UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

From reality to CAD: accurate feature extraction through augmented reality interfaces

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

PROMOTOR :
Prof. dr. Raf RAMAKERS

2024
2025

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

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

From reality to CAD: accurate feature extraction through augmented reality interfaces

Lennert Meurs

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

PROMOTOR :

Prof. dr. Raf RAMAKERS

Acknowledgments

I am grateful for the help I have received from my family and friends during the research and completion of this master thesis. It has been an interesting and instructive journey. From the very beginning of this project until the completion of the development and writing, I have learned a lot and I am very appreciative of that. Many people have supported me along the way and I am thankful for each and everyone's help.

First and foremost, I would like to express my gratitude to Prof. Dr. Raf Ramakers for being my supervisor, for his valuable advice and for his feedback on this project. He was always been positive and his input made it much easier for me to improve my research and achieve a higher quality of output. Every meeting with him was very valuable to me and made feel like I really wanted to continue and to make in-depth assessments and decisions.

Also dr. Danny Leen, dr. Tom Veuskens and mr. Jeroen Ceyssens deserve a big thank you for contributing their knowledge to me during the development and the implementation. Their cooperation was very pleasant and their technical assistance speeded up my problem solving and my progress greatly. I highly appreciated the willingness to support and the responsiveness with which they reacted to my questions and remarks.

I would also like to thank the rest of the research group, including Dries Cardinaels, Mannu Lambrichts, Maties Claesen, Tom De Weyer and Stig Konings. Their honest opinions and constructive feedback have contributed significantly to the successful completion of this thesis. Thanks to their input, both the system and the thesis text have been greatly improved.

Beyond the academic support, I am truly grateful to have such a wonderful family who has always been there for me throughout my education. Their encouraging words, their understanding and their support, that are important for me, have taught me to carry on and especially to continue in difficult times of my thesis. Without the understanding of my relatives and without their encouragement, it would have been much more difficult to complete the project successfully.

Finally, I like to thank everybody who somehow helped me. By giving me advice, by giving me a push in the back or just by being interested in my work, it made this thesis possible. All the support of my friends, colleagues and other people with whom I was in contact with made that I was not only happy with the destination of my trip but I also enjoyed the journey itself. I am very thankful for all the support I have received and for the motivation I have been granted concurrently.

Nederlandse Samenvatting

Introductie

Dit onderzoek verkent het potentieel van Augmented Reality (AR) als hulpmiddel om de precisie en efficiëntie te verbeteren bij het integreren van metingen uit de echte wereld in digitale ontwerpworkflows. Het onderzoekt specifiek hoe AR kan worden toegepast om het proces van het overbrengen van geometrische kenmerken van fysieke objecten naar Computer-Aided Design (CAD)-omgevingen te stroomlijnen. Het doel is een systeem te ontwikkelen waarmee gebruikers ruimtelijke informatie uit de echte wereld direct kunnen vastleggen en hergebruiken tijdens het digitale modelleerproces, waardoor handmatige invoer — die foutgevoelig is — verminderd wordt en complexe meettaken vereenvoudigd worden.

Het uitgangspunt van dit idee ligt in de moeilijkheden die ontwerpers en ingenieurs ervaren bij het modelleren van fysieke objecten met traditionele middelen. Nauwkeurige metingen vereisen vaak dat men voortdurend wisselt tussen fysieke meetinstrumenten en CAD-software, wat leidt tot onderbrekingen in de workflow en een verhoogd risico op onnauwkeurigheden. Subtiele ruimtelijke kenmerken, essentieel voor een accuraat model, zijn vaak moeilijk te detecteren of vast te leggen zonder grote inspanning. Deze masterproef onderzoekt hoe AR kan ondersteunen bij het ter plaatse uitvoeren van nauwkeurige metingen, het visualiseren van ruimtelijke relaties, en het rechtstreeks overzetten van die gegevens naar een CAD-omgeving. Hierdoor wordt handmatig werk verminderd, herhaald gebruik van meetinstrumenten vermeden en ontstaat een soepelere ontwerpervaring waarbij de fysieke en digitale wereld nauw met elkaar verweven zijn.

