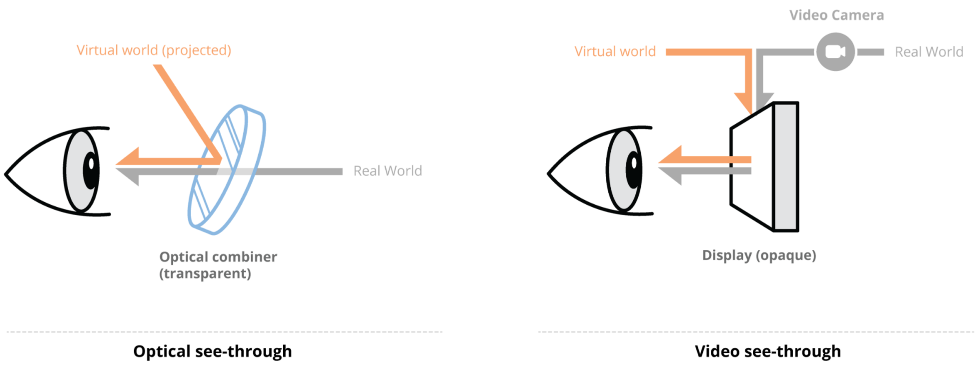


Figure 3.1: Simplified visual representation of the technological working of an optical see-through (left) and video see-through display (right) (Figure from Yadav [Yad22]).

Head-Mounted Displays (HMDs), or Head-Worn Displays (HWDs), describe the technique of mounting a display on top of the user's head with the display in front of their eyes [Azu+01]. This way, virtual artefacts can be superimposed onto the virtual world the user perceives through the HMD [ZDB08]. As mentioned above, optical see-through (OST) and video see-through (VST) HMDs exist. The Microsoft HoloLens and Magic Leap are examples of OST HMDs [DGW22], while VR headsets like the Oculus Quest can provide VST AR besides their mainly intended VR usage. This category could also be expanded to 'wearable display' to include upcoming technologies like wearable glasses, retinal projection, and other non-head-mounted body-worn displays.

3.3.3 Projection-based

Projection-based displays encompass every type of display where the real world is augmented through projections of virtual elements. They can be placed on various surfaces, like walls, floors, maquettes, people or everyday objects. Projections can even be combined to cover an entire room, creating a fully immersive environment called CAVE (CAVE Automatic Virtual Environment). However, projections are not limited to opaque displays only. Virtual elements can also be projected on transparent surfaces or even particles in the air, resulting in holograms.

3.4 Extent of presence

This category wants to capture how immersive a certain AR system is. The question is how the level of immersion can be described, since it is a rather subjective experience. Dwyer et al. [Dwy+18] distinguish between two types of immersion, technological and psychological immersion. Psychological immersion denotes the “cognitive state experienced by a user when they are absorbed by some task”, which is a highly subjective measure, making it difficult to use as a classification criterion. Technological immersion, contrarily, describes the technologies used to immerse a user, which makes it a more objective measure. It can be further subdivided into presence, which refers to the subjective psychological experience of feeling located in a virtual or remote space, and immersion, which describes the objective characteristics of the technology used to present the virtual environment. However, the immersion aspect of a system’s technological immersion relates closely to the display technique used, which makes it ambiguous to use as a classification criterion again. For this reason, this work will focus on the aspect of presence to classify the level of immersion.

While presence remains a subjective experience, it aligns with the concept of the ‘Extent of Presence Metaphor’ of Milgram and Kishino [MK94] as illustrated in Figure 3.2. It defines a spectrum ranging from exocentric (on the left), where the user adopts an observer-like perspective, to egocentric (on the right), where the user is embedded within the augmented environment. Although presence is typically considered a continuous scale, for classification purposes, the scale can be divided into two broad categories: exocentric and egocentric. AR systems that present virtual content from an external viewpoint are exocentric, whereas systems that enhance the user’s sense of being embedded in the augmented space are egocentric. In other words, applications utilising a third-person view are exocentric, while systems implementing a first-person view are egocentric, as illustrated in Figure 3.3 [AB20].

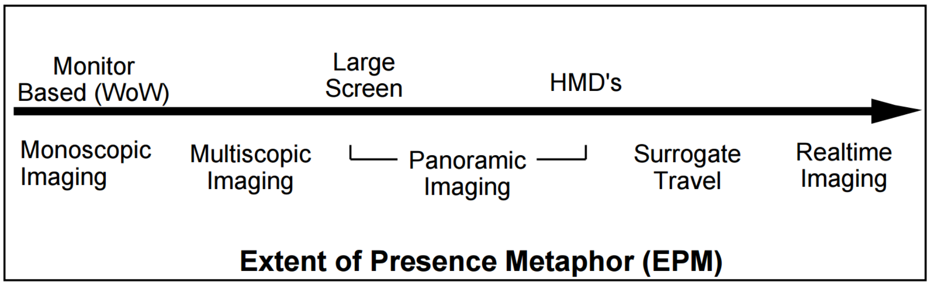


Figure 3.2: Extent of Presence Metaphor (EPM) (Figure from Milgram et al. [Mil+95] without classes).

Although these descriptions are straightforward, categorising AR systems according to the description is not. Most AR systems fall into three categories: handheld, head-mounted or projection-based, as discussed in section 3.3. When an application uses a head-mounted or handheld display, it can be argued that the view is automatically first-person and therefore egocentric. Moreover, when an AR system is projection-based, it would be considered to have a third-person view and therefore be exocentric. However, this results in similar categories as the display techniques discussed in section 3.3 and most AR systems being categorised as egocentric.

On the other hand, some research papers consider an AR system to be egocentric when the visualisations are built around the user, meaning that the viewpoint of the user is at the centre. An AR system is then considered to be exocentric when the user is outside the visualisation and can look onto it, as can be seen in Figure 3.4 [Mar+18b; Yan+18]. This alternative interpretation is often used in research concerning AR systems where they want to compare multiple visualisations using the same setup, like for example a HMD. It is also possible for an AR system to be both egocentric and exocentric when applying both techniques. It is, for example, possible to have a visualisation anchored to the space so that the user is able to

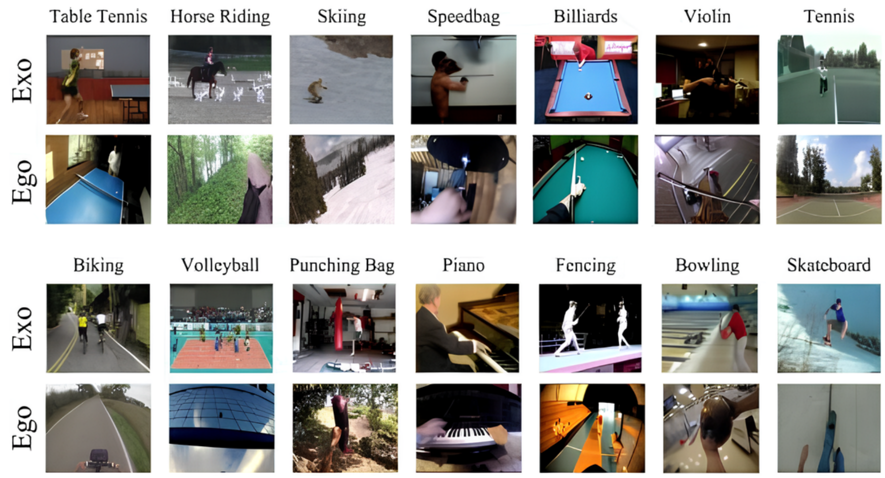


Figure 3.3: Egocentric vs exocentric views for different applications (Figure from Almasi, and Boza [AB20]).

walk around it, while having a menu anchored to the field of view, following the user's head movement. In this case, the visualisation anchored to space makes the application exocentric, while the menu following the head movements makes the application egocentric. In conclusion, the application is both exocentric and egocentric.

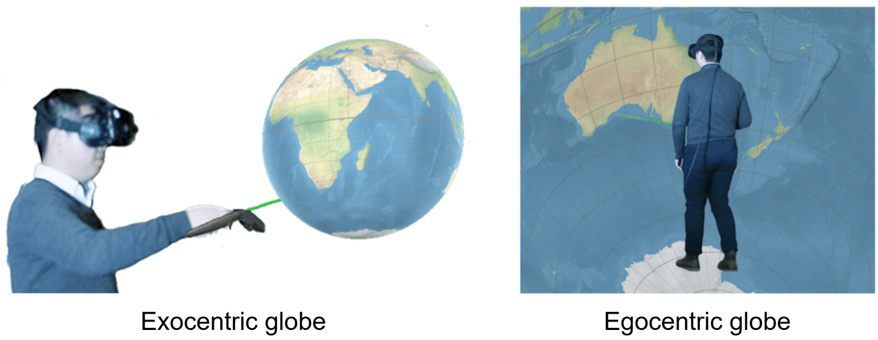


Figure 3.4: Exocentric view (left) vs egocentric view (right) (Figure from Yang et al. [Yan+18]).

Since this interpretation fits better and results in more evenly distributed categories, this thesis will adopt the same technique. However, this description of exocentric compared to egocentric is still not completely straightforward to use for categorisation purposes. Therefore, the related works are divided into the two categories at best effort, keeping this description in mind.

3.5 Collaboration

Fonnet and Prié [FP21] found various aspects of collaboration, including the same or separate physical place, synchronous or asynchronous, same or different visualisation, and modes of communication. Firstly, the distinction between the **same or separate physical place** is straightforward, meaning that people are located in the same physical space and can work together directly or separately, meaning collaboration happens indirectly.

Secondly, **synchronously** suggests that people work together simultaneously, while **asynchronously** means at different times. In the latter case, an example could be that someone makes annotations for someone else to review later. Thirdly, when multiple users work on the same project, it does not mean they must always perceive the exact visualisation. There could be a shared view, which is the same for every user, while providing a personal view. Users could then alter their personal view without affecting the shared view. This could allow for individual inspection, primarily before sharing one's findings. However, there are several possibilities for combining shared and individual views, resulting in **identical or distinct visualisations**.

Lastly, there are different **ways to communicate** while collaborating, like head-tracking a specific user to follow their eye gaze/attention or using a ray cast for finger-pointing. However, not every form of communication is equally fit for every situation. It is, for example, possible to work together in the same physical space using see-through HMDs. In this case, users can see each other through the displays and perceive gestures for communication. However, remote collaboration is another example where it is impossible to see collaborators directly. In this case, an avatar representation of the user is needed or another way of seeing what the user is pointing at. [FP21]

While collaboration can be integrated into a system in various ways, the scope of the categories is limited to whether the AR application supports it. The system is then classified as 'Collaboration' if supported; otherwise, it is classified as 'Solo'. A system can also be categorised as both if it is possible to use alone and supports collaboration.

3.6 System

This category describes a system's composition, whether it is a combined system or a stand-alone system. A system is viewed as stand-alone when no single component can be taken away because it would otherwise no longer be able to function as an AR system. For example, a system consisting of a display and a separate camera needs both components to make AR function, which makes it a stand-alone system. On the other hand, a combined system consists of a barebone AR system with additional elements for extra functionality. This can, for example, be a system combining an HMD with a desktop and an additional database server. The HMD can function as a barebone AR system, while the desktop and database are additional components.

3.7 Anchoring mechanism

This category describes how virtual artefacts are augmented and combined with the real world. Virtual elements are anchored to the actual environment, which can be achieved through various techniques. This can range from marker tracking to plane detection and meshing algorithms. The different techniques are self-determined by categorising prior literature into similar groups according to the coordinate system used for anchoring virtual elements to the real world. In general, the anchoring mechanisms define the coordinate system used for relative anchoring and the needed techniques. They are discussed in the following subsections.

3.7.1 Marker-based anchoring

Marker-based anchoring binds virtual objects to fiducial markers. These "are 2D visual patterns used in AR to provide a reference point in the real world to anchor digital content" [UNT]. The marker can be a QR code, an ArUco marker or even an image. It is defined beforehand, so it is known by the AR system. When the marker is in the camera's view frustum and is recognised by the system, the virtual object is augmented on top. When the marker is moved, the virtual element moves along with the marker, since the coordinate system is defined by the 2D marker (Figure 3.5).

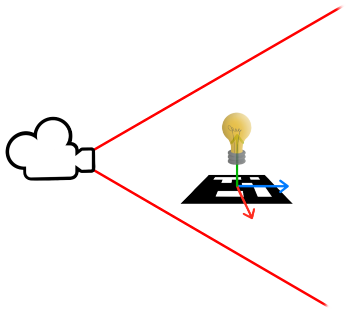


Figure 3.5: Marker-based anchoring illustrated with a virtual lightbulb augmented on top of an ArUco marker. The coordinate system for positioning virtual elements is defined by the 2D marker.

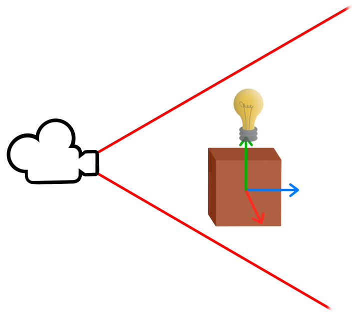


Figure 3.6: Object-relative anchoring illustrated with a virtual lightbulb augmented on top of a 3D tangible object. The coordinate system for positioning virtual elements is defined by the 3D object.

3.7.2 Object-relative anchoring

Object-relative anchoring positions the virtual artefact relative to a 3D tangible object. The virtual object moves along with the pose (i.e. location and orientation) of the tangible object since the object's pose defines the coordinate system (Figure 3.6). This method differs from marker-based tracking because it dynamically tracks a 3D object without needing a marker. This allows for cleaner anchoring, since it is not necessary to put up markers around the physical space. However, this requires more complex algorithms since the objects can be harder to recognise than fiducial markers.

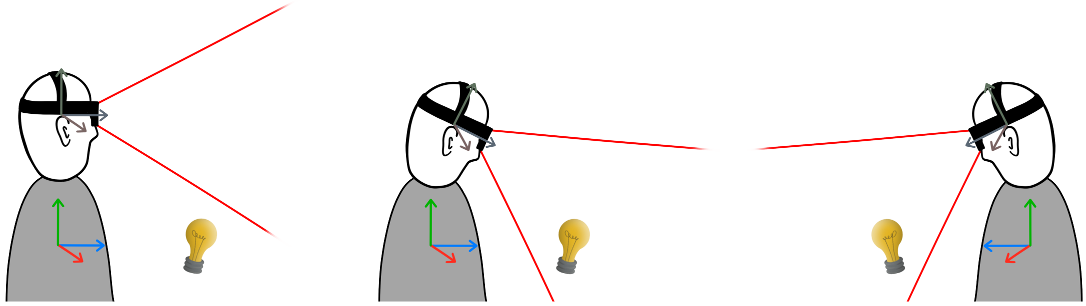
3.7.3 Observer-relative anchoring

This technique anchors virtual artefacts relative to the observer's pose (i.e., location and orientation). The observer is the entity in the AR systems that perceives the environment. This can be a camera (of a tablet) or a user with an HMD. However, this technique is easier to understand when the observer is a user wearing an HMD since this creates a clear distinction between the observer's pose and the view frustum's pose. The observer (i.e. the user) can move around freely throughout the environment, while the view frustum (i.e. the head or HMD camera) is bound to the observer's location. The view frustum can only move slightly from that position, mainly by rotating around, e.g. by turning their head. The virtual elements are relative to the observer's pose, meaning that the virtual object will remain in the same location when looking forward (Figure 3.7a), looking down (Figure 3.7b) and even when turning/walking around (Figure 3.7c). No matter how the view frustum is oriented or where the observer is positioned, the virtual artefacts will remain in the position relative to the observer. This type of anchoring can, for example, be used for a compass that is continuously positioned in front of the user at hip level and can be consulted by looking down, as implemented by Becher et al. [Bec+22].

The concept remains the same when the view frustum's pose is more tightly coupled to the observer's pose. For example, in the case of a tablet or single camera, the observer's pose can be imagined as a transparent box encapsulating the camera (Figure 3.8). This box is equivalent to the user wearing an HMD, as it can be moved around freely. The view frustum, on the other hand, is defined by the orientation of the camera, equivalent to the user moving their head around.

3.7.4 View frustum anchoring

As already mentioned in Section 2.2, the virtual elements can also be anchored to the view frustum of the observer. In this case, the virtual objects are always in the exact location relative



(a) The virtual lightbulb is located in front of the observer, but the view frustum is not oriented to perceive it.

(b) The observer can see the virtual lightbulb as opposed to scenario **(a)** because the view frustum is now oriented downward to perceive it.

(c) The observer can still perceive the virtual lightbulb when turning around and looking down because it is positioned in front of the observer.

Figure 3.7: Observer-relative anchoring illustrated with a virtual lightbulb augmented relative to the observer's pose (i.e. location and orientation). The three-coloured axes on the user's body represent the coordinate system illustrating the observer's pose. The greyed-out coordinate system on the user's head represents the view frustum.

(a) The coordinate system of the observer and the view frustum are parallel to each other.

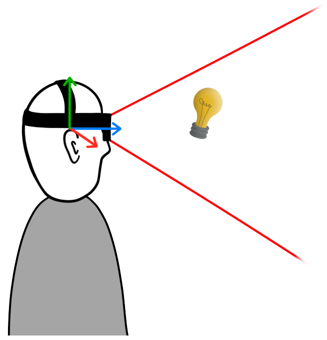
(b) The view frustum is oriented upward in comparison to the pose of the observer.

Figure 3.8: The three-coloured axes represent the coordinate system illustrating the observer's pose. The greyed-out coordinate system represents the view frustum.

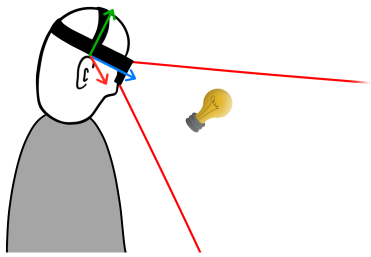
to how the observer is oriented (Figure 3.9). As opposed to observer-relative anchoring, this technique does not differentiate between the observer's pose and the view frustum's orientation. They are viewed as one entity. This technique makes it possible to continuously have real-time information overlayed at the corner of the screen. It is also important to note that when the observer is a person with an HMD, the virtual artefacts are not anchored to the view frustum of the user's eyes but rather to the view frustum of the camera on the HMD. However, with eye-tracking technology, it is possible to anchor elements to the view frustum of the user's eyes.

3.7.5 Spatial anchoring

This technique anchors virtual objects relative to the 3D space in which the observer is located. This can be achieved through plane detection and meshing algorithms. Spatial anchoring can be used to hang virtual posters on a wall, place computer-generated furniture in a room, and position a projection of a colleague in a meeting room, amongst other things. For example, in Figure 3.10, a virtual lamp is located in the corner of a room. The lamp will remain in the same corner when the observer moves around, since the coordinate system is relative to the environment.



(a) The virtual lightbulb is located in front of the observer relative to the view frustum.



(b) When the observer looks downward, the coordinate system (view frustum) tilts downward, resulting in the virtual lightbulb tilting along. The lightbulb remains in the same location on the display.

Figure 3.9: View frustum anchoring illustrated with a user wearing an HMD. The coordinate system depicts the pose (i.e. location and orientation) of the observer's view frustum.

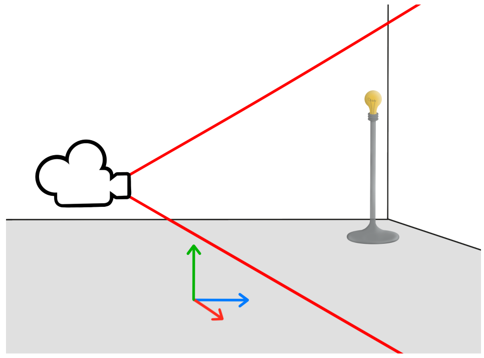


Figure 3.10: Spatial anchoring illustrated with a virtual lamp located in the corner of a room (3D space). The coordinate system is defined by the environment. When the observer moves around, the lamp will remain in the same corner.

3.8 Conclusions about AR

This thesis performed a very extensive literature study concerning AR research and systems. In general, it was found that most studies focus on either custom interaction techniques or devices that can be used to interact with an augmented/virtual environment [Sid+21; Cor+20; Zha+19; Ges+20; Jam+20], specialised systems designed for a specific use case [RMS13; NM19; Bec+22; Zhu+15; CCV20; Dün+12; May+22] or improving comprehension of three dimensional data by visualising it in augmented/virtual reality [But+18; Wan+20; Rei+22]. In some cases, AR systems are preferred over traditional desktops, since traditional monitors have a limited amount of space to work with, while AR can utilise the entire physical spaceWang et al. [Wan+20]. While in other cases, AR is implemented to experiment with collaboration opportunities [But+18; Jam+20]. However, the literature study did not really encounter Augmented Reality dashboards. Because of this, it is interesting to go in a new direction with this thesis in order to reimagine traditional data dashboards with Augmented Reality. To narrow the scope down, this thesis will produce an AR application that focuses on the augmentation of sight through a stand-alone HMD with an egocentric view. It will be intended for individual usage in the application area of presentation & visualisation. Moreover, it will attempt to test out all the anchoring mechanisms discussed in section 3.7.

Furthermore, fourteen papers delivering an AR system were selected to be evaluated against the seven categories of the design space. This resulted in the table as shown in Figure 3.11. From this table, it can be concluded that most AR studies still focus on sight instead of the other senses, since none of the studies augmented any of the other senses. Moreover, most systems use an HMD as a display technique, while it is seldom used as a stand-alone system. Eleven studies used an HMD as a display technique, and nine of them featured a combined system. Furthermore, it can be deduced that most AR systems focus on individual usage and implement an exocentric viewpoint. Lastly, in these fourteen studies, marker-based and spatial anchoring were the most popular anchoring techniques. Other than the findings about the categories of the design space, the selected papers also provided other conclusions regarding technical limitations of AR, interaction with AR and some general findings about AR.

Becher et al. [Bec+22] found in their research that wearing an HMD can become tiring and uncomfortable after some time. They and Wang et al. [Wan+20] also found that the field of view of AR devices is often too narrow to show large amounts of data. May et al. [May+22], on the other hand, found that depth perception can be difficult on an AR HMD with the current available technologies. They also concluded that high accuracy is hard to achieve in AR because of tracking inaccuracies. Lastly, they found that HMDs can have performance issues when they have too many active and detailed objects that need to be rendered in the scene at the same time.

MacWilliams et al. [Mac+03], on the other hand, found that it can be problematic to directly interact with floating menus anchored to the user's view frustum and at a certain distance. In their study, most people tried to keep their hands positioned in the middle of their view frustum, meaning that they could not reach the options anchored to the top of the view frustum. Moreover, they found that the success of this interaction highly depends on the user's physiology. They noticed that children's arms were, in general, too short to cover the distance to where the menus were anchored in the view frustum, while they were within easy reach of an adult. Hence, it is important to keep in mind that items should not be placed too far away when anchoring to the view frustum or observer, when using hand tracking as a method of input. However, this could be solved by using controllers with a raycast in order to interact with items that are placed further away.

Butscher et al. [But+18] found that it can be a bit troublesome to collaborate in AR when wearing HMDs, since they cover the faces of the collaborators, resulting in the inability to read facial expressions. Moreover, it is possible that hand gestures are blocked by virtual elements. Additionally, Wang et al. [Wan+20] found that hand gestures can become tiring when used as an interaction technique. This is because the hands have to be constantly held up in order to be tracked by the HMD. This can be solved by implementing a controller for interaction or as a backup for interaction when the user's arms are tired, since a controller does not need to be held up in order to use it. Lastly, Wang et al. [Wan+20] also found that speech is not a suitable type of input depending on the situation. For example, when the system is supposed to be used in an office space, the user cannot keep talking out loud to interact with the application. This would disturb the colleagues sitting around the user. On top of that, speech input would simply not work in a noisy environment, since the noise would interfere with the speech recognition.

In the context of learning, it was found that an Augmented Reality setup can provide better motivation than its traditional counterpart [RMS13; Dün+12]. Moreover, in the context of spatial understanding of three-dimensional data, AR can prove superior to traditional setups due to its inherent three-dimensional layout [Mar+18b; Dün+12]. In general, it was found that AR is better suited for higher-level navigation, and that it is better to switch to a more traditional method for more detailed navigation or actions [Mar+18b]. Furthermore, the effectiveness of certain tasks could be influenced depending on whether a stereoscopic or monoscopic display is used, since this affects the depth perception of the user [Mar+18b]. As a result, it is difficult to compare the effectiveness of certain tasks across various studies, since there are many parameters that can have an influence on this metric (FOV, tracking accuracies, depth perception...). Regardless, Wang et al. [Wan+20] are convinced that AR will never replace the traditional desktop, but that it can serve as an extension of it.

3.9 Methodology

The design space of this thesis was formed by conducting an extensive investigation of the existing literature around Augmented Reality and Immersive Analytics. First, we conducted research to further familiarise ourselves with Augmented Reality, which led to the discovery of Immersive Analytics. Once we had a few relevant sources, we used ‘Research Rabbit’[1] to search for more related works. We then fed the newly found related sources into ‘Research Rabbit’ again to gain additional related works. We repeated this process a few times until no relevant sources were left to collect.

To keep an overview of the vast amount of literature explored, we composed a summarising table in ‘Google Spreadsheets’ as seen in Figure 3.12. This table contains the publication date, title, subject categories, DOI and URL of all the sources examined during the research. Along with a reading status, a usefulness score and where it would be helpful within the thesis. While a source could be assigned ‘not useful’, it is essential to note that this does not mean the content is useless. This means the source does not contain relevant information concerning the researched topic. When a source is marked as ‘useful’ or ‘very useful’, it holds certain information that could contribute to this thesis. In this case, the source is also categorised into where it could be applied (e.g. in related work or implementation). All these entries have a unique incrementing identifier as well. This way, it is more efficient to refer to a specific source by mentioning the identifier instead of its title or DOI. It is also easier to recall the essential works by their identifier, simplifying the research process. This identifier is used to label all the annotated studies, to label all the entries (in the backend) in the bibliography of this text, as well as the first column (ID) in Figure 3.11. In total, we studied more than 140 different sources, as the summarised table contains 136 entries with a status label other than ‘unstarted’, and not all of the scanned sources were added to the table.

While working through the approximately 140 collected resources, we devised various categories via the identification of similar trends across the studies. The accompanying subcategories indicate slight differences in the implementation of the main categories. For example, all visual Augmented Reality systems have to display their virtual objects through some type of display technique, which forms the main category. However, the implementation of the display technique across the diverse studies varies. Generally, it can be displayed through a handheld display, a head-mounted display or even a projection-based display. These form the three subcategories of the main category (display technique), as discussed in Section 3.3. In total, we devised 7 categories with 22 subcategories combined. Some main categories were based on categories found in related works, while others were newly formed by analysing prior literature and deducing similar trends. This is indicated for every category in chapter 3. From the 136 studied sources, fourteen were selected to be evaluated against the categories of the design space. In order to be selected, the study had to deliver an Augmented Reality system. The fourteen evaluated studies provide an overview of the most implemented techniques across the various categories, as illustrated in Figure 3.11. Section 3.8 then discussed the most important findings of the related works and our general findings across the 136 studied sources. Lastly, subcategories were selected from each category to form the design space for this thesis. As a result, the proof-of-concept application ModulARboard focuses on the augmentation of sight through a stand-alone HMD with an egocentric view. It is intended for individual usage in the application area of presentation & visualisation. Moreover, it attempts to test out all the anchoring mechanisms discussed in section 3.7. ModulARboard is also illustrated separately at the bottom of the table in Figure 3.11, and its implementation will be discussed in the following chapter.

[1] `https://researchrabbitapp.com/`

			Area			Senses					Display					EPM		Colab		System	Anchoring			
ID	Year	Title	Presentation & Visualisation	Industry	Edutainment	Sight	Hearing	Taste	Smell	Touch	Handheld Display	Head-Mounted Display	Projection-based Display	Exocentric	Egocentric	Solo	Collaboration	Stand-alone	Combined	Marker-based	Object-relative	Observer-relative	View frustum	Spatial
4	2013	Fun learning with AR alphabet book for preschool chilc ...		■		■								■		?		■		■				
19	2019	Live data visualization of IoT sensors using Augmente ...	■	■		■					■			■		■			■	■			?	
22	2022	Situated Visual Analysis and Live Monitoring for Manu ...		■		■						■		■	■	■			■	■		■		■
26	2015	AR-Weapon: Live Augmented Reality Based First-Pers ...		■	■	■						■		■	■		■		■		■			■
27	2003	Herding sheep: live system for distributed augmented ...			■	■					■	■	■	■			■		■		■		■	
61	2020	Towards Traceable Design Rationale in Augmented Re ...	■			■						■		■		■		■		■	?			■
82	2018	Clusters, Trends, and Outliers: How Immersive Technc ...	■			■						■		■	■		■		■					■
83	2020	Towards an Understanding of Augmented Reality Exte ...	■			■						■		■		■			■					■
91	2012	Creating interactive physics education books with augı ...		■		■					■			■		■		■		■				
98	2020	Embodied Axes: Tangible, Actuated Interaction for 3D ...	■			■						■		■		■	■		■		■			
101	2020	Personal+Context navigation: combining AR and share ...	■			■						■			■	■	■		■	■			■	
106	2021	Radi-Eye: Hands-Free Radial Interfaces for 3D Interac ...				■						■			■	■		■				■	■	
111	2022	AvatAR: An Immersive Analysis Environment for Humε ...	■			■						■		■		■			■					■
112	2021	The Identification, Development, and Evaluation of BIN ...		■		■						■			■	■			■	■				■
	2025	ModulARboard: Exploring AR Dashboards for Immersi	■			■						■			■	■		■		■		■	■	■
		Total:	8	6	2	15	0	0	0	0	3	12	1	11	7	12	5	5	10	8	4	3	5	8

Figure 3.11: This table summarises the evaluation of fourteen AR studies against the seven categories of the design space. These are the evaluated sources:
4: [RMS13], 19: [NM19], 22: [Bec+22], 26: [Zhu+15], 27: [Mac+03], 61: [CCV20], 82: [But+18], 83: [Wan+20], 91: [Dün+12], 98: [Cor+20], 101: [Jam+20], 106: [Sid+21], 111: [Rei+22] & 112: [May+22]

ID	Published	Title	Type	Subject	DOI	URL	Status	Usefulness?	Where useful?
67		VR Development (Unity)	Utility	Development		https://learn.unity.com/	Unstarted	Not useful	
68	24-1-2017	The design space	Website	Information		https://medium.com/ge	Fully read	Useful	
69		ActiTouch	Website	Interaction AR		https://www.youtube.co	Unstarted		
70		Immersion	Website	Information AR		https://www.interaction-	Skimmed		
71		Examples of marker based AR of Unity	Website	AR Information		https://learn.unity.com/t	Skimmed	Useful	
72		Categories of AR applications of Unity	Website	Information AR		https://learn.unity.com/t	Skimmed	Useful	
73		Aprio - AR advertising: 10 examples	Website	AR Information		https://www.arpr.io/blog	Skimmed	Useful	
74	9-12-1968	A head-mounted three dimensional display	Paper	AR	https://doi.org/10.1145/	https://dl.acm.org/doi/a	Unstarted	Very useful	
75	27-3-2024	24+ augmented reality stats (2024 - 2028)	Website	AR Information		https://explodingtopics.	Skimmed	Very useful	
76	6-11-2016	Immersive Analytics: Exploring Future Interaction and Visualization Technologies for Data Analytics	Paper	Visualization AR	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Skimmed	Not useful	
77	17-10-2017	Investigating the Use of Spatial Interaction for 3D Data Visualization on Mobile Devices	Paper	AR Interaction Visualization	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Mostly read	Not sure	
78	1-8-1997	A Survey of Augmented Reality	Paper	AR	https://doi.org/10.1162/	https://www.cs.unc.edu	Mostly read	Useful	Introduction arguments
79	7-8-2002	Recent advances in augmented reality	Paper	AR	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Mostly read	Very useful	Category argumentation
80	15-9-2008	Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR	Paper	AR	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Mostly read	Very useful	Category argumentation
81	1-10-2019	Evaluating an Immersive Space-Time Cube Geovisualization for Intuitive Trajectory Data Exploration	Paper	AR Interaction Visualization	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Skimmed	Useful	Category example
82	19-4-2018	Clusters, Trends, and Outliers: How Immersive Technologies Can Facilitate the Collaborative Analysis of Multidimensional Data	Paper	AR	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Skimmed	Useful	Category example
83	23-4-2020	Towards an Understanding of Augmented Reality Extensions for Existing 3D Data Analysis Tools	Paper	AR Visualization	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Summarized	Maybe	Category example
84	23-3-2019	IATK: An Immersive Analytics Toolkit	Tool	AR Development	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Read summary	Not sure	Development
85	1-1-2005	Beyond the 'wow' factor: developing interactivity with the interactive whiteboard	Article	AR Interaction		https://www.researchga	Read abstract	Not useful	
86	31-8-2011	Handbook of Augmented Reality	Book	AR	https://doi.org/10.1007/	https://link.springer.com	Unstarted		
87	1-11-2007	Toward a deeper understanding of the role of interaction in information visualization	Paper	AR Visualization Interaction	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Skimmed	Maybe	
88	8-12-2021	TangibleData: Interactive Data Visualization with Mid-Air Haptics	Paper	Interaction AR Visualization	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Skimmed	Not sure	Category example
89	7-5-2021	Grand Challenges in Immersive Analytics	Paper	AR Visualization	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Mostly read	Useful	Related work Introduction arguments
90	11-12-2013	Augmented Reality Learning Experiences: Survey of Prototype Design and Evaluation	Paper	Learning AR	https://doi.org/10.1109/	https://ieeexplore.ieee.o	Skimmed	Not useful	
91	26-11-2012	Creating interactive physics education books with augmented reality	Paper	Learning AR Interaction	https://doi.org/10.1145/	https://dl.acm.org/doi/1	Skimmed	Maybe	Category example
92	1-10-2006	Augmented reality applications in manufacturing: a survey	Paper	AR	https://doi.org/10.1080/	https://www.tandfonline	Read abstract		

Figure 3.12: This table contains the majority of researched sources with accompanying information, status and notes. This table provided the structure needed in order to perform the extensive literature study. The implemented identifiers were utilised everywhere throughout the research process.

Chapter 4

ModulARboard

Traditional data dashboards designed for standard monitors have been around for a few decades and have gone through many stages of improvement. However, they still have some shortcomings. Currently, one of the biggest shortcomings of traditional data dashboards (as mentioned in section 3) is that they have a limited amount of space to work with. The amount of available space is tied to the size of the screen it is visualised on. When additional information needs to be shown and there is no room available, the content needs to be scrollable or hidden behind an extra menu, which makes it less intelligible. Another solution could be to zoom out on all the content, but this could make the information too small to read, which would require zooming in and panning around the content. This limited amount of space has motivated researchers to extend the traditional desktop or replace it altogether with Augmented Reality to achieve a larger work area [But+18; Wan+20]. This motivation is often accompanied by the second issue of traditional data dashboards, namely that inherently three-dimensional visualisations can be hard to understand on a two-dimensional screen. This is because the 3D visualisation needs to be projected onto a 2D screen, usually through orthographic or perspective projection. It can be even harder to understand the 3D visualisation if the user is unaware of what type of projection is implemented. Because of this, researchers try to implement AR to display three-dimensional or multi-dimensional data. Augmented Reality can make it easier to understand three-dimensional visualisations due to its own 3D nature, which allows users to intuitively walk around the visualisation, instead of having to manoeuvre it around with a mouse or specific shortcuts.

While many researchers try to improve collaboration or multi-dimensional data visualisations through the implementation of AR (as mentioned in chapter 3), this will not be the focus of this thesis. Instead, it will focus on the limited amount of space available on traditional desktops and how Augmented Reality can be leveraged to reimagine traditional data dashboards to facilitate Immersive Analytics. More specifically, it will focus on different possibilities of positioning information within the virtual environment, while providing customisation opportunities in order to realise a personalised Augmented Reality dashboard. Which ties back to the second and third subquestions of the research question. To achieve this, ModulARboard was developed, a proof-of-concept application, which implements a number of anchoring mechanisms devised by many related works in chapter 3, along with several customisation options. All these specifics and the flow of ModulARboard are discussed in the following section, while the implemented technologies are explained in section 4.2.

4.1 Application flow

ModulARboard is a modular Augmented Reality dashboard, meaning that the user is enabled to divide an otherwise traditional data dashboard into modules and position them in the augmented environment. There are multiple ways to navigate through the application to achieve the same goal, as can be seen in Figure 4.1, which illustrates the flow of the application. It all starts with ① reading the JSON file and initialising all the dashboard components with their according data. This includes matching the component names with the default icons available in ModulARboard (section 4.2). When there is no icon available for the component, a default one is provided. Once the A dashboard is loaded, there are four actions available. The user can 'right-click' (use the bumper) on a component to open a pop-up menu to ② choose a visualisation directly or ③ anchor the component's label to the augmented environment. It is also possible to select multiple components to ④ group together and anchor the group's label to the environment. The anchoring mechanisms will be explained in section 4.1.5.

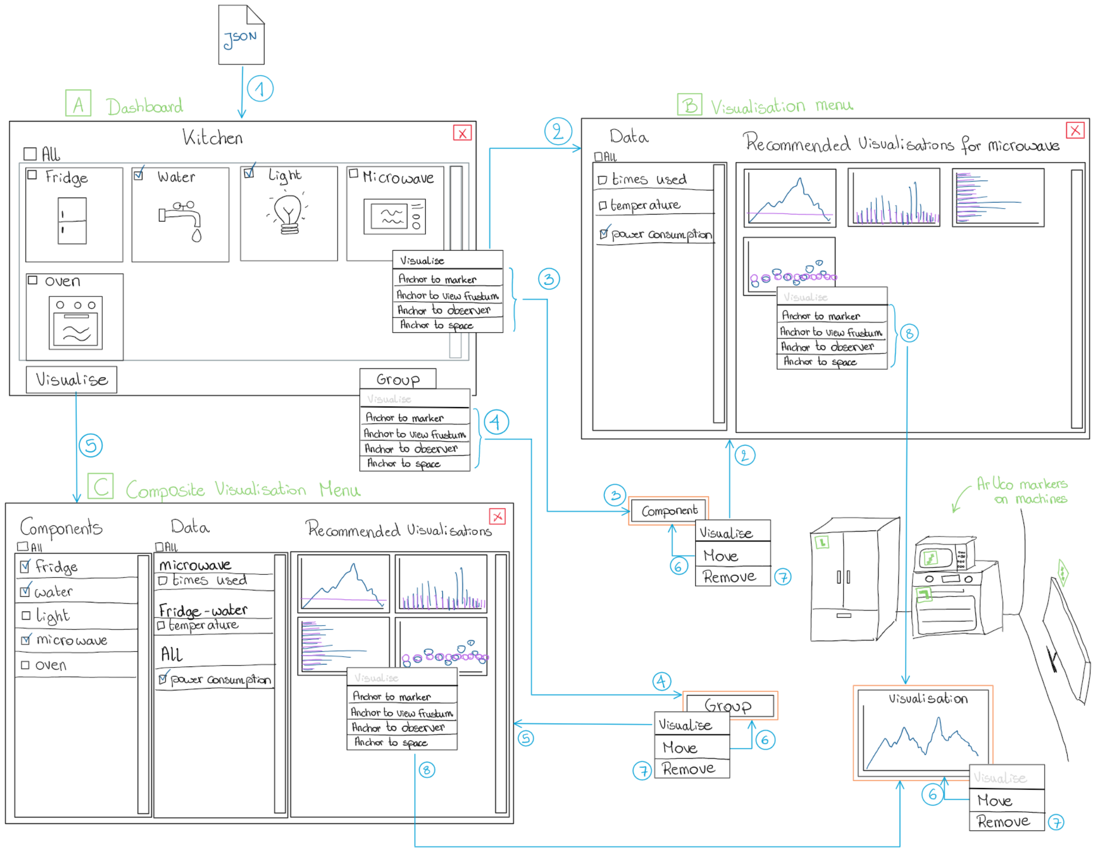


Figure 4.1: General outline of the flow of ModulARboard:
① Loading data from a JSON file to build the dashboard.
② Opening the visualisation menu for a single component.
③ Choosing an anchoring mechanism to anchor a component tag to space.
④ Selecting multiple components to group them and anchor them to space.
⑤ Opening the composite visualisation menu.
⑥ Moving components within the selected anchoring mechanism and resizing.
⑦ Removing the selected component.
⑧ Anchoring a visualisation, from one of the visualisation menus, to space.

Lastly, the user can ⑤ open another menu to build composite visualisations. In this menu ⑤, a single or multiple components can be selected, which dynamically loads the available data options per group of components. Then in both menus ② and ⑤, it is possible to select a single or multiple data options, which dynamically loads the recommended visualisations for the selected data. The recommended visualisations are customised depending on the data types of the selected data and whether it's a single component or multiple, and whether a single data option was selected or multiple. This will all be explained in sections 4.1.2 and 4.1.3.

When ‘right-clicking’ the desired visualisation, a pop-up menu is presented to ⑧ anchor the visualisation in the preferred location, similarly to ③ anchoring the label (of a group ④).

The anchoring process is the same for (group) labels and visualisations. The orange border around the item indicates that it can be moved around and scaled to the user’s liking. The specific interaction mechanisms to perform these operations are explained in section 4.1.4. Once the item is manoeuvred into the desired position, the user can confirm the placement by pressing the bumper again. Now that the item is anchored to the augmented environment, it can be accessed again with the bumper to bring up another pop-up menu. In this menu, the user can always choose to ⑥ move the item again or to ⑦ remove it from the scene. If the item is a label, then it is also possible to ② open the B visualisation menu or ⑤ the C composite visualisation menu if it is a group label.

All menus can be closed at all times by clicking the red close button in the top-right corner with the trigger of the controller. They can also be moved around by ‘grabbing’ the top of the menu by holding the trigger and then dragging the menu to the desired location. This proves useful when the menu is spawned in an inconvenient location or orientation. The A dashboard can be opened and closed by using the menu button on the controller. When the dashboard is not visible (because it is toggled off or simply not in sight), it will spawn in front of the user where the controller is pointing when the button is pushed. Otherwise, the A dashboard will close if it is in sight when the button is pushed.

This concludes the high-level outline of ModulARboard as illustrated in Figure 4.1. In the following sections, the different mechanisms and design choices will be explained. As well as all the available visualisations and customisability options. Throughout these sections, the same numbers and names will be utilised to provide a clear understanding of where the explained part is located in the flow of the application. It is also important to keep in mind that most screenshots are taken in the scene editor of Unity to provide higher-quality images, since the application simulator provides low-quality graphics, as can be seen in Figure 4.2. Because of this, most images have a white background instead of being shown in an augmented environment. However, ModulARboard is an Augmented Reality application. Hence, all menus are in reality located in an AR environment.