Op basis van deze overwegingen stelt deze thesis een centrale onderzoeksvraag op: **Hoe kunnen effectieve meet- en CAD-modelleringsmethoden worden ontwikkeld die gebruikmaken van het in-situ karakter van augmented reality-omgevingen?**

Om deze hoofdvraag te verdiepen, wordt ze opgedeeld in drie praktische deelvragen, die elk een andere invalshoek van het probleem behandelen en helpen bij het gestructureerd ontwikkelen en evalueren van het systeem:

- **Hoe kan AR worden ingezet om in-situ metingen in een driedimensionale echte omgeving te ondersteunen?**
  Hierbij wordt het gebruik onderzocht van AR-headsets (specifiek Magic Leap) om fysieke objecten direct in de omgeving waar te nemen en te meten, zodat de gebruiker geometrische gegevens kan vastleggen zonder tools te wisselen of handmatig gegevens in te voeren.
- **Hoe kunnen vastgelegde metingen efficiënt worden overgedragen naar CAD-omgevingen zoals Fusion360?**
  Dit deel onderzoekt een naadloze gegevensintegratie tussen het fysieke meetproces en het digitale modelleren, met als doel het verminderen van foutgevoelige handmatige invoer via automatisering en directe export, bijvoorbeeld via STEP-bestanden.
- **Hoe kunnen complexere of indirecte kenmerken, die niet onmiddellijk zicht-**

baar of meetbaar zijn, nauwkeurig worden vastgelegd met behulp van AR? Dit betreft technieken zoals as-decompositie, offsetvlakken en slicing om minder evidente ruimtelijke gegevens af te leiden door geometrische relaties te berekenen.

De drijfveer achter dit onderzoek is het overwinnen van beperkingen van huidige meetmethoden, die vaak gehinderd worden door onnauwkeurige sensoren, omgevingsinvloeden en noodzaak tot manuele afstemming. Via een gebruikersgerichte systeembenadering toont deze thesis aan hoe AR-interfaces intuïtieve en contextbewuste kenmerken kunnen vastleggen, met verbeterde nauwkeurigheid en gebruiksvriendelijkheid voor CAD-processen als resultaat.

Zo biedt deze studie een fundament om na te denken over hoe digitale ontwerptools nauwer kunnen worden geïntegreerd met de fysieke wereld. Door realtime ruimtelijke gegevensverzameling te combineren met gevestigde CAD-platformen, overbrugt dit onderzoek de kloof tussen manuele meetmethoden en digitale prototyping — waardevol voor ontwerpers, ingenieurs en ontwikkelaars die werken aan de volgende generatie AR-ondersteunde ontwerpsystemen.

Gerelateerd Werk

Een van de belangrijkste uitdagingen die in eerder onderzoek worden besproken, is de onzekerheid die tijdens het meetproces ontstaat [1]. Dit probleem wordt bijzonder tastbaar in alledaagse situaties—bijvoorbeeld bij het meten van de breedte van een tafel, wanneer de rand precies tussen twee millimeterstrepen valt. Het wordt dan onduidelijk welke maat leidend is, en die onzekerheid kan zich doorzetten in het verdere ontwerp, wat mogelijk tot kostbare fouten leidt. Dergelijke voorbeelden illustreren hoe kleine ambiguïteiten in handmatige metingen, gecombineerd met factoren zoals omgevingsruis, beperkte resolutie van meetgereedschap, of inconsistente interpretatie door gebruikers, de nauwkeurigheid van een model kunnen ondermijnen. Dit soort casussen vormen een nuttig referentiepunt bij het overwegen hoe digitale tools deze praktische problemen proberen op te lossen.

Op basis van deze observaties bouwt dit onderzoek voort op een brede waaier aan eerder werk rond digitale metingen, kenmerkextractie en immersieve ontwerptools. Een terugkerend thema in deze literatuur is de aanhoudende strijd om zowel nauwkeurigheid als gebruiksvriendelijkheid te bereiken bij de overgang van fysieke objecten naar digitale modellen. Hoewel eerdere systemen gedeeltelijke oplossingen bieden, schieten ze vaak tekort als het gaat om het ondersteunen van complexere of indirecte metingen op een flexibele en gebruiksvriendelijke manier.