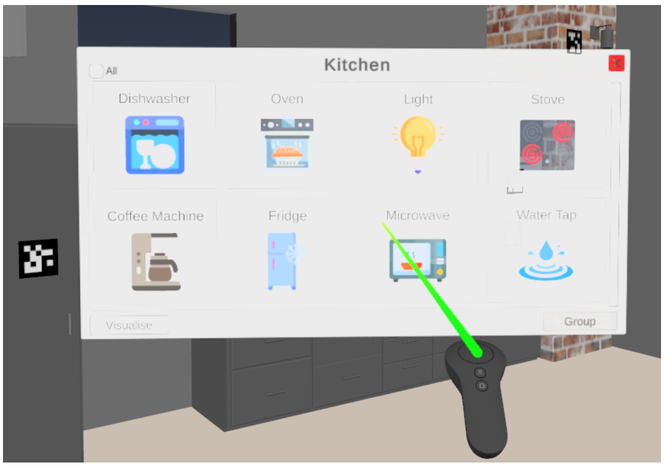


Figure 4.2: Low-quality graphics of the application simulator in Unity provided by Magic Leap. This serves as a motivator to take screenshots outside an augmented environment to provide higher-quality images in order to perceive the application properly. Keep in mind, however, that all screenshots provided outside the augmented environment are in reality located in one.

4.1.1 Main dashboard

The A main dashboard is the core of ModulARboard and is shown in Figure 4.3. All the components available for information visualisations are located in the main dashboard. Or when there are no components available, a feedback message is displayed to indicate so (Figure 4.12a). As mentioned in the previous section, there are four different ways to navigate through the application starting from this dashboard. The user can choose to ② visualise a single component directly, ③ anchor a component's label to the environment, ④ select several components to group them together and then anchor the group to the environment. Lastly, it is possible to ⑤ compose a custom visualisation. This functionality is supported by the **adaptive visualisation recommendation system**. It consists of two main compartments which form the backend of the C composite visualisation menu. The first compartment stands in for the adaptive loading of data options and visualisations. It ensures that the application responds immediately to the user's requests. This will be further explained in section 4.1.2. The second part, on the other hand, stands in for the visualisation recommendations, which will be discussed in section 4.1.3. The B visualisation menu consists solely of the visualisation recommendation system. All the other pathways (②, ③ & ④) with their specifics are now explained in more detail.

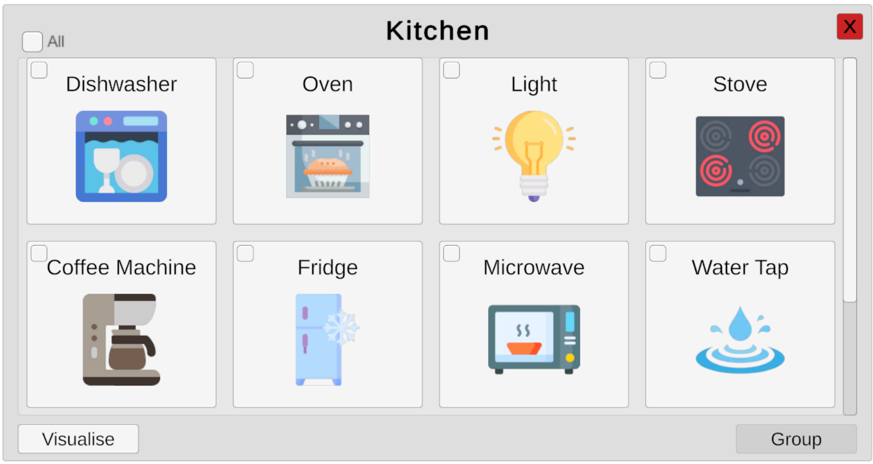


Figure 4.3: Screenshot of the A main dashboard of ModulARboard. From this dashboard, it is possible to 'right-click' a component in order to open the pop-up menu illustrated in Figure 4.4a. From here, the user can ② visualise the component directly or ③ anchor the component's label somewhere in the augmented environment. Other than that, it is possible to ④ select components by their checkboxes to group them together or choose to ⑤ compose custom visualisations.

② To visualise a component directly, the user can open a pop-up menu by 'right-clicking' a component in the dashboard by using the bumper on the controller and aiming the ray at the component (Figure 4.4a). This action closes the main dashboard and opens a new menu (B) to browse visualisations recommended for the selected component (Figure 4.5). From this same pop-up menu, it is also possible to ③ anchor the component's label somewhere in the environment. The specifics depend on the chosen anchoring mechanism (section 4.1.5). However, the general process remains the same. The component's label is spawned in its start position with an orange border around it (Figure 4.4b). From there, the user can move it around and scale it to the desired format. When the user is satisfied with the scale and position, they can press the bumper again to confirm the anchoring, and the orange border will disappear. In case the user is unsure about the chosen anchoring mechanism, the process can be cancelled by pressing the menu button on the controller. Once the label is anchored in the desired position, the user can open the pop-up menu of the label to visualise, move or remove the component (Figure 4.4c). The visualise option opens the same menu to browse the recommended visualisations, as mentioned before (Figure 4.5). The move option reinitialises the anchoring process relative to its current position, while the remove option destroys the label, erasing it from the environment.

(a) Pop-up menu opened for the 'Light' component in the main dashboard.

(b) 'Light' component's label spawned in the default position with an orange border to indicate edit mode.

(c) Anchored 'Light' label with its pop-up menu opened.

Figure 4.4: Single component label anchoring process: **(a)** the user chooses an anchoring mechanism from the list (options 1 - 4), **(b)** the label is spawned in the default position where the user can move it around and scale it. When the label is anchored, the user can **(c)** open the pop-up menu of the label for further actions.

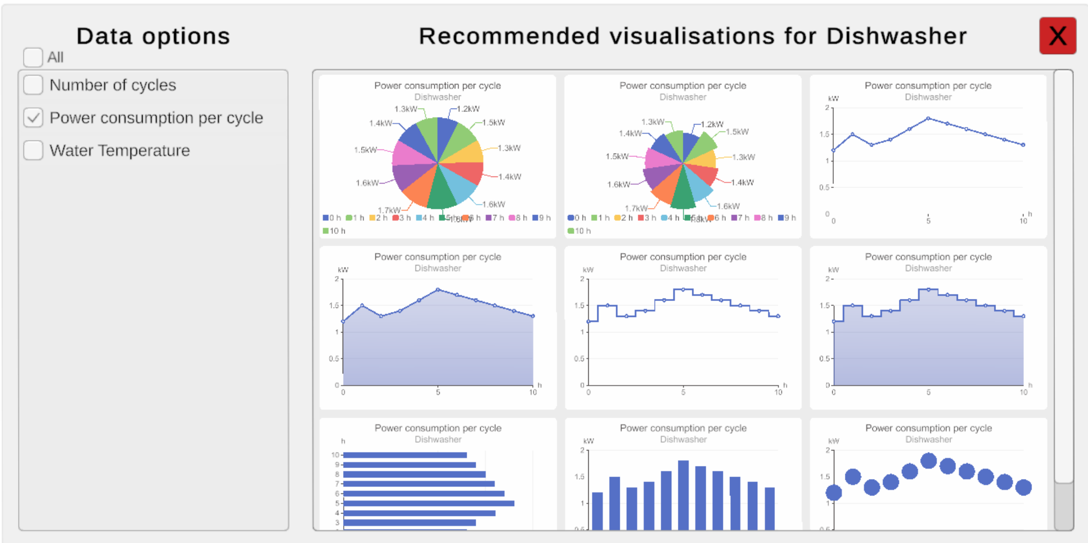


Figure 4.5: Screenshot of the B Visualisation menu with recommended visualisations for the dishwasher. There are three different data options available for the dishwasher: The number of cycles, the power consumption per cycle and the water temperature. In this case, only the power consumption per cycle is selected with the recommended visualisations shown in the panel on the right.

④ Other than the options available for a single component, it is also possible to select multiple components to group them together. This can be done by either clicking the checkbox itself or anywhere on the component tile, since the entire component is clickable for easier interaction. Once at least two components are selected, the 'group' button becomes available, as can be seen in Figure 4.6. This button resembles a drop-down menu, opening the same pop-up menu as before, with the 'visualise' option now disabled. Since this functionality is embedded in the composite visualisation menu, which is activated by the 'Visualise' button on the bottom left of the dashboard. Moreover, the purpose of creating a group is to actually save the multiple components together for easier access later on. To achieve this, a new blank label is generated and placed in the default position (Figure 4.7a). The anchoring process is exactly the same as the anchoring of a single component's label, as explained in the previous paragraph. The only difference is that after the group label is confirmed, the blank label can be clicked with the trigger to summon the keyboard (Figure 4.7b). The user can now give a name to the group and then close the keyboard again by pressing Enter or the 'hide keyboard' button (Figure 4.7c). The name can always be altered by clicking it again when not in edit mode. From here on out, the same pop-up menu can be summoned by 'right-clicking' the tag as in Figure 4.4c. However, the visualise option now opens the C composite visualisation menu instead of the B visualisation menu.

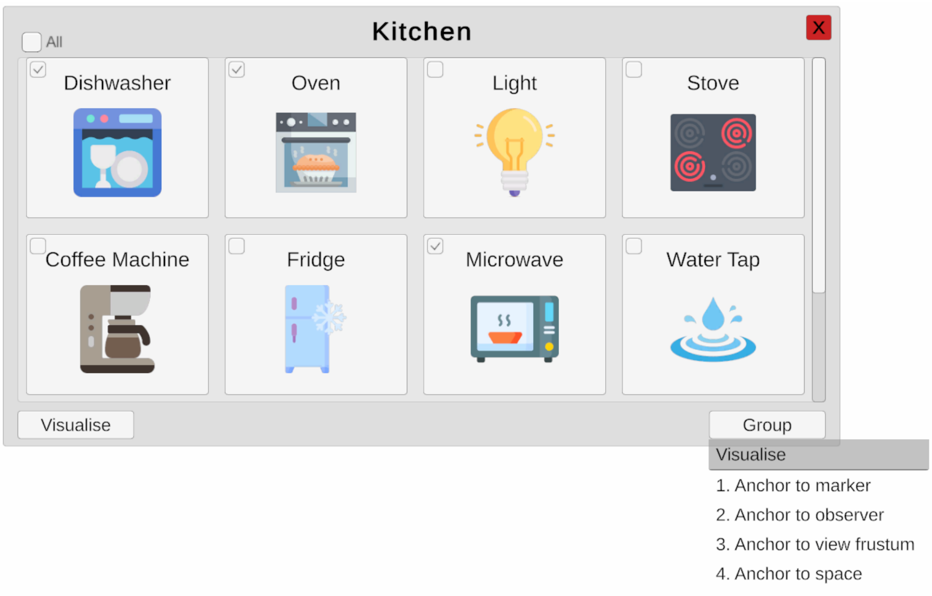


Figure 4.6: Screenshot of the A main dashboard with three selected components. When at least two components are selected, the group button becomes enabled. This button serves as a dropdown menu with the four available anchoring mechanisms. This makes it possible to ④ create a group of the selected components and place its label somewhere in the augmented environment.

(a) The group label with the orange border indicates that it can be moved around and rescaled. As indicated in the placeholder text, the name can be set when the placement is confirmed.

(b) Once placement is confirmed, the name can be set by clicking (trigger) the label. A keyboard pops up to enter the name.

(c) The name can be entered using the controller and its raycast. When ready, the keyboard can be closed again by pressing Enter or the hide keyboard button.

Figure 4.7: Group label anchoring process.

4.1.2 Adaptive data loading

As mentioned before in section 4.1.1, the adaptive data loading forms the first part of the adaptive visualisation recommendation system, which makes up the entire backend of both visualisation menus B and C. The mechanics of the composite visualisation menu (C), however, are slightly more extensive compared to the single-component visualisation menu (B). This is because it has an extra panel to select components, as can be seen in Figure 4.8 as opposed to Figure 4.5. Other than that, the functionality for both menus is identical.

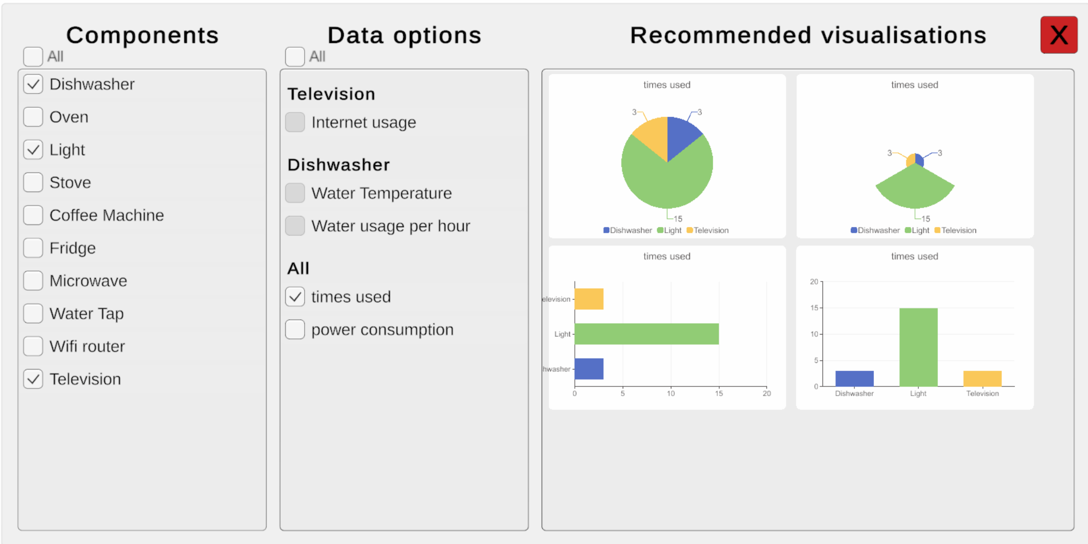


Figure 4.8: Screenshot of the C Composite visualisation menu. In the list of components, there are three selected components: the dishwasher, the light and the television. This makes 5 different data options available in three different groups (Television, Dishwasher & All). In this case, only the times used data option is selected for all three selected components, which results in four different visualisations in the panel on the right.

It was decided to have two separate menus, instead of using the C composite visualisation menu everywhere. The reasoning behind this is that not every user is going to want to group multiple components together. If it is preferred to simply add single-component visualisations to the environment, the extra panel for selecting components is redundant and hinders. When the first panel is absent, there is more room to show visualisations, resulting in three charts per row instead of two (Figure 4.5 & Figure 4.8). It could be argued that the composite visualisation menu can simply be made wider to fit the same number of visualisations in a row. However, the field of view (FOV) of the Magic Leap 2 (and other AR HMDs) is limited, meaning that the full menu would not be visible at a glance. To solve this, the user has to look around the menu in order to see everything. Another solution would be to step further away from the menu, but this could result in the content being too small to read. Because of this, all menus in ModulARboard have a limited width to prevent these issues from occurring. This explains the reasoning behind the two separate menus with similar functionality. Because of this identical process, except for the first panel, the adaptive visualisation recommendation system will be explained in terms of the composite visualisation menu (Figure 4.8). This also prevents having to explain the same functionality twice.

The adaptive visualisation recommendation system starts with the available components list. Every component has a number of data options, each with a name, measure (identifier) and data depending on the type. The measure identifier is used for grouping purposes, as will be explained later on. In the current version of ModulARboard, there are three different supported data types: discrete, fractional and continuous data. However, these can be easily extended.

Discrete data is a single number representation, which can be used to represent data such as the number of times something was used. Secondly, fractional data has a value and a maximum value, which can be used to encode percentages or fractions. Continuous data, on the other hand, can be used to encode time series data. It has a series of labels and a series of values, which can be used to represent x- and y-values. All the components with their data options are immediately imported from the main dashboard and displayed in the first panel. When a component is selected, the system automatically adds all the available data options to the list in the next panel. Contrarily, the deselection of a component results in the removal of its data options from the list. In the scenario that multiple components are selected (Figure 4.8), the data options are combined, as illustrated in Figure 4.9.

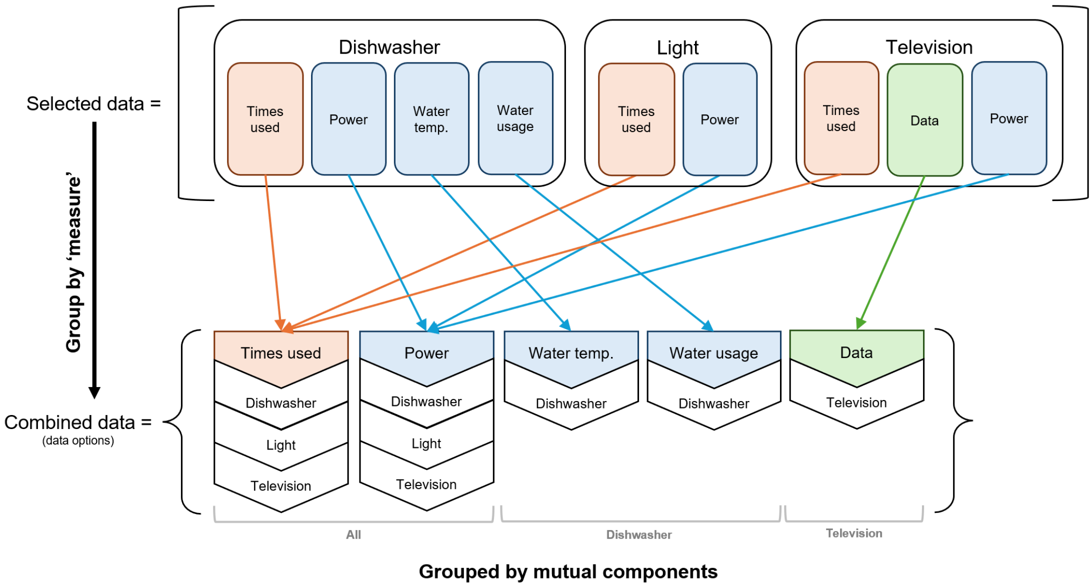
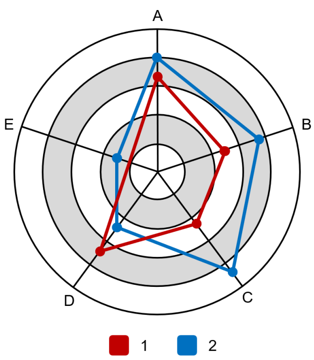


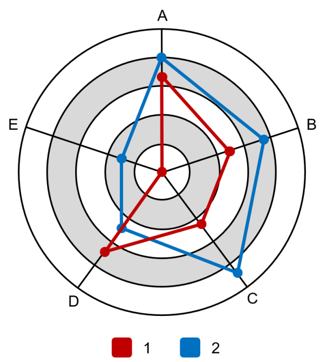
Figure 4.9: Illustration of the grouping process performed every time a component is selected or deselected. The 'measure' identifier is responsible for the correct grouping of components. It indicates when two data options can be grouped together, even when their name differs. In this schema, the selected components are shown at the top (Dishwasher, Light & Television). The dishwasher has 4 different data options, the light has 2, and the television has 3. These data options can be one of three different types (orange = discrete, blue = continuous & green = fractional). As indicated by the arrows, the data options of each component are grouped by the measure they have in common. In total, there are five different measures (Times used, Power, Water temperature, Water usage & Data), meaning that there will be five combined data options when the grouping process is finished. Each measure is now a newly formed data option containing the data from each selected component. The data is then grouped together by the components they have in common, as illustrated by the brackets at the bottom. When loading the newly formed data options in the middle panel, there will be three different groups of data options (Television, Diswhasher & All), as illustrated in Figure 4.8.

The grouping process happens based on the measure variable that each data option contains. This measure was implemented to use as an identifier to indicate that various data options with a similar value can be grouped together. The name was not used for this purpose since it can vary slightly from component to component, which would make it hard to group them together. A unique identifier makes this process easier and more straightforward. The measure identifier is illustrated in Figure 4.9 in the coloured rounded squares and arrowed squares, and the list at the bottom contains the selected components from the first panel. As illustrated by the arrows, the data options containing the same measure identifier get grouped together and form a new data structure (data option). In Figure 4.9, there are five different data options available as a result of the grouping process. 'Times used' and 'Power' are available for the dishwasher, light and television. ‘Water usage’ and ‘Water temperature’ are available only for the dishwasher, and ‘Data’ is available only for the television. After this step, the data options are grouped together by the set of components they have in common. In this example, this results in three groups. The first group contains ‘Times used’ and ‘Power’ since they both contain all three components. The second group contains ‘Water temperature’ and ‘Water usage’, since they both contain the dishwasher component. The last group only contains the ‘Data’ option, since it is the only measure that contains solely the television.

The data options are then displayed in the middle panel of C composite visualisation menu, with a separate title per group. It is only allowed to select data options from the same group, since it is not possible to visualise data for one component if it is absent for another component. This is because of a limitation of the visualisation package XCharts (section 4.2). The package does not provide the ability to build a radar chart with missing values for certain parameters (figure 4.10a). If attempted, this would result in having values of zero for the component with the absent data, misleading the user into thinking the values are zero instead of non-existent (Figure 4.10b). As a result, the design choice was made to only group components together with mutually available data options. Because of this, the checkboxes of the other groups become disabled when a data option is selected (Figure 4.8).