Enkele van de meest relevante eerdere systemen trachtten de meetonzekerheid inherent aan traditionele workflows te verminderen. Tools zoals SPATA [2] en HandSCAPE [3] ondersteunen gebruikers door realtime digitale feedback en bewegingsregistratie te integreren. SPATA maakt een tweerichtingsverkeer mogelijk tussen fysieke meetinstrumenten en digitale CAD-omgevingen, terwijl HandSCAPE sensoren in een meetlint inbouwt om bewegingen te registreren en 3D-vectoren te genereren tijdens het meten. Deze systemen vormen een belangrijke vooruitgang in het terugdringen van menselijke fouten en het overbruggen van de fysieke-digitale kloof. Toch worden ze in de praktijk vaak beperkt door complexe installatie, afhankelijkheid van specifieke hardware en een gebrek aan ondersteuning voor meer subtiele of indirecte meettypes.

Wat betreft kenmerkextractie zijn traditionele tools afhankelijk van handmatige input om metingen vast te leggen. Recentere benaderingen, zoals Immersive Sampling [4], proberen dit proces te automatiseren door visuele en geometrische kenmerken automatisch te detecteren. Sommige systemen maken ook gebruik van gebarenherkenning voor metingen [5] om kenmerken te extraheren. Deze tools zijn waardevol bij exploratief ontwerp, maar beperken zich vaak tot zichtbare dimensies of grove benaderingen. Ze missen de precisie en bieden zelden een gestructureerde workflow die geschikt is voor vervolgmodellen in CAD.

Een verwante uitdaging is het bepalen welke kenmerken voldoende betekenisvol zijn om vast te leggen. Sommige tools kunnen wel randen, oppervlakken of eenvoudige contouren detecteren,

maar overstelpen de gebruiker met irrelevante informatie. Geavanceerdere benaderingen zoals SuperPoint [6] proberen dit probleem op te lossen door middel van deep learning om robuuste keypoints te identificeren over verschillende perspectieven en lichtomstandigheden heen.

Hoewel deze technieken goed presteren bij visuele matching-taken, benadrukken ze vooral algemene visuele kenmerken in plaats van de kenmerken die relevant zijn voor specifieke modelleringstaken. Uiteindelijk hangt het bepalen van welke kenmerken relevant zijn sterk af van de specifieke ontwerpopdracht van de gebruiker. In de praktijk kunnen betekenisvolle metingen alleen betrouwbaar worden vastgelegd wanneer het systeem rekening houdt met het perspectief van de gebruiker en de taakgerichte vereisten.

Een andere tak van onderzoek richt zich op tools binnen AR en VR die ontwerp en meting ondersteunen. Toepassingen zoals pARam [7], DesignAR [8] en ScaffoldSketch [9] tonen aan hoe immersieve technologie ondersteuning kan bieden bij ruimtelijk inzicht, schetsen en prototyping. Toch schieten veel van deze tools tekort wanneer hoge nauwkeurigheid vereist is. Metingen gebaseerd op gebarenherkenning of handtracking vertonen vaak drift, en de meeste systemen integreren niet goed met gevestigde CAD-software.

Deze thesis onderscheidt zich door de immersieve en interactieve voordelen van AR te combineren met de precisie die vereist is voor CAD-workflows. In plaats van ruwe metingen of visuele benaderingen te gebruiken, maakt het voorgestelde systeem gebruik van STEP-bestanden voor nauwkeurige geometrische referentie, ondersteund door ArUco-markers voor objectherkenning. Daarnaast stelt het systeem gebruikers in staat om complexe kenmerken — zoals verborgen dimensies of assen-georiënteerde afstanden — te decomponeren met ingebouwde tools zoals slicing en offsetvlakken. Op deze manier overbrugt het systeem de kloof tussen verkennende AR-interfaces en gestructureerd, foutgereduceerd CAD-modelleren.