(a) Radar chart with skipped missing values: This would be the desired solution to missing values for certain parameters. This radar chart skips over the missing values and connects the line straight to the next parameter.



(b) Radar chart with missing values substituted with zeros: This is the actual result of missing values for parameters by XCharts. Missing values are substituted with zero values. This misleads the user into thinking data is present, while it is not.

Figure 4.10: This illustrates the difference between **(a)** the desired radar chart and **(b)** the actual functionality of the radar chart by XCharts. It features a radar chart with five different parameters (A, B, C, D & E) and two series (1 & 2). Series 1 has data available for all parameters, while series 2 is missing data for parameter E.

Lastly, to improve computing performance and ensure that this process does not need to be repeated, the available data options are stored per set of selected components. This way, when a certain combination of components has been selected before, the resulting data options can simply be loaded from memory and don’t need to be computed again.

After the grouping process, as illustrated in Figure 4.9, the middle panel displays the newly formed data options. The user can then select data options to request recommended visualisations. This selection forms a new combination of a number of components and data options. Depending on the number of components and data options present in the selected data, it can fall into one of four classes, as illustrated in Figure 4.11. If the combined data contains a single component and a single data option, it falls into the first class called ‘**single**’ data. Secondly, if a single component is selected with multiple data options, it falls into the second category called **(single-)combined**. Both these classes of data can be formed in menus B and C, since they require only one component to be selected. The last two classes, however, are only available in the C composite visualisation menu, since the B visualisation menu cannot select multiple components. When the combined data contains multiple (at least two) components with only

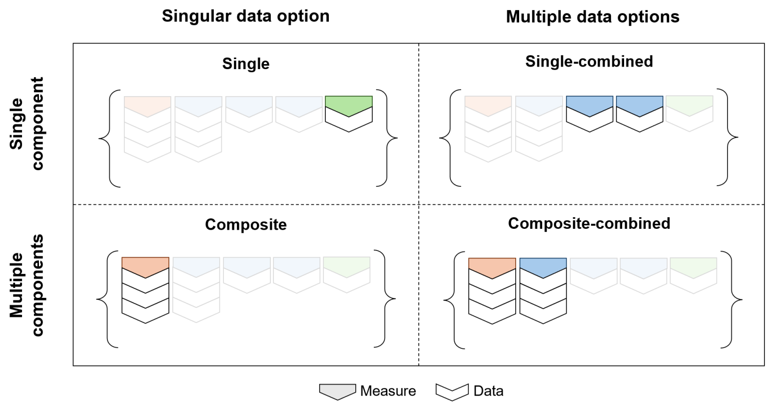


Figure 4.11: After the components are grouped together by their measure identifier, as explained in Figure 4.9 and data options are selected, the newly formed data can fall into one of four categories/classes. When a single component is selected in the first panel, it is only possible for the combined data to fall into one of the classes in the top row. When only one data option is selected, the combined data falls into the **'Single'** data class. On the other hand, when multiple data options are selected, the combined data falls into the **'Single-combined'** data class, or simply the **'Combined'** data class. When there are multiple components selected in the first panel, the possible classes of the data are limited to the ones in the bottom row. Subsequently, if a single data option is selected, the combined data falls into the **'Composite'** data class. Similarly to single data, if multiple data options are selected, the combined data falls into the **'Composite-combined'** data class. In conclusion, the data class can partly be known depending on the number of selected components. However, the class only becomes fully clear when the number of selected data options is known as well.

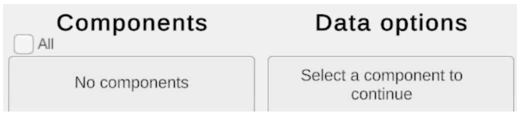
one data option (that they have in common), it falls into the third class called **composite** data. The combined data falls into the last class, called **composite-combined** data, when multiple components and data options are grouped together. These four data classes form the foundation of the visualisation recommendation system explained in section 4.1.3. This is because these four data classes directly decide what visualisations will be generated by the system.

The previous explanation can be clarified by analysing the middle panel in Figure 4.8 and its component grouping process as illustrated in Figure 4.9. There are three groups present with available data options. The first 'group' consists of a single component, the television, as displayed in the title. It only contains one data option, since the television is the only component with the 'Data' measure (the data option is called Internet usage). Similarly, the second group consists of a single component, the dishwasher. It contains two data options, since the dishwasher is the only component with the 'Water usage' and 'Water temperature' measures. So, even though three components are selected in the visualisation menu, there are two groups consisting of only one component. Consequently, if only the first option, 'Internet usage', were selected, the grouped data would contain only one component (television) and only one data option (Internet usage). In this scenario, the combined data falls into the first class of 'single' data because it consists of only one component and one data option. With this setup, it could never become 'single-combined' data, because the television group only contains one data option, meaning that it is not possible for multiple data options to be selected.

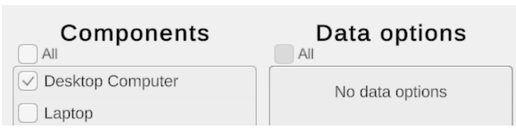
Similarly to the selection of the first data option, if only 'Water usage' or only 'Water temperature were selected, the combined data would fall into the category of 'Single' data. This is because it would contain only one component and one data option. However, if both 'Water usage' and 'Water temperature' were selected in the second group, resulting in all options being selected, the combined data would fall into the category of 'Single-combined' data or simply 'Combined' data. This is because it would contain one component with multiple (two) data options. Contrarily, the last group consists of three components (dishwasher, light & television). This group has two available data options (times used and power). This is because all three components have data with the 'Times used' and 'Power' measures. In Figure 4.8, only the 'Times used' data option is selected. As a result, the combined data contains a single data option combined from three components and falls into the 'Composite' data class. However, if the second data option, 'Power', were selected as well, the combined data would consist of two data options from three different components. As a result, it would fall into the 'Composite-combined' data class. In conclusion, the combination of the number of components and selected data options defines the data class of the combined data.

The last panel showcases the recommended visualisations according to the selected components and their selected data options. Similarly to the data options, the visualisations are also saved to improve computing performance. When a certain combination of components and data options has been made before, the resulting visualisations can simply be retrieved, instead of having to generate them all again. Depending on the data class from Figure 4.11, the visualisations are built differently, as will be explained in section 4.1.3. In the case that there are no recommended visualisations for the selected data options and/or components, a feedback message is displayed indicating so (Figure 4.12d). However, this is currently only possible when a JSON file is loaded with not yet supported data types or when there is simply no data available. Similarly to the feedback message of the recommended visualisations, there is one for the data options and components as well. These get displayed when there are no data options available for the selected components (Figure 4.12c) or if there are no components available (Figure 4.12a & 4.12b). In this case, a different JSON file with available components and accompanying data with supported data types should be loaded.

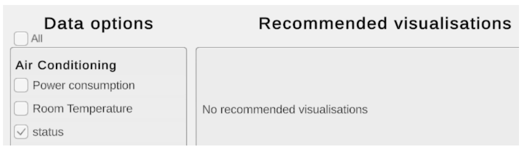
(a) No available components on the dashboard



(b) No available components in the visualisation menu



(c) No data options for 'Desktop Computer'

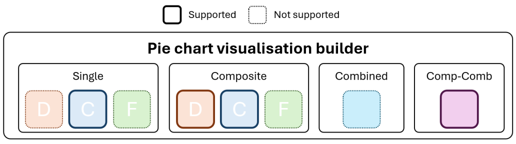


(d) No recommended visualisations for the status of the Air Conditioning

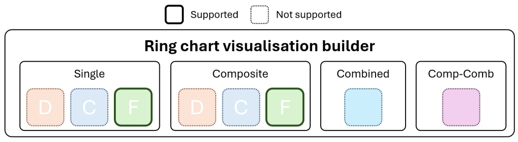
Figure 4.12: The different messages implemented throughout the application to provide feedback on the status of the availability of components, data options and recommended visualisations.

4.1.3 Visualisation recommendation system

As mentioned before in section 4.1.2, the visualisation recommendation system forms the second half of the adaptive visualisation recommendation system. It is responsible for the generation of recommended visualisation, and since ModulARboard is a proof-of-concept application, it provides a set of default visualisations. Each visualisation has a corresponding builder responsible for generating an instance of the visualisation based on the provided data class and type. As mentioned before, there are four possible classes of data (Figure 4.11) and currently three data types (Discrete, Continuous & Fractional). The builder consists of various modules, with each one being responsible for the generation of a visualisation for a certain data class and type. It is also possible to add modules for combined data focused on a single data type or a certain combination of data types, instead of only combining all data types together. However, this functionality is not integrated in this version of ModulARboard. The single and composite data classes (first column) consist of one component or multiple components, respectively, with only one data option. Because of this, a module must be implemented for any data type for which a visualisation is desired. The combined and composite-combined data classes, on the other hand, consist of multiple data options. In this case, all the various data options are combined together regardless of their type. This results in a single module needed for these data classes. If a module is present, a visualisation can be generated. Otherwise, the builder does nothing. For example, a pie chart can represent continuous data (Figure 4.15a), combined continuous data (Figure 4.18a & 4.18c), composite discrete data (Figure 4.19a) and even composite-combined data (Figure 4.22c). This is because the pie chart builder has modules for these types of data, as illustrated in Figure 4.13a. However, a visualisation can also only support one data type, like the ring chart. It only has recommended visualisations for single fractional data (Figure 4.17) and composite fractional data (Figure 4.21). Because the ring chart builder only has modules for single and composite fractional data, as illustrated in Figure 4.13b. In conclusion, any visualisation can be supported by providing a builder with modules according to the desired data class (Figure 4.11) and data types. On top of that, it is possible to extend the current set of default visualisations by implementing new builders or even data types with corresponding builder modules. Both data types and builders can be derived from the provided base classes through polymorphism. This makes the system easily extendable and alterable.



(a) The pie chart builder in ModulARboard has support for single continuous data, composite discrete data, composite continuous data and composite-combined data.



(b) The ring chart builder in ModulARboard has support for single fractional data and composite fractional data.

Figure 4.13: A visualisation builder consists of different modules for the four visualisation classes of Figure 4.11. In order to support a certain data type, a module must be implemented to generate the desired visualisation for that data type. Single and composite visualisations are generated for a single selected data option. Because of this, there is a different module per data type. In this case, they are Discrete, Continuous & Fractional data. On the other hand, (single-)combined and composite-combined visualisations combine all types of data together to form one big visualisation. The supported data types and builder modules can easily be extended by deriving from the provided base classes.

As mentioned before in section 4.1.2, the adaptive data loading ensures that when a component is selected, it immediately updates the available data options. Similarly, selecting a data option immediately updates the recommended visualisations. This happens at the very last step of the adaptive visualisation recommendation system, by feeding the combined data in turn to every builder in the pipeline. If a module is present, it generates a visualisation, which is added to the list. If not, the process is continued by consulting the next builder in the pipeline. This process is illustrated in Figure 4.14, with one of the many possible scenarios for each of the four different data classes. In the first scenario, single data of type green (fractional) is fed to the pipeline of builders. However, there is only one builder with a green module present in the pipeline. Hence, this is the only builder which can generate a visualisation for green single data. As a result, all the other builders will be skipped. In the second scenario, multiple components are selected with only the orange (discrete) data option, which is classified as composite data. This results in two generated visualisations, because there are two builders present in the pipeline, equipped with modules specialised in orange composite data visualisations. This means that the other builders will be ignored, since they do not have modules for this data class and type. Thirdly, when a single component is selected with multiple data options, which is classified as single-combined data, there will be two visualisations generated. This is because there are two builders present in the pipeline, which can generate combined visualisations. Again, the remaining builders are ignored. Lastly, there are three builders present in the pipeline with modules for composite-combined data, resulting in three generated visualisations.

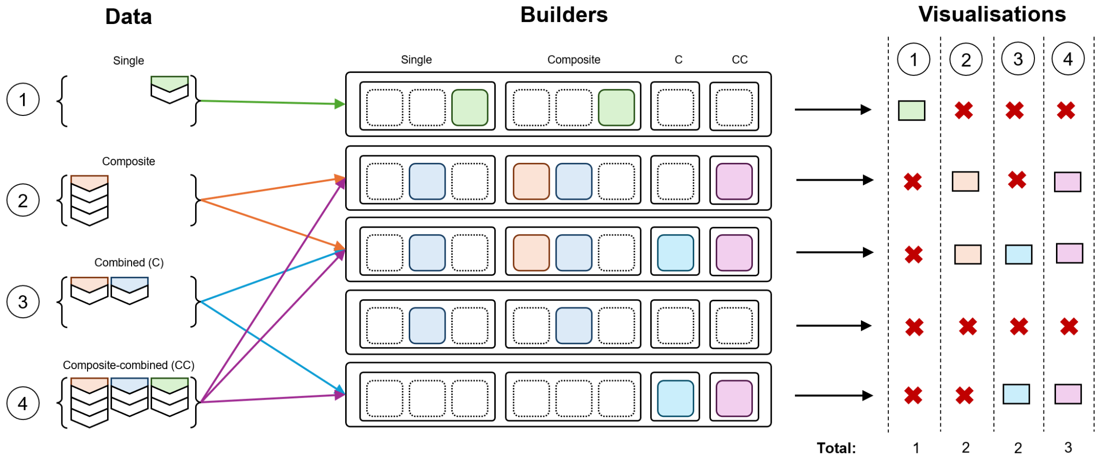


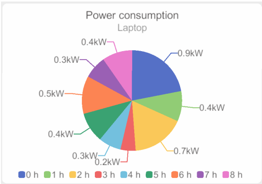
Figure 4.14: Illustrated flow of the visualisation recommendation system. It all starts by receiving the combined data, which falls into one of the four categories from Figure 4.11. Next, there is a pipeline of designated builders with specialised modules for variations of their visualisations. Depending on these four classes of data and data types, the builders in the pipeline either generate a visualisation if a module is present, indicated by the coloured rectangle. Otherwise, the builder is ignored (as indicated with the red cross) and the next one in line is consulted.
Scenario 1: Single green data is available. There is only one builder with a green module, resulting in only one visualisation.
Scenario 2: Orange composite data is selected. There are two builders available with orange modules, which means two visualisations are generated.
Scenario 3: Combined data (orange & blue) is provided as input, with only two builders having specialised modules (turquoise). Two visualisations are generated. The other builders are skipped.
Scenario 4: Composite-combined data is selected. There are three builders in the pipeline with a specialised module (pink), resulting in three generated visualisations.

All the builders in the pipeline in Figure 4.14 are builders that are actually present in ModulARboard. The first builder (at the top) represents the ring chart builder as previously illustrated in Figure 4.13b. The second builder in the list displays the pie chart builder as was already explained in Figure 4.13a. The third builder in line is the column chart builder, since it has support for single continuous, composite discrete, composite continuous, combined and composite combined data. The fourth builder from the top, or second from the bottom, represents the scatter chart builder. It only has support for single continuous and composite continuous data. Lastly, the builder at the bottom illustrates the radar chart builder, with support for combined and composite-combined data. All these builders with their resulting visualisations will be explained in more detail in the following sections.

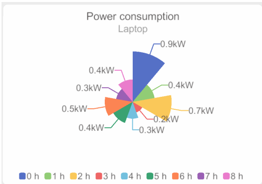
Single visualisations

Single visualisations are the outcome of the builder pipeline after providing it with single data. In ModulARboard, there are seven different builders present in the pipeline with modules for single data. It is possible to have different variants of the same builder in the pipeline by providing it with differing construction variables. For example, there is only one builder class for line charts. However, there are four different instances of the builder present in the pipeline. The first builder has all the default parameters to generate a regular line chart (Figure 4.15c), the second builder generates an area line chart (Figure 4.15d), the third one has parameters for a step line chart (Figure 4.15e), and the last builder generates a step area line chart (Figure 4.15f). By providing the builder with configurable parameters, there is no need to copy and paste entire classes to adjust a single variable. This makes the code reusable and clear.

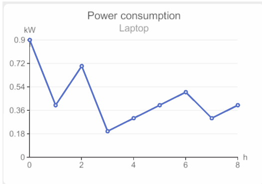
Similarly to the line chart builders, there are two instances of the pie chart builder in the pipeline as well. One builder generates a default pie chart (Figure 4.15a), while the second builder outputs an area pie chart, also called Florence Nightingale's pie chart (Figure 4.15b). The remaining five builder types only have one instance in the builder pipeline, resulting in eleven distinct visualisations for single data. With four different variations of the line chart, two distinct pie charts, one bar chart, one column chart and one scatter chart, there are nine supported visualisations for single continuous data. They are all displayed in Figure 4.15. Contrarily, there is only one builder for discrete and fractional data. There is a number visualisation builder for the discrete data visualisation (Figure 4.16) and a ring chart builder for the fractional data visualisation (Figure 4.17).



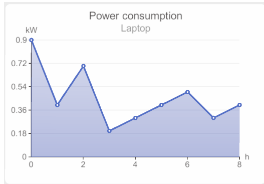
(a) Pie chart



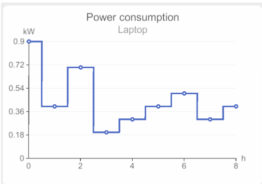
(b) Pie chart



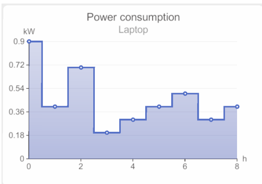
(c) Line chart



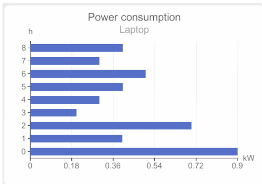
(d) Area line chart



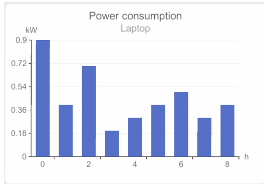
(e) Step line chart



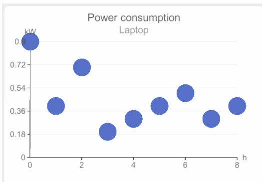
(f) Step area line chart



(g) Bar chart



(h) Column chart



(i) Scatter chart

Figure 4.15: All the available recommended visualisations for a single data (single component & single data option). In this case, the single continuous visualisations are for the laptop. They all visualise the power consumption, which is continuous data.

Figure 4.16: The only available recommended visualisation for single discrete data. It shows the number of times the office light has been turned on.

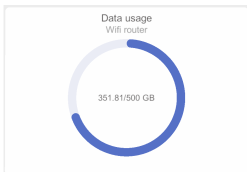
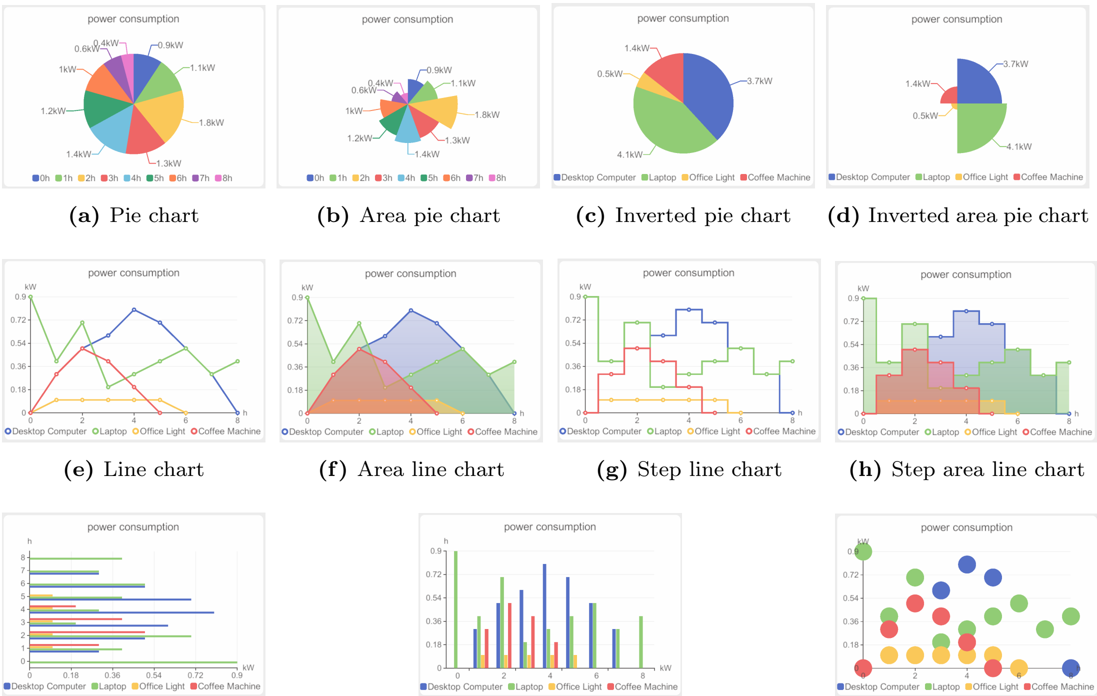


Figure 4.17: The only available recommended visualisation for single fractional data: a ring chart. It shows how many Gigabytes (GB) of data are used compared to the amount of GB available.

Composite visualisations

As explained previously, composite data consists of multiple components grouped together with only one data option (of any supported data type). Composite visualisations are the type of visualisations generated for composite data. Similarly to the single continuous data, there are four instances of the line chart builder (Figures 4.18e, 4.18f, 4.18g & 4.18h), one bar chart builder (Figure 4.18i), one column chart builder (Figure 4.18j) and one scatter chart builder (Figure 4.18k). However, for composite data, there are four different instances available of the pie chart builder. The first two are identical to the single continuous pie charts. For these two charts, the continuous data values (y-values) are summed over every component per label in the series (x-value). This combined data is then visualised with a regular pie chart (Figure 4.18a) and an area pie chart (Figure 4.18b). The other two pie charts, on the other hand, are the result of summing up all the values (y-values) in the series per component. One of them is a regular pie chart (Figure 4.18c), while the other is an area pie chart (Figure 4.18d). This results in eleven distinct visualisations for composite continuous data, as illustrated in Figure 4.18. Composite fractional data, on the other hand, is only supported by the ring chart builder (Figure 4.21).



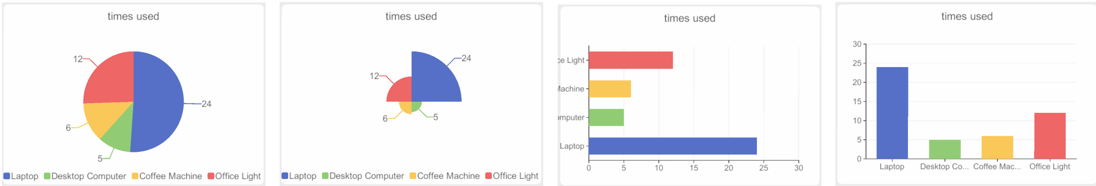
(a) Pie chart **(b)** Area pie chart **(c)** Inverted pie chart **(d)** Inverted area pie chart

(e) Line chart **(f)** Area line chart **(g)** Step line chart **(h)** Step area line chart

(i) Bar chart **(j)** Column chart **(k)** Scatter chart

Figure 4.18: All the available recommended visualisations for composite continuous data (multiple components & single data option). In this case, the composite continuous visualisations are for four different components: desktop computer, laptop, office light and coffee machine. They visualise the power consumption, which is continuous data.

The single discrete data visualisation, as can be seen in Figure 4.16, may seem simple. However, when combining discrete data of multiple components together, it quickly becomes a much more powerful tool. Contrary to the single discrete data, the composite discrete data has four builders in the pipeline, instead of only one. There are again two instances of the pie chart builder: one regular pie chart (Figure 4.19a) and one area pie chart (Figure 4.19b). On top of that, there is one bar chart builder (Figure 4.19c) and one column chart builder (Figure 4.19d) with support for composite discrete data. This results in four distinct visualisations for composite discrete data. Which means that in total, there are sixteen different visualisations available for composite data.

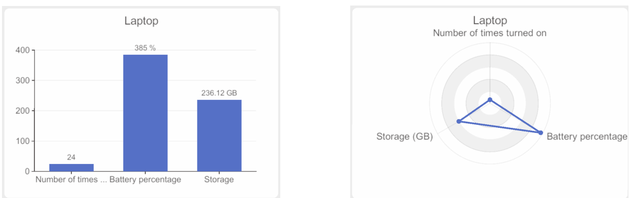


(a) Pie chart **(b)** Area pie chart **(c)** Bar chart **(d)** Column chart

Figure 4.19: All the available recommended visualisations for composite discrete data (multiple components & single data option). In this case the composite discrete visualisations are for four different components: laptop, desktop computer, coffee machine and office light. They visualise the number of times they were used, which is discrete data.

Combined visualisations

Lastly, there are two different types of combined data with their respective visualisations. The combined data visualisations are only for one component, while the composite-combined data visualisations are the extended version for multiple components. The only difference is in the number of components. For the combined visualisations, this results in a single series of data, as illustrated in blue in Figure 4.20. The composite-combined visualisations, on the other hand, have a series for each component, as illustrated in blue, green and yellow in Figure 4.22. However, combined visualisations combine data of any supported data type. This is achieved by the `GetValue()` method provided for every derived data type. This method returns a single value to be used in the visualisation. For discrete and fractional data, it simply returns its value, while continuous data combines all its values by summing them together.



(a) Combined column chart **(b)** Combined radar chart

Figure 4.20: Both recommended visualisations for single-combined data (single component & multiple data options). In this case, there are three selected data options from the laptop: The number of times used, battery percentage and storage.

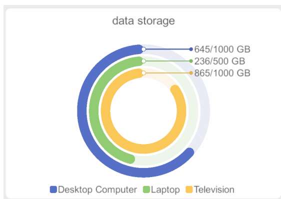
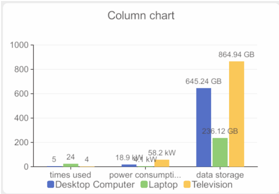


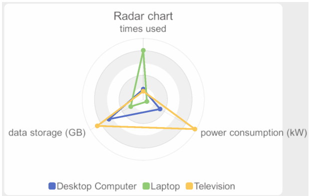
Figure 4.21: The only recommended visualisation for composite fractional data: a ring chart. In this case, three components are shown, each having its own ring: Desktop computer, laptop and television. The data storage is displayed as a function of the maximum amount available.

For single-combined data visualisations, there are two builders with supporting modules. There is a column chart builder and a radar chart builder, which can combine any data type together to form a combined column chart (Figure 4.20a) or a combined radar chart (Figure 4.20b), respectively. It is also possible to add extra modules for multiple data options of the same type, instead of only combining all data types together (resulting in only one module for composite data). This was briefly done as an illustration of a specific use case, as can be seen in Figure 4.23. In this case, a separate module was added to support single-combined data consisting of only continuous data options with identical labels (x-values). Every continuous data option is added as a series in the column chart. However, this builder is not part of the standard builder pipeline from ModulARboard.

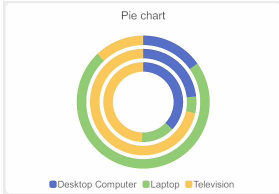
The composite-combined visualisations, on the other hand, have an extra builder in the pipeline, resulting in a total of three builders. The column chart and radar chart builders are identical to the single-combined data builders, except they add a series for every component instead of only



(a) Composite combined column chart



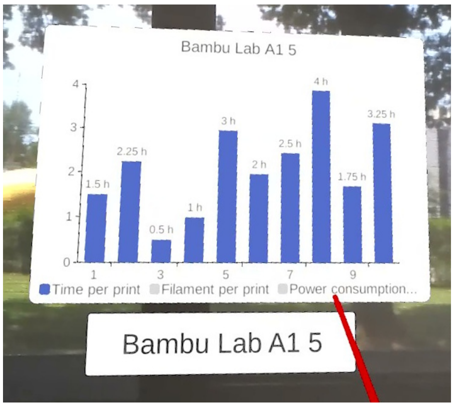
(b) Composite combined radar chart



(c) Composite combined pie chart

Figure 4.22: All the available recommended visualisations for composite-combined data (multiple components & multiple data options of any type). In this case, there are three components (desktop computer, laptop & television) and three data options (times used, power consumption & data storage).

one (Figure 4.22a & 4.22b). The third builder is a combined pie chart builder, which adds a nested doughnut per data type to the visualisation, as illustrated in Figure 4.22c. Unfortunately, the XCharts package (section 4.2) is limited, resulting in the inability to add labels to the different doughnuts in order to indicate the data name. It is, however, possible to request the data name in the tooltip, which becomes available on hover (Figure 4.23b). It is also possible to filter the visualised data series by disabling and enabling them via the legend (Figure 4.23a).



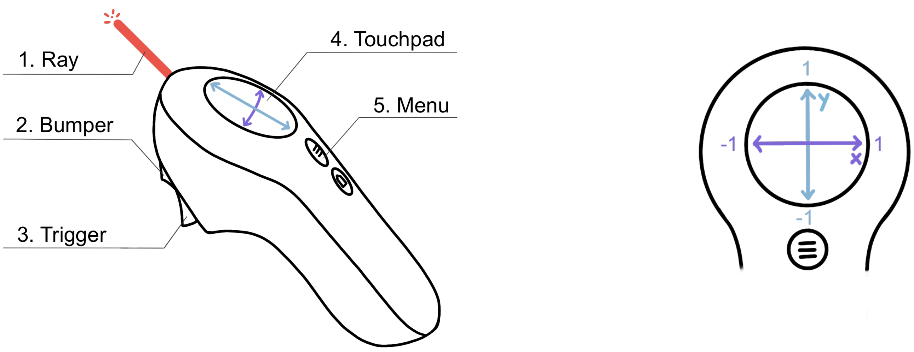
(a) This showcases the ability to filter series. In this screenshot, only the first series (time per print) is enabled. The last two series (filament and power per print) are disabled.

(b) This showcases a tooltip that becomes available when hovering over the visualisation with the raycast.

Figure 4.23: Screenshots of ModulARboard in use in a 3D printer lab. This illustrates an extra custom visualisation added to the pipeline for a demonstration. It combines multiple continuous data options of a single component that have identical labels, resulting in a continuous single-combined visualisation.

4.1.4 Interaction mechanisms

To better understand the anchoring mechanisms in section 4.1.5, it is best to first understand how the interaction with ModulARboard works. The user input is registered through a six-degree-of-freedom (6DoF) controller as part of the Magic Leap 2 HMD. This controller has components to provide input to interact with the application, as illustrated in Figure 4.24a. The controller is augmented by the HMD with a raycast coming from the controller's top. This ray can be used to directly interact with the virtual artefacts in the augmented environment by aiming at them with the controller. Usually, this action is paired with a button press from either the bumper or trigger, which are located at the back of the controller. Furthermore, there is a touchpad located at the top of the controller. This can be used to input certain gestures or simply utilise its Cartesian coordinates, ranging from -1 to 1 (Figure 4.24b). Lastly, there is a menu button which is called the 'App Menu for developers' by the documentation [ML25a], meaning it can be used for anything.



(a) (b)

Figure 4.24: The different components available on the controller to interact with the Magic Leap 2 (ML2) HMD and its applications. The ray follows the orientation and position of the controller, and is used to interact with the virtual artefacts located within the augmented environment displayed through the ML2. The button and trigger are often used in combination with the raycast, while the menu button can be programmed for anything. The touchpad is illustrated in more detail in **(b)**. It features a Cartesian coordinate system ranging from -1 to 1.

In order to provide a clear mental model to the user concerning the interaction with the application, every action is assigned its unique combination of controls, which is consistently used throughout ModulARboard. All the available actions with their respective combination of controls are summarised in Table 4.1. For example, the combination of pointing at something with the ray and pressing the trigger results in the selection of the pointed-at item. This action can be compared to the traditional desktop, where the mouse (ray) points at something and the left button (trigger) is used to click/press/select it. Similarly, the bumper of the controller can be used to request extra information, as is done with the right button on the traditional mouse. Additionally, the bumper on its own is used to confirm the placement of a component. The other controls, however, are less closely tied to a traditional desktop, since they are more intuitive with an HMD.

Action	**Controls**
Select/click	Ray + trigger
'Right-click'	Ray + bumper
Open or reposition dashboard	Ray + menu
Close dashboard	Menu
Confirm anchoring item	Bumper
Cancel anchoring item	Menu
Resize item	Touchpad y-axis
Move item (marker-based & view frustum anchoring)	Trigger + full touchpad
Move item (observer-based & spatial anchoring)	Trigger + ray (+ touchpad)
View tooltip	Hover with ray

Table 4.1: The controller bindings in ModulARboard needed to perform certain actions.

The Magic Leap 2 provides native support for hover interactions as well as grab interactions by adding their class to the items that should be grabbable. The interaction is performed by pointing the ray at something and then pressing the trigger to 'grab' it. It can then be moved around however the user prefers. Depending on the settings of the grabbed item, it can be spun around to the left or right by pressing the left or right side of the touchpad, respectively. It can even be moved closer or further away from the user by pressing the top or the bottom of the touchpad, respectively. However, these two options depend on the settings of the grabbed item and might be disabled when not needed. This grab functionality is added to the top part of all the menus (A, B & C) in order to place them wherever the user prefers.

Besides, this is not the only way to grab items in ModulARboard, since some components are not allowed to be grabbed freely. They are tied to a certain orientation and z-position, meaning that only their x- and y-values can be altered. In this scenario, the item is already highlighted to indicate it can be moved around. Hence, it is no longer needed to point at the item to grab it. The user can simply press the trigger to grab it and start moving it around. However, the controls of the touchpad have a different meaning in this context. The y-axis of the touchpad is used to move the item up and down, while the x-axis of the touchpad is used to move the item from left to right. While both repositioning actions are not entirely the same, they remain similar in order to preserve the user's mental model. In line with the alternative grab action, it is possible to resize a highlighted component by only using the y-axis of the touchpad without pressing any other buttons. The positive values at the top of the touchpad scale the component up, while the negative values at the bottom of the touchpad scale the component down.

The last controls are chosen in a way to fit with the rest of the controls available in ModulARboard. This way, the menu button is programmed to open, reposition and close the dashboard, as well as to cancel the anchoring of a component. When the dashboard is closed, it can be opened by pressing the menu button on the controller. Conversely, when the dashboard is opened (and looked at), it can be closed by pressing the menu button again. Hence, the menu button serves as a toggle button for the dashboard's visibility. On top of that, if the component is 'toggled on', but it is not in sight by the HMD, it can be repositioned in front of the user again by pressing the menu button. It acts similarly to opening the dashboard, since the user could not see the dashboard, which makes it feel as if the dashboard was opened instead of repositioned. When opening or repositioning, the dashboard is placed 2 meters in front of the controller at the position it was pointing at when the menu button was pressed. Besides, as mentioned before, the menu can always be moved around by performing a grab interaction. This gives the user more freedom in deciding where to put the dashboard when working with it.

4.1.5 Anchoring mechanisms

The anchoring mechanisms are responsible for the positioning of the labels and visualisations within the augmented environment. They form the basis of ModulARboard along with the adaptive visualisation recommendation system. As mentioned before in section 4.1, the anchoring menu can be accessed in various stages throughout the flow of ModulARboard. It is possible to anchor ③ a label or ④ a group's label directly to the environment. But it is also possible to access the anchoring menu through ⑧ one of the visualisation menus B or C. From this anchoring menu, it's possible to select one of the four provided anchoring mechanisms: marker-based, view frustum, observer-relative or spatial anchoring. Depending on the selected anchoring mechanism, the positioning mechanisms will be slightly different, as will be explained in the following sections. Once the label or visualisation is in the desired location and size, the anchoring can be confirmed by pressing the bumper on the controller, as was explained in section 4.1.4. The item can now be accessed again by 'right-clicking' it with the bumper to open the editing pop-up menu. If the item is a label, the user can ②/⑤ choose to visualise it. The other options in the menu are available regardless of the anchored item. Both labels and visualisations can be moved and rescaled again by selecting ⑥ the 'move' option. This triggers the same process as the initial anchoring process. Lastly, all components can be removed from the scene by choosing ⑦ the remove option in the pop-up menu.

While most of the anchoring mechanisms discussed in section 3.7 are implemented in ModulARboard, it was decided not to implement object-relative anchoring, since it resembles marker-based anchoring a lot. The only difference between the two is that marker-based anchoring utilises a 2D marker for image recognition, while object-relative anchoring requires more visual analysis to recognise 3D objects. This would require more extensive research and advanced algorithms in order to perform image recognition. In this case, the cost of supporting object-relative anchoring outweighed the benefits. Since ModulARboard is a proof-of-concept application, it suffices to test the functionality with 2D markers. There is no difference for the user except for looks. Instead of anchoring something to a 3D object, it is now anchored to a 2D marker. In the end, when the application is deployed, the functionality can always be extended to also support object-relative anchoring for a more professional feel.

Marker-based anchoring

Marker-based anchoring, as previously explained, anchors a virtual artefact relative to the position of a fiducial marker, which is a 2-dimensional visual pattern which is easily recognised, like QR codes or EAN codes, also known as barcodes. The Magic Leap 2 currently has support for four different types of fiducial markers: QR, ArUco, EAN_13 and UPC_A. However, the last two are still experimental. Since the article explaining how to implement marker tracking in augmented reality for the ML2 utilises ArUco markers [UNT], it was decided to work with ArUco markers as well. Nonetheless, this can be changed very easily in settings by simply altering a variable.

The anchoring process of marker-based anchoring, however, is as follows. When marker-based anchoring is selected in the anchoring menu, ModulARboard starts looking for all the visible ArUco markers and highlights them green when found (Figure 4.25a). The user can then see which markers are recognised by the application and are available for anchoring. A marker can be selected by hovering over it so it becomes yellow (Figure 4.25b) and pressing the trigger. This positions the item on top of the marker with an orange highlight, indicating that it is in edit mode and can be moved around (Figure 4.25c). With marker-based anchoring, the items are tied to the location and orientation of the ArUco markers, meaning that when the marker is angled or moved, the items tied to it will move accordingly. As a result, only the x- and y-values can be altered relative to the position of the marker. Because of this, the second moving mechanism, as described in section 4.1.4 (Table 4.1), is available for marker-based anchoring (Figure 4.25d). The scaling is identical for all anchoring mechanisms. Once the user is satisfied with the position of the item relative to the marker, they can confirm the placement by pressing the bumper, which removes the orange highlight (Figure 4.25e). The relative position of the item can always be altered by choosing to move it in the editing pop-up menu.

View frustum anchoring

As explained in section 3.7, this type of anchoring positions virtual elements within the view frustum of the observer. In this case, the observer is the user wearing the ML2 HMD. In ModulARboard, view frustum anchoring is implemented by having a Unity Canvas at 1 meter distance from the main camera (user). When this type of anchoring is chosen from the menu, the item is positioned in the default location in the middle of the canvas/view frustum. From here, it can be moved around in the same way as with marker-based anchoring, by pressing the trigger and swiping the touchpad. This is because the position and orientation of the item are defined by the view frustum, meaning only the x- and y-values can be altered relative to the canvas. Once the user is satisfied with the positioning, it can be confirmed by pressing the bumper on the controller.

Observer-relative anchoring

Observer-relative anchoring, as described in section 3.7, follows the position and orientation of the observer. In order to do this, the position and orientation of the observer must be tracked independently of the orientation of the view frustum, as illustrated in Figure 4.26. However, the

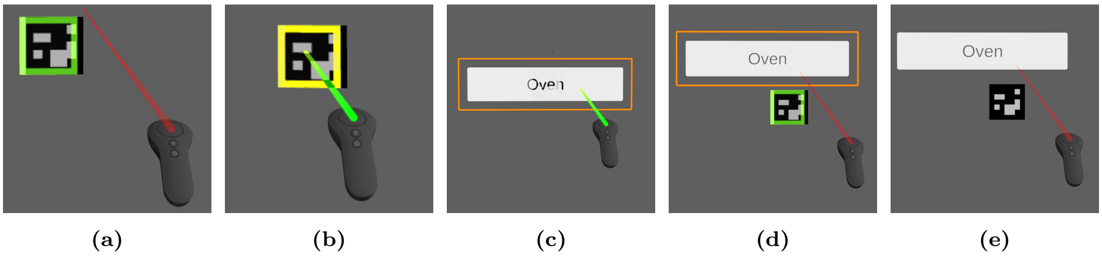


Figure 4.25: Marker-based anchoring mechanism process. Screenshots taken in the Unity Application Simulator of Magic Leap.
(a) Marker anchoring got selected to position an item. All markers get highlighted with green borders to indicate availability.
(b) When hovering over a marker, it becomes highlighted in yellow.
(c) When pressing the trigger while hovering over a marker, it places the item on the marker.
(d) From here, the item can be moved around by pressing the trigger and the touchpad at the same time. It can also be scaled up or down by pressing up or down on the touchpad.
(e) The placement can be confirmed by pressing the bumper. The orange border disappears, and the item is confirmed.

ML2 system consists solely of the HMD and the 6DoF controller. There is no separate sensor available to track the body of the user, independent of the head movements. This means that the position and orientation of the observer coincide with the orientation of the view frustum. As a result, observer-relative anchoring in ModulARboard is an approximation of the observer-relative anchoring as explained in section 3.7. This is achieved by letting the item follow head rotations around the y-axis, while ignoring rotations around the x- and z-axis. In other words, the item will follow the head rotations when the user looks to the left or to the right, but will remain in the same spot when the user looks up or down, or tilts their head sideways. Technically, this is achieved in two parts. The first part calculates the displacement of the item relative to the position of the user and stores these values. This happens every time the item is released after moving it into the desired position. The second part constantly updates the position of the anchored item relative to the position of the user, using the stored displacement values. This process is illustrated in Figure 4.27.

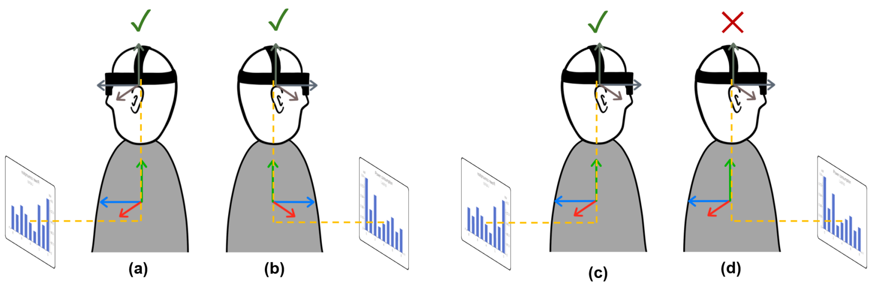


Figure 4.26: This figure illustrates how observer-relative anchoring should behave. The item is located in front of the user, so when they **(a)** are turned to the left with both body and head, the item will stay in front of the user. When the user **(b)** turns to the right with both their head and body, the item will still remain in front. On the other hand, when the user **(c)** turns their head to the other side from where the body is facing, the item will remain in front of the body, but the user will not be able to perceive it. However, the ML2 cannot track body orientation independently from the head orientation. Because of this, observer-tracking cannot be fully implemented and will be approximated instead. This results in scenario **(d)**, where the item remains in front of the user following the orientation of their head, even though their body is oriented in the opposite direction.

When the user releases the item after moving it, the displacement vector and the angle (θ) between the item and the forward vector are calculated in world space. The forward vector has its origin at the main camera's position (user's head) and points forward in the direction the camera (user) is looking. The displacement vector has the same origin and points towards the item anchored relative to the observer. However, in order to keep the item at the same height (y-axis), even when looking up and then, the displacement vector is divided into two parts. There is the offset, which is the y-component of the 3-dimensional displacement vector. And there is the projected displacement vector onto the xz-plane, to get rid of the y-value. The projected displacement vector is used to calculate the xz-position relative to the user by multiplying its magnitude by the normalised projected forward vector. The full displacement is then restored by adding the y-offset to the xz-position, which is then rotated by the stored angle around the position of the user. This keeps the item in the right spot at all times.

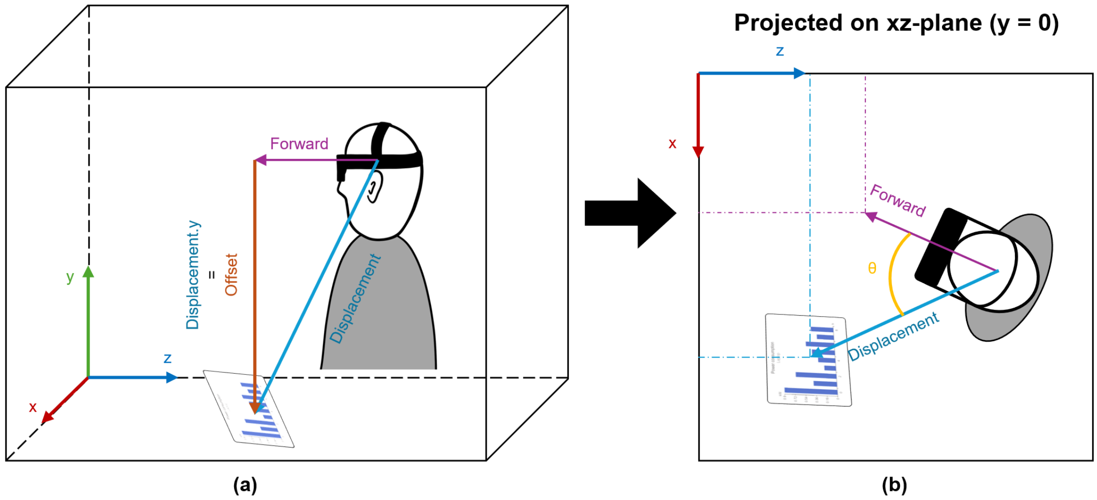


Figure 4.27: This schema illustrates the various parameters used to anchor the item relative to the observer's location and orientation. **(a)** illustrates this in world space, while **(b)** illustrates it from a top-down view (projected onto the xz-plane). The item is positioned in the xz-plane by multiplying the projected forward vector by the magnitude of the projected (onto the xz-plane) displacement vector. This position is then rotated by the angle θ around the user. Lastly, in order to keep the item at the same y-position relative to the user's head, the y-offset (displacement.y) is added to the rotated projected displacement vector to become the full displacement of the item.

In summary, when the user selects observer-relative anchoring in the anchoring menu, the item is positioned right in front of the user. They can then grab the item with the raycast in combination with the trigger and move it to the desired location. Every time the trigger is released, the displacement vector and angle are recalculated. When the user is satisfied with the positioning, it can be confirmed by pressing the bumper. The position is then updated and faced towards to user at all times to remain in the anchored position relative to the observer.

Spatial anchoring

The last implemented anchoring mechanism is spatial anchoring. This mechanism anchors items to the augmented environment in world space. The world space is determined by the 'session map' from the Magic Leap 2 [ML25d]. This is a temporary spatial map created every time the device is powered on, which exists until the device is powered off again. The origin of this session map is determined by the physical location of the ML2 when it was powered on. The Magic Leap 2 head tracker runs continuously in order to render the augmented content relative to the user's physical world when moving around. This ensures a seamless integration of the real world with the virtual augmented artefacts.

When the user selects spatial anchoring in the anchoring pop-up menu, the item is spawned in the default position in front of the user. The item can be rescaled by using the touchpad, as explained in section 4.1.4. For this anchoring mechanism, the user has to move the item by pointing at it with the ray and then 'grab' it by pressing the trigger. It can then be moved closer or further away by pressing down or up on the controller's touchpad. The user can repeat this process as many times as needed. They can even step away or walk around the item to take in the bigger picture. Because of the script present on the item with spatial anchoring, the item always faces the user. This makes sure the user can read the item at all times. If the user is satisfied with the positioning, it can be confirmed by pressing the bumper. The item is now anchored to the session map as long as the device is powered on.

Summary

In summary, when the user selects an anchoring mechanism in the anchoring pop-up menu, the according anchoring process is initiated. The specifics depend on the selected anchoring mechanism. However, the general process is similar across all anchoring mechanisms. The item is spawned in its default location with an orange border around it, indicating that it can be altered. The item can be resized with the touchpad or moved around by combining the trigger and touchpad, as explained in section 4.1.4. The user can fully test out the positioning and orientation during the anchoring process, since the anchoring is already active during this process. The repositioning, resizing and examination can be repeated as many times as needed. Once the user is satisfied, the anchoring can be confirmed by pressing the bumper. The orange border disappears, and the item is officially anchored. However, the item can always be moved or removed by consulting its pop-up menu with the bumper. This process can be repeated for any label, group label or visualisation available in ModulARboard, enabling the user to fully customise the AR dashboard to their heart's desire. Figure 4.28 and Figure 4.29 illustrate ModulARboard in use and the customisation it has to offer.

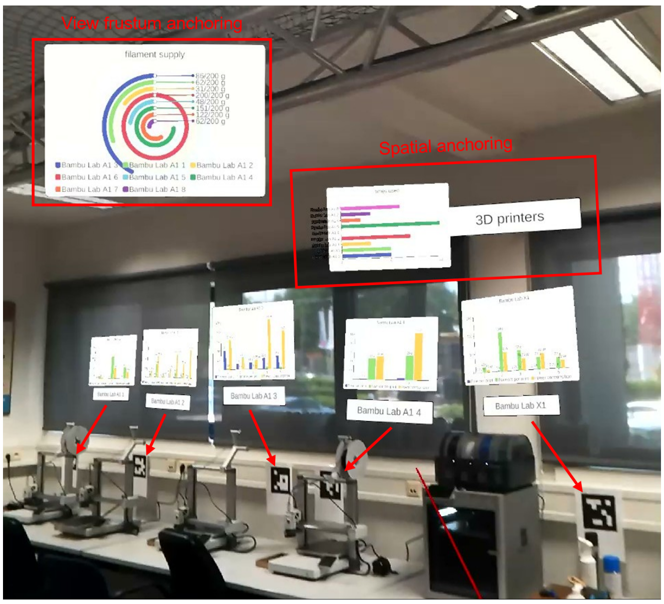


Figure 4.28: This is a screenshot of ModulARboard in a real-life scenario. It illustrates three different anchoring mechanisms. First, there is marker-based anchoring as indicated by the red arrows. The labels and the visualisations are anchored relative to the ArUco markers. Secondly, an overview of the filament supply of all the Bambu A1 printers is anchored to the view frustum at the top left. Lastly, all printers (including a Bambu X1 printer) are grouped by a label and anchored to space in the middle of the wall. This allowed for a composite visualisation to be made that visualises the number of times used per printer, as seen anchored next to the group label.

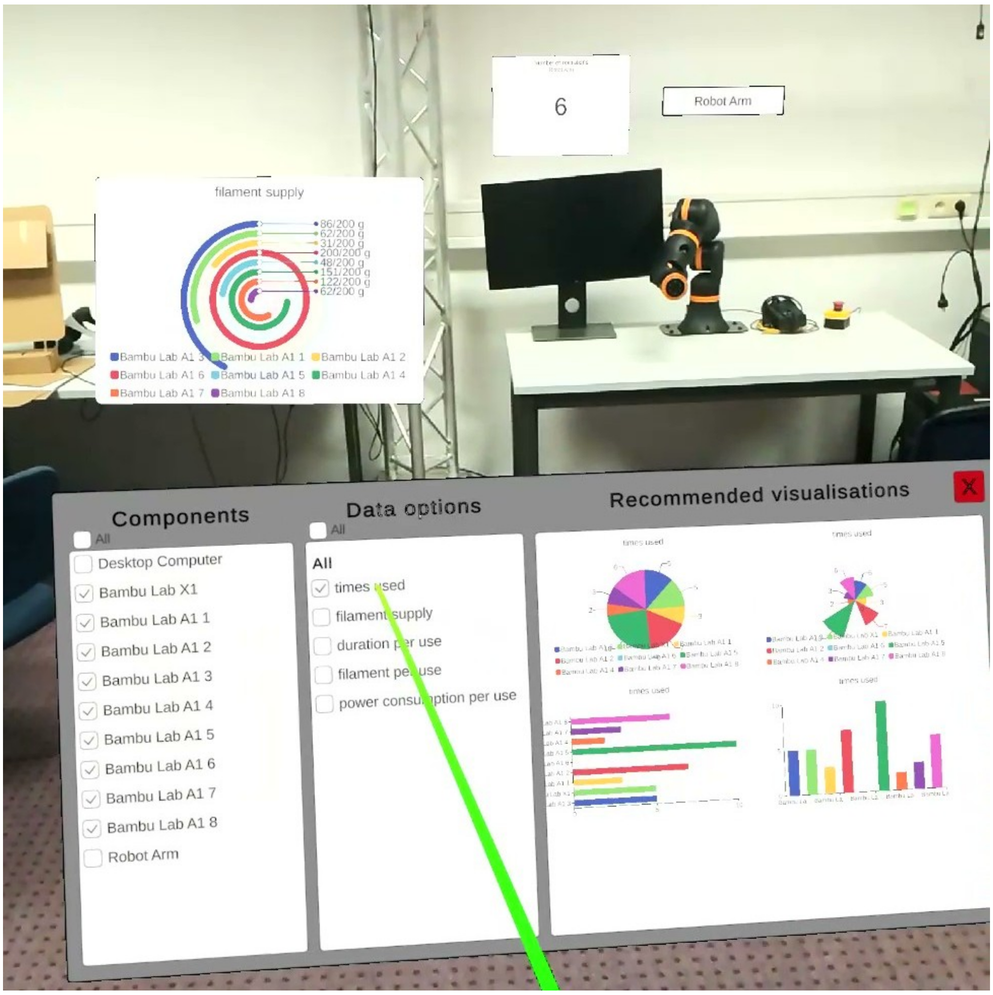


Figure 4.29: This is a screenshot of ModulARboard in a real-life scenario. It illustrates the C composite visualisation menu with all 3D printers selected and the times used data option in the middle panel, resulting in composite visualisation on the right. There is also a composite ring chart visualisation anchored to the view frustum at the top left. Lastly, it illustrates a label and a single number visualisation anchored to space next to the robot arm.

4.2 Technologies

ModulARboard was developed for the Magic Leap 2 (Figure 4.30), a head-mounted Augmented Reality headset using optical see-through technology. With a graphics display of 1440 by 1760 pixels per eye (stereoscopic display), it has a field of view (FOV) of 45 degrees horizontally by 55 degrees vertically and 70 degrees diagonally [ML25b]. The headset is equipped with a six-degree-of-freedom controller, and supports hand gestures, eye tracking and voice commands [ML]. The Magic Leap 2 is built on top of the Android Open Source Project, Android 10 API Level 29, which means that ModulARboard was developed for Android.

Figure 4.30: The Magic Leap 2 headset with designated 6DoF controller. (Image from [SYN25])

There are multiple environments available for Magic Leap application development. However, since the facility has the most experience working with Unity, it is the preferred development environment. An attempt was then made to implement the Magic Leap OpenXR Unity SDK, seeing that the Unity Magic Leap SDK (MLSDK) is deprecated. Despite that, the environment would not work with OpenXR because it is not fully implemented yet, resulting in not all features being available [ML25c]. Hence, Unity was set up with the MLSDK regardless, resulting in a fully functional development environment.

ModulARboard uses several packages for core functionalities, including mainly XCharts[1], which forms the basis for the 2D visualisations throughout the application. It is a Unity charting and data visualisation library which offers a wide variety of visualisations with plenty of customisation options. Furthermore, the Newtonsoft JSON Unity Package[2] provides utilities to convert JSON files to class objects. It is used to convert the input JSON file with the dashboard and component data to C# class objects to use throughout the application. Lastly, the Magic Leap XR Keyboard[3] is used to implement a floating visual keyboard which allows for keyboard input in AR environments using solely controller input or hand gestures. Lastly, the data used to power ModulARboard was synthetically generated by OpenAI's ChatGPT (May 24 version)[4] based on user-defined parameters. While the icons in the A main dashboard (Figure 4.3) are downloaded from Flaticon[5].

[1] `https://xcharts-team.github.io/en/`
[2] `https://docs.unity3d.com/Packages/com.unity.nuget.newtonsoft-json@3.0/manual/index.html`
[3] `https://github.com/magicleap/MagicLeapXRKeyboard?path=/Packages/MagicLeapXRKeyboard`
[4] `https://chat.openai.com/`
[5] `https://www.flaticon.com/free-icons`

Chapter 5

Use cases

The following use cases illustrate ModulARboard in the context of a deployed application in order to highlight its versatility and to paint a picture of the possible future applications it has to offer. In these use cases, ModulARboard is no longer a proof-of-concept application, meaning that it can now store the modular dashboard in memory. This way, the user does not need to rebuild the dashboard every time the AR HMD is powered on again. On top of that, the third use case also assumes that the deployed version of ModulARboard supports live data streams as well, in order to perform real-time monitoring. These are two minor additions in order to fully deploy ModulARboard. However, since ModulARboard is still a proof-of-concept, the focus was more on exploring different ways to reimagine the traditional dashboard, instead of providing an application that is ready to be deployed immediately.

In total, there are four different use cases, illustrating the wide range of possibilities that ModulARboard has to offer. The first use case is about John, a homeowner and tech enthusiast who is interested in monitoring his smart devices. The second use case describes Jane and her office. She owns a startup business and wishes to monitor her employees and her office in general. The third use case illustrates Dave, a technician for a large-scale factory. He is responsible for monitoring all the equipment on a day-to-day basis. The last use case, on the other hand, is more focused on the possibilities that the anchoring mechanisms have to offer and how they can be employed to facilitate collaboration. It illustrates Ann and Marie gathered in a meeting room together with other people to have a sticky-note brainstorm session.

5.1 Use case 1: Average (smart) home usage

John is a true tech enthusiast, always up to date with the latest gadgets. He finds out about ModulARboard and cannot wait to try it out in his own home decked out with smart plugs and appliances. He collects the data from all his devices and combines it in the supported JSON format. He is very happy because ModulARboard already supports all the data types he needs. He even prints out several ArUco markers and pastes them everywhere around the house. He then puts on the Magic Leap 2 and starts up ModulARboard. The JSON is automatically converted into components, and all the components get assigned their designated icon. John clicks on the controller's menu button, and the dashboard opens in front of him. He scrolls a bit through the dashboard to see all his appliances displayed. He immediately opens the pop-up menu of the television and selects to visualise it, eager to find out what visualisations are available. The visualisation menu for the television opens with two available data options: power consumption (continuous data) and data storage (fractional data). He selects the data storage option, and a ring chart is presented, showing that he still has 5 GB of storage left. He immediately wants to place this visualisation next to his TV, so he uses his bumper to open the pop-up menu and selects 'anchor to marker', since he has already put a marker above his television. He selects the green highlighted marker above his TV and scales it a bit down with the controller's touchpad. He does not move the visualisation, since he wants it to cover the marker. Then he confirms the placement by pressing the bumper, which makes the orange border disappear.

John previously noticed that apart from directly visualising components, they can also be anchored to the environment themselves. He reopens the dashboard, and this time he opens the pop-up menu for the electric drill. He chooses 'spatial anchoring' in order to place it with his device. A label with an orange border and 'Electric drill' on it appears right in front of him. He intuitively grabs the label by pointing at it and pressing the trigger. While grabbing it, he walks over to his garage where the drill is and puts the label on top of it. He confirms the placement and walks around it to find the label turning along, constantly facing him. He is now excited to use this functionality to label everything in his garage. He knows that the bumper is similar to 'right-clicking' on a traditional mouse, so he tries it on the recently placed label. The pop-up menu opens, and he sees that he can visualise the drill. John is curious to know what visualisations are available for the drill and chooses this option. The visualisation menu opens for the electric drill with two available data options: power consumption and the number of times used. He selects the power consumption data options and scrolls through the recommended visualisations. He selects a column chart for the power consumption and anchors it to the space again. He is, however, not really interested in seeing the power consumption for his electric drill, so he removes the visualisation again and reopens the visualisation menu to select another visualisation. This time, he selects 'times used' instead, and a number visualisation is recommended. He chooses to anchor this visualisation next to the label because he would like to know how much he actually uses the drill. He also finds the label very helpful, since he can request the component's visualisation menu directly, instead of having to search for the component through the dashboard. Especially because he is a versatile person and often likes to change things up a little.

Then, John opens the dashboard again by pressing the menu button on the controller. He sees that all the components have checkboxes and that there is a group button at the bottom of the dashboard. He tries to click the group button with the trigger to see what it would do, but this does nothing. He then selects two components and sees the group button becoming available. He then clicks the group button again, which this time, drops down the four anchoring mechanisms. He selects 'anchor to observer' and the empty label spawns in front of him. It says in a placeholder that a name can be entered once the placement is confirmed. When he moves around a little, he notices that the label follows him. He confirms the placement and clicks the label, which opens up a floating keyboard. However, he does not really need a group and was just testing it out. So he uses the bumper on the label and removes it from the scene.

Lastly, John reopens the dashboard and then clicks the visualise button at the bottom left of the dashboard, which opens the composite visualisation menu. He selects a few components and sees the data options update immediately. He then selects a few data options, and the visualisations load in immediately again. John finds it satisfying how fast the system responds to his requests. He then deselects all the components by pressing the 'All' checkbox at the top of the panel. He is actually looking for a visualisation that allows him to monitor both his television data storage and the remaining data in his internet subscription. Because he is always exceeding the limit of his data subscription, which means he has to pay extra. On top of that, he often forgets to delete recorded episodes from his TV, causing his storage to become full. As a result, his favourite TV show is occasionally not recorded. Hence, John selects the television and the Wi-Fi in the first panel of the composite visualisation menu. The adaptive data loading automatically combines the data options of both components and displays them in the next panel. The television is the only component with the power consumption data option. However, both components have a 'data usage' data option available under the 'All' title. John selects this data option, and the composite ring chart is recommended as a composite fractional data visualisation. He is very excited about this visualisation and since it is so important, he chooses to anchor it to his view frustum. This way, he can continuously monitor the data usage of both his television and Wi-Fi router.

5.2 Use case 2: Office monitoring & optimisation

Jane is a small-scale IT business owner. She likes to follow up on her employees by keeping track of a few statistics, like when they clock in and out, how often they put their computer to sleep to take a break and how long this takes, and how many tasks they complete. She already has a traditional dashboard that she consults when she walks around the office. However, she finds it inconvenient to walk around the office and search through the dashboard in order to find the right employee. On top of that, not all the visualisations she desires are supported, and it is not possible to extend its functionality. She is thus looking for a dashboard system that allows her to walk around and check statistics simultaneously. While also allowing her to extend the core functionality by providing her own data types and visualisations. Because of this, she replaces the old dashboard with ModulARboard to fulfil her needs.

She first converts her data to the supported JSON format, but quickly notices that some of her data types are not supported by the main functionality of ModulARboard. However, because she studied something IT-related herself and ModulARboard is well structured, she decides to add the support herself. Jane does this by adding her own data types by deriving them from the provided base class and adding them to the JSON converter. Now that all the required data types are supported, she uploads the JSON file to the Magic Leap 2 and starts up ModulARboard. She then opens the dashboard by clicking the menu button on the controller and scrolls through the dashboard to see that all her components have loaded in correctly. She selects one of her employees on the dashboard and chooses to visualise their data directly, opening the visualisation menu. However, when she tries to request the recommended visualisations for one of her custom data type options, she gets a feedback message telling her that there are no recommended visualisations available. She realises that she forgot to add her own visualisation builders with modules for her custom data types, and immediately goes to fix this. Now that Jane provided both custom data types and visualisation builders with support for those data types, she starts up ModulARboard again. This time, when she opens up the visualisation menu for one of her employees and selects one of her custom data type options, she does get her recommended visualisations. Meaning that she can now visualise any data from any employee.

Eager to test it out, she hands out a unique ArUco marker to every one of her employees. She then uses ModulARboard to anchor her custom visualisations to every employee's marker with their respective data. She uses markers as the anchoring mechanism, since the employees do not have a designated spot, meaning that the office can have a different set-up every day. The markers ensure that the employee's data is always correctly matched with the right employee, fully supporting the dynamic office arrangement. Now, when she walks around the office, she can monitor all her employees by simply locating their markers. ModulARboard then automatically displays the visualisations as previously configured, relative to the marker. When she sees that someone is taking too many or too long breaks, she can immediately discuss this with the employee at hand. Similarly, she can congratulate someone for working hard and completing a lot of tasks.

Later, when the new coffee machine finally arrives, she has an idea to add the machine to ModulARboard as well. She initially wants to offer all the different types of coffee that are available in order to give her employees the chance to try them all out. She then wants to narrow them down by only providing the most popular coffees in the coffee machine. So her goal is to keep track of the number of times a certain type of coffee was taken from the coffee machine in order to find the most popular coffee flavours. To do this, she adds a counter for every type of coffee. Once a week has passed, she collects the qualitative data and adds it to the JSON file as well. With ModulARboard open on the HMD, she walks over to her coffee machine and opens the composite visualisation menu. She selects all the different types of coffee and selects their counter data option. This combines all the different types of coffee into one visualisation. Satisfied with the visualisation, she chooses to anchor it to space, since the machine is not moving from its location, unlike the employees. Now, when she makes her daily tour around the office, she can check up on her employees and keep track of the most popular coffees. Eventually, when she decides on the most popular coffees, she can change the visualisation to a composite ring chart to keep track of the inventory of the coffee pads.

5.3 Use case 3: Large-scale factory management

Dave is the head technician for a very large factory/manufacturer and is responsible for monitoring all of its machines, making sure they are fully operational at all times. Since the factory is so large, it is divided into multiple halls, each equipped with designated machines for a specific step in the manufacturing process. Dave really likes his job, but often finds it difficult to keep an overview of all the various machines distributed throughout the manufacturing halls. Because of this, he asked the IT department to connect all the machines to ModulARboard via live data streams. This way, he can monitor all his devices in real-time.

Now that all the machines directly stream their data to ModulARboard, Dave begins to compose his modular AR dashboard. He opens up the main dashboard in front of him by pressing the menu button on the designated controller. He scrolls through the menu and finds all the machines available. He starts by anchoring single visualisations to the most important machines. He does this with spatial anchoring instead of marker-based anchoring, since all the machines have their designated location and are permanently positioned there. Meaning that Dave does not have to print out a marker for every machine, which is faster and also better for the environment. He also wants to have a few combined visualisations, which he can place at the beginning of every hall. This way, he can take a look at the summarising visualisation of a hall, instead of having to go through it completely. He starts by opening the dashboard and selecting all the components of the first hall. He then selects the group button and chooses spatial anchoring to place its label at the beginning of the first hall. He then clicks the label to give it the name of the first hall. Now that the label is anchored, he chooses to visualise the group, which opens the composite visualisation dashboard with all the components of the group already selected. He checks a number of combinations of different data options and eventually anchors some composite and composite-combined visualisations next to the group label of the hall. He then repeats this process for all the different halls in the factory.

Dave is very satisfied with ModulARboard and how much it has already improved the efficiency of his daily tasks. However, he thinks that it can be improved even more. Because of this, he goes to the IT department and requests to extend ModulARboard's functionality. He heard this should not be too hard, since it was designed to be an extendable framework provided with derivable base classes. He particularly asks to add a visualisation that allows him to see alerts when something happens with a certain machine. On top of that, he requests a visualisation of the entire work floor, including all the machines. He wants the machines to be colour-coded according to their operational status. This way, he can immediately tell which device is malfunctioning. Lastly, he wants extra visualisations customised per hall according to their machines, since every hall has its own specific parameters to monitor.

Excited to see the new visualisations from the IT department, Dave starts up ModulARboard and goes to the first hall. He then opens the composite visualisation menu via the hall's group label, and all the components in the hall are selected by default with their respective data options. He again checks different combinations of data options in order to explore the next recommended visualisations added by the IT department. He then removes the old visualisations from the scene and replaces them with new visualisations from the composite visualisation menu. Dave is really glad that he does not need to select all the components of the hall again in order to renew the visualisations and goes on his way to repeat this process for the remaining halls.

Lastly, he opens the dashboard to retrieve the other two visualisations he requested and immediately sees them at the top of the dashboard. He uses his bumper on the first component and opens the visualisation menu. Here, he selects the data option that allows him to get live alerts about malfunctioning machines. He chooses to anchor it to his view frustum, since it is extremely important. He needs to be able to immediately see when something goes wrong. Secondly, he selects the second component and opens the visualisation menu again. Here, he selects the data option that allows him to see the ground plan of the entire factory with colour-coded machines according to their status. Dave chooses to anchor this visualisation to himself at hip-level with observer-relative anchoring. This combination of visualisations allows him to see alerts at all times (view frustum) and then look down at the map to see which machine is malfunctioning in particular.

5.4 Use case 4: Meeting room of the future

Ann and Marie are in a meeting room along with 4 other people. They have to come up with a solution for the parking problem at their office and are organising a sticky-note brainstorm session. They are all sitting around a table wearing HMDs that are connected to the same session. Ann is the meeting room leader and starts by putting up a virtual sticky note with the question “How can we resolve the office parking problem?”. They now all have two minutes to come up with as many solutions as they can think of. Each person writes down their ideas by creating a virtual sticky note and typing in their answer. This can be done by either having a virtual keyboard or a physical keyboard connected to the HMD. When the two minutes are over, everyone has various sticky notes in front of them with one idea per sticky note. They are anchored to the observer for as long as the idea has not been processed.

Now that everyone has a few ideas in front of them, Marie starts by explaining her idea. When doing this, the sticky note with her idea gets displayed in front of all the other people as well by anchoring it to the view frustum or to the observer. The first sticky note with an idea is then positioned somewhere in the room through spatial anchoring. Now that the sticky note is processed, it disappears from the sticky notes anchored to the observer. Then Ann asks if anyone has a similar idea. If so, they get grouped together by anchoring them next to each other. Marie then explains her next idea, and the sticky note is again displayed in front of the other people participating in the brainstorming session. This idea is different from her previous idea, and because of that, it is anchored in another location. Ann asks again if anyone has similar ideas to anchor alongside Marie’s sticky note. This process is repeated by going over every sticky note from every person in the session. When all the sticky notes are spatially anchored, the groups can be reevaluated by moving around the sticky notes to form new clusters. It is also possible to save certain setups. This way, many different groupings can be explored while still being able to easily go back to a previous setup.

Once every idea has been processed and all the sticky notes are spatially anchored in groups to the environment, it is time to evaluate all the ideas. Ann goes to the first cluster of sticky notes and reads out an idea. The sticky note appears in front of each person again, now accompanied by two buttons for voting. When they like the idea, they vote ‘IN’ by pressing the green button. Otherwise, they vote ‘OUT’ by pressing the red button. Everyone votes for the idea by pressing one of the two buttons. When everyone votes ‘IN’, the sticky note turns green. On the other hand, when nobody likes it, the sticky note turns red. When the feelings are mixed, the sticky note turns blue. This process is repeated for every sticky note in order to find the best ideas.

This AR version of a sticky note brainstorming session allows the user to position the sticky notes anywhere, instead of being limited to smooth surfaces. It is also possible to utilise marker-based or object-relative anchoring by tracking the position of physical sticky notes. However, this limits the possibilities for saving setups. With the full virtual sticky note brainstorming session, it is possible to save different setups of sticky notes and reload them. On top of that, it would be possible to request extra statistics, like the size of clusters, what idea is the most popular (by voting) and from whom the most popular idea was.

Chapter 6

Discussion

This work completely focused on Augmented Reality (AR) and Immersive Analytics (IA). More specifically, what IA and AR are exactly, and how AR can be leveraged in order to reimagine traditional data dashboards. In order to do this, it studied what techniques could be used in order to position virtual visual elements in the physical environment. Moreover, it explored how customisation could be introduced. This resulted in a proof-of-concept application called ModulARboard, designed for data analysis in AR and the demonstration of the various anchoring mechanisms and customisation possibilities. In general, an extensive literature review was essential to ground the work and yielded several key insights. On top of that, the development and testing process of ModulARboard resulted in additional findings. These will be discussed in the following sections, along with the current scope and possible future directions.

6.1 Research findings

The work in this thesis consists of two main parts: an extensive literature study and the proof-of-concept application ModulARboard. The literature study first focused on the definitions of Immersive Analytics (section 2.1) and Augmented Reality (section 2.2) in order to get a better perspective on the subject. It discussed the definitions of Milgram and Kishino [MK94] and Azuma [Azu97] concerning AR in order to fully synthesise what should qualify as AR and what should not. This resulted in a combination of the two definitions, stating that Augmented Reality refers to the augmentation of an otherwise 'real' environment by means of virtual artefacts. These computer-generated artefacts not only augment sight, but can be used to augment the other senses like hearing, smell, touch, or even taste. On top of that, they no longer need to be registered in 3D, in order to provide more inclusion towards the other senses. However, another criterion was added in return to limit the allowed distance between the sampling and resynthesisation process. This distance now determines whether the sampled subject is 'real' or 'virtual'. This results in a less strict definition than Azuma's, yet more inclusive than Milgram's definition. In conclusion, this combined definition answers the first subquestion of the research question, concerning what should qualify as Augmented Reality.

Secondly, the literature study focused on various existing AR systems and studies in order to form a design space (chapter 3). This design space consists of seven different categories containing 22 subcategories in total. They were devised by studying many related works and summarising aspects (categories) in which they were similar or differed (subcategories). For example, all AR systems augmenting sight have a way of displaying the combination of virtual artefacts with the real world. This category is called the display technique. However, not all AR systems implement this functionality in the same way. Some systems use a handheld device, while others implement an HMD. This means that 'handheld device' and 'HMD' are subcategories of the display technique category. In total, the seven design space categories include application area, senses, display technique, extent of presence, collaboration, system and anchoring mechanisms. These categories were devised because they were the most consistent across all studied related works. On the other hand, interaction techniques and input devices were not included in the design space, since many studies concerning these topics come up with their own custom solutions [Sid+21; Cor+20; Zha+19; Ges+20; Jam+20]. As a result, these categories would be too extensive for summarisation purposes.

Moreover, the design space with its categories and subcategories was utilised to categorise fourteen distinct studies concerning Augmented Reality. Each paper was evaluated against every category and divided into one or multiple of its subcategories. This summarisation (Figure 3.11) illustrates the most popular techniques employed for Augmented Reality development. It can be concluded that the other senses are rarely employed, since all fourteen summarised studies focused solely on sight. Moreover, it shows that most systems are designed to be operated by a single user and do not support collaboration. On top of that, it seems that an HMD is the most popular display technique. However, most studies (9 out of 11) combined the HMD with another device, resulting in a combined system. Meaning that only two of the fourteen studies used an HMD as a stand-alone system. In conclusion, most Augmented Reality studies still focus primarily on sight, while the other studied categories vary more often.

Lastly, the extensive literature study revealed that most studies focus on either custom interaction techniques or devices that can be used to interact with an augmented/virtual environment [Sid+21; Cor+20; Zha+19; Ges+20; Jam+20], specialised systems designed for a specific use case [RMS13; NM19; Bec+22; Zhu+15; CCV20; Dün+12; May+22] or improving comprehension of three dimensional data by visualising it in augmented/virtual reality [But+18; Wan+20; Rei+22]. The literature study did not really encounter Augmented Reality dashboards, meaning that ModulARboard provides a new possible direction for Augmented Reality studies. ModulARboard focuses on reimagining traditional data dashboards by providing users with customisation possibilities and ways to organise virtual items in an augmented environment, more specifically called the anchoring mechanisms.

6.1.1 Anchoring mechanisms

As mentioned before, the anchoring mechanisms form one of the categories of the design space and are very thoroughly discussed in section 3.7. They consist of marker-based, object-relative, observer-relative, view frustum and spatial anchoring. While the design space devised five anchoring mechanisms, only four of them are implemented in ModulARboard, leaving out object-relative anchoring. This is because object-relative anchoring is very similar to marker-based anchoring, since they both anchor virtual artefacts to a visually recognisable item in the real environment. With marker-based anchoring, the visually recognisable item is a fiducial marker, while with object-relative anchoring, it is a three-dimensional object. The only difference of importance would be in the algorithms needed to implement its functionality. However, the implementation cost does not outweigh its benefits, since ModulARboard is still a proof-of-concept application. The aim of ModulARboard was to explore different possibilities of anchoring artefacts to an otherwise real environment, while focusing more on interaction and HCI instead of visual computing. In the end, object-relative anchoring can always be implemented in order to fully deploy ModulARboard and seamlessly anchor virtual artefacts to three-dimensional objects without needing a fiducial marker.

The anchoring mechanisms also form one of the core functionalities of ModulARboard, along with the adaptive visualisation recommendation system. The mechanisms allow the environment to be fully utilised and personalised by positioning virtual elements all around, depending on the required setup. Even though this research did not include a user study, there are still various conclusions about these anchoring mechanisms that can be discussed. As found during the development and testing process, each anchoring mechanism has its own strengths and weaknesses. First of all, marker-based anchoring is ideal in situations that require a more dynamic setup. Markers each have a unique identifier and can be used to attach certain virtual artefacts to them. This ensures that the virtual artefacts are always anchored to the item they belong to. For example, in order to track smaller appliances that do not have a designated spot and could be placed anywhere, each appliance can be assigned a unique marker. When the appliances are moved around, the virtual artefacts bound to them will move along. However, the downside of marker-based anchoring is that it requires fiducial markers in order to make it work. For each item, another unique marker is required. This either needs to be printed out or shown via a display. This can make the setup look less appealing in a professional setting and can take some effort to set up, depending on the number of markers needed. Yet these concerns can be resolved by implementing object-relative anchoring.

View frustum anchoring is more suited to situations where it is crucial to have important information in sight at all times. Like having a window that displays real-time alerts when something happens. However, this type of anchoring should only be used for the most essential items, since the space of the view frustum is limited. Otherwise, the user's vision might get cluttered, which prevents them from seeing properly. This could prove dangerous in hazardous situations where the user is required to fully perceive their environment. On the other hand, observer-relative anchoring performs well in situations where it is ideal to have important information available that is not required to be in sight at all times. It can, for example, be used to display a map at hip level, which can be consulted from time to time by looking down. However, in the case of ModulARboard, the implemented observer-relative anchoring is approximated because the user's body movements cannot be tracked independently of the user's head movements. As a result, items cannot be anchored next to the observer, since the items would move along when turning the head sideways horizontally (turning left or right). To solve this, an extra sensor would have to be positioned at hip level to track the orientation of the body (e.g. by means of a belt). This would allow the user to position items on either side by looking to the left or right, since they are now bound to the body's orientation instead of the head's. Unfortunately, observer-relative anchoring has similar downsides to view frustum anchoring. Only a limited amount of space is available to anchor items to, meaning only a select number of items can be anchored to it. Otherwise, it would completely obstruct the user's vision. Moreover, when an item is bound at hip level, it prevents the user from seeing their feet. This proves especially dangerous in situations where the user needs to see where they are walking, like when going down the stairs. To solve this, the item can temporarily be hidden or should be transparent at all times. On top of that, it needs to be kept in mind that it can be difficult for the user to interact with items anchored to the view frustum or observer when using hand gestures as an interaction technique [Mac+03]. This is because, depending on the user's physiology, some items might be anchored too far away, meaning that they cannot be reached by the user's hands and, as a result, cannot be interacted with. This can be mediated by anchoring items closer to the observer or by using another form of interaction for items that are further away, like a controller with a raycast.

Lastly, spatial anchoring binds virtual artefacts to the physical space. This proves ideal for situations with a static setup, meaning they do not need to be moved around. In ModulARboard, users can check the contents of the spatially anchored item from any direction and even from a distance, since the item is updated to face the user at all times. Spatial anchoring is, however, less ideal in scenarios where a dynamic setup is preferred. In this case, it is better to implement marker-based anchoring. The downside of spatial anchoring is that it requires extra initialisation compared to the other anchoring mechanisms. In order for spatial anchoring to remember where all the virtual artefacts are supposed to be positioned, it needs to have some sort of understanding of the environment. The Magic Leap 2 supports this through 'spaces' and 'localisation' [ML25d]. Spaces are permanent spatial maps of the physical environment, which have to be set up by scanning the environment in detail. The HMD can then localise itself within the saved space in order to support permanent spatial markers. These markers can then be used to save the location and orientation of a spatially anchored item. With this setup, it does not matter where the HMD is turned on. It automatically localises itself in the selected space, allowing spatial anchoring to remember where to position the virtual items. However, ModulARboard utilises a temporary session map which disappears when the device is turned off, instead of a permanent space with localisation. This means that the HMD should be turned on each time in the exact same location and orientation. Otherwise, a different session map will be generated, offsetting every spatially anchored item by the newly appointed world space origin. Regardless, this shortcoming can be resolved by simply implementing support for Magic Leap spaces and localisation.

In conclusion, the anchoring mechanisms were devised by comparing many Augmented Reality studies to find similarities in the means of positioning items in AR. Hence, they answer the second part of the research question concerning the various possibilities in positioning visual elements in Augmented Reality. They allow ModulARboard to offer the user full freedom in positioning the provided items anywhere in the physical environment. Though these techniques are not limited to ModulARboard and could be applied in any Augmented Reality application. It is important to keep in mind that each anchoring mechanism has its own strengths and weaknesses, and scenarios it suits best. In general, when anchoring too many components in the environment, the user's vision can become too cluttered and overwhelming. In some scenarios, it could even prove dangerous if the virtual artefacts prevent the user from noticing hazardous situations.

6.1.2 Customisation

Other than providing ways to anchor virtual elements to the physical space, this thesis also focused on how customisation can be supported when reimagining traditional data dashboards. As mentioned before, the anchoring mechanisms and the adaptive visualisation recommendation system are two of the most important parts of ModulARboard. The anchoring mechanisms enable the user to fully customise the positioning of virtual elements in the physical space, along with the resizing and grouping functionality. All types of anchoring are available for all types of items in ModulARboard. This ensures that the system does not restrict the user from fully customising their Augmented Reality Dashboard. As a result, it is possible to anchor a label (of a group) to the view frustum or the observer. For example, when someone often switches between visualisations and does not wish to search through the dashboard for the item every time. Or in the scenario that a group visualisation is often renewed. Instead of having to regroup the components every time, or go to the marker or space where the label is anchored, the user can simply anchor the label to themselves to make access easier. Hence, it is best to support full customisability instead of limiting options according to self-determined practicality.

Additionally, there is the adaptive visualisation recommendation system, providing the user with visualisation customisability options. Depending on the selected components, the system automatically loads their data options accordingly and immediately. The selection of data options then automatically results in the generation of recommended visualisations. Depending on the selected data, these recommended visualisations could result in line, pie, bar, column, radar, scatter or even ring charts, available in different variations. Through this selection process, the user can generate visualisations fully tailored to meet their needs. These visualisations can then be resized and positioned through the previously discussed anchoring mechanisms.

ModulARboard is already highly customisable through the various personalisation opportunities like grouping components, building custom visualisations, rescaling and anchoring items. On top of that, the framework behind ModulARboard is extendable and customisable as well. Meaning that ModulARboard is customisable for the user and its framework for developers. The functionality of ModulARboard can be extended by adding support for extra data types and/or visualisations. Additional data types can be implemented by deriving them from the provided base class. This ensures that the data is equipped with a name and a measure identifier for grouping purposes. On top of that, it ensures the data type can be handled by the various builders in the visualisation recommendation pipeline. These builders can be extended as well by adding modules to the builders in the pipeline or providing new builders in general. This can be done by deriving from the provided builder base class. This ensures that an instance of the builder can be added to the pipeline of builders and the correct module is called.

All these features provide an answer to the last research question, regarding how customisability can be introduced when reimagining traditional data dashboards. In this thesis, this was achieved through providing the user with personalisation options in ModulARboard, while providing base classes to developers in order to extend and customise the framework behind ModulARboard. As a result, customisability options are not limited to usage within the application, but can also be applied externally.

6.2 Scope and Future Directions

This thesis combines various findings concerning AR from many studied sources, as well as the findings encountered through the development and testing process of ModulARboard. Although these findings are already very interesting, a user study would enable a more extensive and thorough evaluation of ModulARboard in order to deduce more qualitative and quantitative findings. A user study was not performed because the scope of this thesis focused on the extensive literature as well as the exploration of anchoring mechanisms and customisation in order to reimagine traditional data dashboards. Hence, an interesting future direction would be to perform a user study with ModulARboard.

Besides the decision not to perform a user study, this thesis focused on ModulARboard as a proof-of-concept instead of an application that is ready to be deployed, in order to preserve its scope. Because of this, some functionalities are not yet supported, like persistent storage and live data streams. However, these would require minimal additions to the ModulARboard framework. Other than that, ModulARboard suffers from some technical limitations like the limited FOV from the Magic Leap 2 (similar to other AR HMDs). Because of this, the information visible at a glance is limited, meaning that items should not be made too wide, otherwise the user has to look around. Moreover, the observer-relative anchoring in ModulARboard is an approximation of actual observer-relative anchoring. This is because there is no separate sensor to track the user's body movements independent of the head movements. As a result, the amount of space to anchor items around the user is limited, since it is not possible to anchor items on the sides. Consequently, it could prove interesting to add an extra sensor in order to implement observer-relative anchoring fully.

In general, ModulARboard provides several customisation options like choosing between recommended visualisation, building custom composite/combined visualisations, making groups, anchoring items in whatever location is desired and resizing them. Besides, the framework behind ModulARboard is customisable as well by deriving from the provided base classes. However, it would be interesting to explore more customisation options in the future. For example, instead of only summing continuous data in order to obtain a single value, the average or median value could be taken as well, as this would make more sense for certain situations. It could also be interesting to test out different encodings of the data types in order to achieve new visualisation possibilities. For example, it could be possible to convert continuous data into a histogram in order to analyse its distribution. With this, it would be possible to figure out during what times the television consumes the most power. These serve as illustrative examples of a few potential future directions. However, there are still many options that can be explored in order to provide more customisation.

Lastly, it could be interesting to explore how collaboration can be facilitated in order to allow multiple users to simultaneously set up the environment and perform data analysis collaboratively. This can be explored in the context of ModulARboard and how it can be extended in order to allow collaboration. However, it is also possible to explore collaboration in the context of ModulARboard's principles, like the anchoring mechanisms and customisation. This could allow for new AR applications to emerge outside the context of Immersive Analytics, while still employing the anchoring mechanisms.

Chapter 7

Conclusion

With the rapid improvements in the development of immersive technologies like Augmented Reality and Virtual Reality Head-Mounted Headsets, a revolution is bound to happen. It will change the way in which people use and interact with their computers. Researchers envisioned how these immersive technologies could be adapted to perform data analysis and came up with a new research field in 2015, Immersive Analytics. The relatively young research field is actively getting more attention, and researchers are exploring how immersive technologies can facilitate and improve data analysis. Yet, there is still a lot to explore and many questions remain unanswered. In order to contribute to the emerging field of Immersive Analytics, this thesis explored how Augmented Reality can be leveraged in order to reimagine traditional data dashboards. More specifically, this thesis aimed to study how virtual items can be positioned in the physical space in order to achieve Augmented Reality. On top of that, it explored how customisability can be facilitated when reimagining traditional data dashboards with AR. However, the first step in this process was to first truly understand what Augmented Reality stands for and what technologies qualify as AR.

In order to answer the first part of the research question, "What qualifies as Augmented Reality?", we had to go back to the past when the definitions of AR were formulated by Milgram and Kishino [MK94] and Azuma [Azu97]. While these definitions date back to the 1990s, they are still often perceived as *the* definitions of Augmented Reality. However, since they have become outdated with the rise of new technologies, attempts have been made to improve these definitions or come up with new ones entirely. We also found that the definitions fall short due to the exclusion of the other senses and limited distinction between 'real' and 'virtual'. Consequently, this thesis combined and slightly altered both Milgram's definition of 'real' and 'virtual', and Azuma's requirements in order to provide a more inclusive definition. This resulted in the following criteria. Something is considered 'real' when it can be sampled and resynthesised *in approximately the same location*. From the moment the location where the sampling happens is too far away from the location of the resynthesisation, so that none of the senses can perceive the sampling process, it is considered 'virtual'. On top of that, it is also considered 'virtual' when there is no sampling or resynthesisation involved, meaning that it needs to be simulated. Consequently, it is considered AR when there is a combination of 'real' and 'virtual', and it is possible to interact in real-time. This automatically means that when the sampling and resynthesisation process does not happen simultaneously, it is not possible to interact with the sampled subject in real-time, meaning that it is not AR. Ultimately, it is naturally not AR when there are no virtual elements present in order to augment the real world.

In order to answer the second part of the research question regarding different techniques to position virtual elements in a physical space, an extensive literature review had to be performed to compare these positioning techniques across various studies. This resulted in a design space consisting of seven categories, indicating similar trends amongst AR systems. On top of that, each of these categories contains a set of subcategories indicating the different adaptations within that category. One of these derived categories defines the anchoring mechanisms, including marker-based, object-relative, observer-relative, view frustum and spatial anchoring as subcategories. These anchoring mechanisms define how a virtual item can be anchored to the physical space in order to facilitate Augmented Reality. Hence, answering the second part of the research question "What techniques exist/can be applied to position visual elements in AR?". The remaining categories of the design space include the application area, augmented senses, display technique, extent of presence, whether it supports collaboration and whether it is a standalone or combined system. Subsequently, fourteen studies were selected that feature an AR system to evaluate them against the seven categories of the design space. This resulted in the summarised table as illustrated in Figure 3.11. This allowed us to conclude that it rarely occurs that a study focuses on any sense other than sight. It could also be remarked that most AR systems are designed for a head-mounted display, while this is often in combination with something else in order to form a combined system. Furthermore, it could be deduced that most AR systems focus on individual usage or feature an exocentric viewpoint. Lastly, marker-based and spatial anchoring seemed to be the most popular anchoring mechanisms among the fourteen analysed studies. Other than allowing us to form these conclusions, the design space also enabled us to form the scope of the thesis by selecting a set of subcategories from each category to focus on. In conclusion, this thesis focused on the augmentation of sight through a stand-alone HMD with an egocentric viewpoint. It is situated in the application area of presentation & visualisation. Lastly, the application is designed for individual usage and the exploration of the anchoring mechanisms. This combination yielded the proof-of-concept application ModulARboard.

ModulARboard is a prototype AR application that enables the user to divide their traditional data dashboard into separate modules and anchor them in the physical space. Hence, the name Modular AR dashboard. The prototype consists of two main parts, including the anchoring mechanisms and the adaptive visualisation recommendation system. As already mentioned, the anchoring mechanisms are responsible for the anchoring of items to the physical space. The design space, as part of the related work section, found that there are five anchoring mechanisms. However, ModulARboard supports four of them since marker-based anchoring and object-relative anchoring are very similar. The only noticeable difference for the user is the item needed to anchor the items to. This is a fiducial marker for marker-based anchoring, as opposed to any physical object for object-relative anchoring. Conclusively, it is sufficient to only support one of them since ModulARboard is still a proof-of-concept application. On top of that, during the development and testing process, we were able to assess strengths and weaknesses for each of the implemented anchoring mechanisms. In summary, marker-based and object-relative anchoring best suit dynamic setups. However, object-relative anchoring can provide a cleaner implementation without the need for fiducial markers. View frustum and observer-relative anchoring can prove useful for keeping an eye on important information. Although it should only be used for a select set of visualisations, since it can obstruct the user's view and cause hazardous situations. Lastly, spatial anchoring is ideal for static setups as opposed to marker-based and object-relative anchoring. However, spatial anchoring requires meshing algorithms in order to orient itself in a physical space. Without it, spatial anchoring does not function across multiple sessions. In general, it is best to be frugal with the number of anchored items, since it can become cluttered and overwhelming when too many items are present in the augmented environment.

While the anchoring mechanisms form one part of ModulARboard, the adaptive visualisation recommendation system forms the second half. This system allows the user to build their own custom visualisations by combining multiple components or data options. This already forms a big part of the customisation options available in ModulARboard. On top of that, the user can also choose between recommended visualisations of a single component with one selected data option. Users are enabled to anchor both labels and visualisations wherever desired, due to the anchoring mechanisms, as well as rescale them to their liking. It is also possible to combine multiple components and group them together in a label. This enables the user to give their custom names to a group of components and more easily open the composite visualisation menu (Figure 4.8) in order to compose custom visualisations. Aside from the customisation options available in ModulARboard, it is also possible to extend and customise the framework behind ModulARboard. This can be achieved by deriving from the provided base classes to extend the set of supported data types and visualisation builders, as each builder is responsible for the generation of their visualisation based on the provided data types. All of these options provide customisability to the user, as well as the developer. As a result, they answer the last part of the research question. Ultimately, this also enables us to answer the main research question: "How can augmented reality be leveraged to reimagine traditional data dashboards for immersive analytics?". Augmented Reality can be leveraged to reimagine traditional data dashboards through the implementation of anchoring mechanisms and provision of customisability towards the user.

While this study could already deduce interesting findings, a user study would be able to provide additional and more in-depth conclusions. Because of this, it would prove valuable to perform a user study in the future. Additionally, it could be interesting to further explore alternative visualisations in order to improve customisability throughout ModulARboard. For example, by offering the option to request the average or median value of continuous data, instead of only the sum. On top of that, the implementation of a histogram could prove interesting as well, to analyse the distribution of continuous data. Other than experimenting with further customisation options within ModulARboard, another future direction could be to explore how collaboration can be facilitated when working with modular AR dashboards. Ultimately, in order to fully deploy ModulARboard, it would prove beneficial to implement persistent storage and potentially live data streams along with spaces and localisation. These functionalities would make ModulARboard go from a proof-of-concept application to a final product.

This thesis allowed me to expand my knowledge of Augmented Reality and data analysis. While researching these two topics, I discovered the field of Immersive Analytics. It is still a relatively young research field, dating back to 2015. However, as it is young, it is also interesting. The field still has many directions left to further explore and many questions yet to answer. This was a really big motivator for the topic of this thesis and how I could contribute to this interesting research field. This thesis also allowed me to further develop my coding skills through Unity. While Unity can take some time getting used to, it can really prove to be a very versatile tool. Lastly, I learned how hard it can be to come up with a description in order to define something. When I first started analysing the definitions of Milgram and Kishino [MK94] and Azuma [Azu97], I could find several areas where they fell short, and I did not fully agree with them. However, when I tried to come up with an improved definition, it took me several iterations to get to the current definition as explained in section 2.2.4. Every time I thought I had it, I later found an exception to the definition or something slightly incorrect. Because of this, I realise that we need to give credit to Milgram and Azuma for the foundation that their definitions provided in the research field of Augmented and Mixed Reality.

Bibliography

[AB20] Mohammad Almasi, jordi riera jordi, and Santiago Boza. "Understanding Human Motions From Ego-Camera Videos". PhD thesis. July 2020. DOI: 10.13140/RG.2.2.31884.54409.

[Arp] *AR Advertising. 10 Powerful Examples of Branded AR Campaigns.* Arprio. URL: https://www.arpr.io/blog/augmented-reality-advertising-10-powerful-examples-of-branded-ar-campaigns (visited on 11/08/2024).

[Azu+01] R. Azuma et al. "Recent advances in augmented reality". In: *IEEE Computer Graphics and Applications* 21.6 (2001), pp. 34–47. DOI: 10.1109/38.963459. URL: https://doi.org/10.1109/38.963459.

[Azu97] Ronald T. Azuma. "A Survey of Augmented Reality". In: *PRESENCE Virtual and Augmented Reality* 6.4 (Aug. 1, 1997), pp. 355–385. DOI: 10.1162/pres.1997.6.4.355. URL: https://doi.org/10.1162/pres.1997.6.4.355.

[Bec+22] Michael Becher et al. "Situated Visual Analysis and Live Monitoring for Manufacturing". In: *IEEE Computer Graphics and Applications* 42.2 (2022), pp. 33–44. DOI: 10.1109/MCG.2022.3157961.

[But+18] Simon Butscher et al. "Clusters, Trends, and Outliers: How Immersive Technologies Can Facilitate the Collaborative Analysis of Multidimensional Data". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3173664. URL: https://doi.org/10.1145/3173574.3173664.

[Cam25] *augmented reality.* In: Jan. 15, 2025. URL: https://dictionary.cambridge.org/dictionary/english/augmented-reality?q=Augmented+reality (visited on 01/17/2025).

[CCV20] Jeroen Ceyssens, Sven Coppers, and Davy Vanacken. "Towards Traceable Design Rationale in Augmented Reality". In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems.* CHI EA '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–8. ISBN: 9781450368193. DOI: 10.1145/3334480.3382835. URL: https://doi.org/10.1145/3334480.3382835.

[Cor+20] Maxime Cordeil et al. "Embodied Axes: Tangible, Actuated Interaction for 3D Augmented Reality Data Spaces". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376613. URL: https://doi.org/10.1145/3313831.3376613.

[DB11] Andreas Dünser and Mark Billinghurst. "Evaluating augmented reality systems". In: *Springer eBooks*. Springer, New York, NY, Jan. 1, 2011, pp. 289–307. ISBN: 978-1-4614-0063-9. DOI: 10.1007/978-1-4614-0064-6_13. URL: https://doi.org/10.1007/978-1-4614-0064-6_13.

[DGW22] Mitchell Doughty, Nilesh R. Ghugre, and Graham A. Wright. "Augmenting Performance: A Systematic Review of Optical See-Through Head-Mounted Displays in Surgery". In: *Journal of Imaging* 8.7 (July 20, 2022), p. 203. DOI: 10.3390/jimaging8070203. URL: https://doi.org/10.3390/jimaging8070203.

[Dim] *Dimensions.ai*. Dimensions.ai. URL: https://app.dimensions.ai/discover/publication?search_mode=content&search_text=%22augmented%20reality%22&search_type=kws&search_field=text_search (visited on 11/11/2024).

[Dün+12] Andreas Dünser et al. "Creating interactive physics education books with augmented reality". In: *Proceedings of the 24th Australian Computer-Human Interaction Conference*. OzCHI '12. Melbourne, Australia: Association for Computing Machinery, 2012, pp. 107–114. ISBN: 9781450314381. DOI: 10.1145/2414536.2414554. URL: https://doi.org/10.1145/2414536.2414554.

[Dwy+18] Tim Dwyer et al. *Immersive Analytics: An Introduction*. Vol. 11190. Springer, Cham, Oct. 16, 2018, pp. 1–23. DOI: 10.1007/978-3-030-01388-2_1. URL: https://doi.org/10.1007/978-3-030-01388-2_1.

[EDUa] *Assemblr EDU. Fun & Interactive Learning in 3D & AR*. Assemblr EDU. URL: https://edu.assemblrworld.com/ (visited on 11/08/2024).

[EDUb] *Merge EDU. Hands-on Augmented Reality for Education*. Merge EDU. URL: https://mergeedu.com/.

[FP21] Adrien Fonnet and Yannick Prié. "Survey of Immersive Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (2021), pp. 2101–2122. DOI: 10.1109/TVCG.2019.2929033.

[Ges+20] Travis Gesslein et al. "Pen-based Interaction with Spreadsheets in Mobile Virtual Reality". In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2020, pp. 361–373. DOI: 10.1109/ISMAR50242.2020.00063.

[How24] Josh Howarth. *24+ Augmented Reality Stats (2024-2028)*. Exploding Topics. Mar. 27, 2024. URL: https://explodingtopics.com/blog/augmented-reality-stats#ar-stats (visited on 11/12/2024).

[Jam+20] Raphaël James et al. *Personal+Context navigation: combining AR and shared displays in network path-following*. 2020. arXiv: 2005.10612 [cs.HC]. URL: https://arxiv.org/abs/2005.10612.

[LR05] Christine Ludwig and Christian Reimann. *Augmented Reality: Information im Fokus*. No. 1. 2005, pp. 4–15. URL: https://www.c-lab.de/fileadmin-eim/clab/C-LAB_Reports/1_C-LAB-TR-2005-1-Augmented_Reality_Information_im_Fokus.pdf.

[Mac+03] A. MacWilliams et al. "Herding sheep: live system for distributed augmented reality". In: *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* 2003, pp. 123–132. DOI: 10.1109/ISMAR.2003.1240695.

[Mar+18a] Kim Marriott et al. *Immersive Analytics*. 1st ed. Springer Cham, Jan. 1, 2018. DOI: 10.1007/978-3-030-01388-2. URL: https://doi.org/10.1007/978-3-030-01388-2.

[Mar+18b] Kim Marriott et al. *Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation*. Vol. 11190. Oct. 16, 2018, pp. 25–55. DOI: 10.1007/978-3-030-01388-2_2. URL: https://doi.org/10.1007/978-3-030-01388-2_2.

[May+22] Kieran W. May et al. "The Identification, Development, and Evaluation of BIM-ARDM: A BIM-Based AR Defect Management System for Construction Inspections". In: *Buildings* 12.2 (Jan. 28, 2022), p. 140. DOI: 10.3390/buildings12020140. URL: https://doi.org/10.3390/buildings12020140.

[Mil+95] Paul Milgram et al. "Augmented reality: a class of displays on the reality-virtuality continuum". In: *Proceedings of SPIE, the International Society for Optical Engineering/Proceedings of SPIE* (Dec. 21, 1995). DOI: 10.1117/12.197321. URL: https://doi.org/10.1117/12.197321.

[MK94] Paul Milgram and Fumio Kishino. "A Taxonomy of Mixed Reality Visual Displays". In: *IEICE Trans. Information Systems* vol. E77-D, no. 12 (Dec. 1994), pp. 1321–1329.

[ML] *Magic Leap 2 — Augmented Reality — Optics Technology*. Magic Leap. URL: https://www.magicleap.com/magic-leap-2 (visited on 05/07/2025).

[ML25a] *Controller Overview – Care — Magic Leap*. Magic Leap. Jan. 20, 2025. URL: https://www.magicleap.care/hc/en-us/articles/4424698871565-Controller-Overview (visited on 05/29/2023).

[ML25b] *Hardware Specs — MagicLeap Developer Documentation*. Magic Leap. Jan. 20, 2025. URL: https://developer-docs.magicleap.cloud/docs/device/hardware/hardware-specs/ (visited on 05/07/2025).

[ML25c] *OpenXR Overview — MagicLeap Developer Documentation*. Magic Leap. Jan. 20, 2025. URL: https://developer-docs.magicleap.cloud/docs/guides/openxr/openxr-overview/ (visited on 05/07/2025).

[ML25d] *Spatial Mapping & Localization — MagicLeap Developer Documentation*. Jan. 20, 2025. URL: https://developer-docs.magicleap.cloud/docs/guides/features/spaces/ (visited on 05/30/2025).

[NM19] Worawan Natephra and Ali Motamedi. "Live Data Visualization of IoT Sensors Using Augmented Reality (AR) and BIM". In: *Proceedings of the ... ISARC* (May 24, 2019). DOI: 10.22260/isarc2019/0084. URL: https://doi.org/10.22260/isarc2019/0084.

[Qui] *QuiverVision*. QuiverVision. URL: https://quivervision.com/ (visited on 11/08/2024).

[Rei+22] Patrick Reipschläger et al. "AvatAR: An Immersive Analysis Environment for Human Motion Data Combining Interactive 3D Avatars and Trajectories". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517676. URL: https://doi.org/10.1145/3491102.3517676.

[RMS13] Dayang Rohaya Awang Rambli, Wannisa Matcha, and Suziah Sulaiman. "Fun Learning with AR Alphabet Book for Preschool Children". In: *Procedia Computer Science* 25 (Jan. 1, 2013), pp. 211–219. DOI: 10.1016/j.procs.2013.11.026. URL: https://doi.org/10.1016/j.procs.2013.11.026.

[Rus18] Sydney Rustemeyer. "Food Color and Perception". In: *State of Science Papers* (2018). DOI: 10.31219/osf.io/mjwav.

[Sid+21] Ludwig Sidenmark et al. "Radi-Eye: Hands-Free Radial Interfaces for 3D Interaction using Gaze-Activated Head-Crossing". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445697. URL: https://doi.org/10.1145/3411764.3445697.

[Sut98] Ivan E. Sutherland. "A head-mounted three dimensional display". In: *Seminal Graphics: Pioneering Efforts That Shaped the Field, Volume 1*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 295–302. ISBN: 158113052X. URL: https://doi.org/10.1145/280811.281016.

[SYN25] *Magic Leap 2 - Authorized reseller*. Feb. 4, 2025. URL: https://www.synergiz.com/en/produit/magic-leap-2-en/ (visited on 05/24/2025).

[Tec] Unity Technologies. *Unity - Manual: Introduction to the camera view*. URL: https://docs.unity3d.com/Manual/UnderstandingFrustum.html (visited on 05/12/2025).

[UNT] *Unity learn*. URL: https://learn.unity.com/course/magic-leap-2-development/tutorial/marker-tracking?version=2022.2 (visited on 05/29/2025).

[Van] Tomas Van Der Aa. *Training emergency services*. Flashgroup. URL: https://dev.flashgroupxr.com/ (visited on 11/08/2024).

[Wan+20] Xiyao Wang et al. "Towards an Understanding of Augmented Reality Extensions for Existing 3D Data Analysis Tools". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376657. URL: https://doi.org/10.1145/3313831.3376657.

[Yad22] Niteesh Yadav. *Understand display techniques in augmented reality*. July 1, 2022. URL: https://niteeshyadav.com/blog/understanding-display-techniques-in-augmented-reality-7485/ (visited on 01/13/2025).

[Yan+18] Yalong Yang et al. "Maps and Globes in Virtual Reality". In: *Computer Graphics Forum* 37 (June 2018). DOI: 10.1111/cgf.13431.

[YYJ11] Steve Chi-Yin Yuen, Gallayanee Yaoyuneyong, and Erik Johnson. "Augmented Reality: An Overview and Five Directions for AR in Education". In: *Journal of Educational Technology Development and Exchange* 4.1 (June 1, 2011). DOI: 10.18785/jetde.0401.10. URL: https://doi.org/10.18785/jetde.0401.10.

[ZDB08] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR". In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. 2008, pp. 193–202. DOI: 10.1109/ISMAR.2008.4637362.

[Zha+19] Yang Zhang et al. "ActiTouch: Robust Touch Detection for On-Skin AR/VR Interfaces". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 1151–1159. ISBN: 9781450368162. DOI: 10.1145/3332165.3347869. URL: https://doi.org/10.1145/3332165.3347869.

[Zhu+15] Zhiwei Zhu et al. "AR-Weapon: Live Augmented Reality Based First-Person Shooting System". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 618–625. DOI: 10.1109/WACV.2015.88.