Het Systeem

Het systeem dat in deze thesis is ontwikkeld, bouwt rechtstreeks voort op de beperkingen die in eerder onderzoek zijn geïdentificeerd en biedt een praktische, geïntegreerde benadering voor nauwkeurige kenmerkextractie binnen Augmented Reality (AR)-omgevingen. Het is ontworpen om ontwerpers en ingenieurs te ondersteunen bij het vastleggen van geometrie uit de echte wereld op een flexibele, precieze en CAD-compatibele manier — rechtstreeks vanuit een AR-interface. Het systeem draait op de Magic Leap-headset en stelt de gebruiker in staat om in-situ ruimtelijke metingen uit te voeren, terwijl het visuele contact met het fysieke object behouden blijft.

Een fundamenteel element van het systeem is het gebruik van ArUco-markers, die een cruciale rol spelen bij het uitlijnen van de virtuele en fysieke omgeving. Deze markers worden op het te meten object bevestigd en dienen als vaste referentiepunten voor de digitale overlay. Zodra een stabiel coördinatensysteem is opgezet met behulp van deze markers, kan de gebruiker interactie aangaan met digitale tools die visueel aan het object zijn verankerd in de reële ruimte.

Het systeem biedt twee primaire modi voor kenmerkextractie: directe en indirecte metingen:

- Directe metingen richten zich op kenmerken die duidelijk en zichtbaar aanwezig zijn op het oppervlak van een object. De gebruiker kan deze kenmerken rechtstreeks selecteren met behulp van een virtuele aanwijzer om geometrische primitieve elementen te definiëren, zoals afstanden, hoeken of paden. Deze metingen zijn afhankelijk van direct visueel contact en worden realtime uitgevoerd.
- Indirecte metingen daarentegen hebben betrekking op kenmerken die niet direct zichtbaar of toegankelijk zijn. Dit omvat interne structuren of ruimtelijke relaties die een contextueel begrip vereisen van de geometrie van het object.

Naast de kernmethoden voor meting ondersteunt het systeem een speciale set geavanceerde

tools — zoals as-decompositie, offsetvlakken en slicing — die gebruikers in staat stellen om meer complexe metingen af te leiden. Deze tools helpen bij het identificeren van geometrische relaties die niet direct zichtbaar of gemakkelijk bereikbaar zijn. Afhankelijk van de manier waarop deze tools worden toegepast, worden de resulterende metingen automatisch geclassificeerd als direct of indirect. Deze set hulpmiddelen vergroot het vermogen van de gebruiker om een gedetailleerd en gestructureerd ruimtelijk model op te bouwen, terwijl duidelijk blijft hoe elke meting is gecategoriseerd.

Om compatibiliteit met bestaande CAD-workflows te waarborgen, is een speciale plugin ontwikkeld voor Fusion360. Deze plugin maakt het mogelijk om de vastgelegde meetgegevens rechtstreeks te importeren vanuit het Magic Leap-systeem. Eenmaal geïmporteerd, worden de gegevens automatisch onderverdeeld in afstanden, hoeken en paden. Afstanden en hoeken worden gedefinieerd als gebruikersparameters, wat parametrisch modelleren mogelijk maakt, terwijl paden direct in een schets worden ingevoegd zonder dat de gebruiker handmatig hoeft te tekenen. Deze automatisering stroomlijnt de overgang van AR-gebaseerde vastlegging naar CAD-modellering aanzienlijk.

Al met al onderscheidt dit systeem zich niet alleen door de immersieve meetinstrumenten, maar ook door de workflowgerichte ervaring die het biedt. Het stelt gebruikers in staat om vloeiend te schakelen tussen de fysieke en digitale wereld, ondersteunt zowel eenvoudige als geavanceerde kenmerkextractie, en integreert rechtstreeks met gevestigde CAD-software — waarmee het veel van de kernbeperkingen van bestaande benaderingen aanpakt.

De kracht van contextuele kenmerktoepassing

De ware kracht van het systeem voor kenmerkextractie wordt vooral duidelijk wanneer het wordt toegepast in praktische ontwerpscenario's. Verschillende representatieve gebruikscasussen tonen aan hoe de vastgelegde geometrische gegevens geïntegreerd kunnen worden in CAD-workflows, op manieren die veel verder gaan dan het simpelweg repliceren van metingen.

Een bijzonder waardevolle toepassing betreft het gebruik van een vastgelegd pad als referentieobject in Fusion360. In plaats van een volledig 3D-scan van een object te moeten maken — vaak een tijdrovend en middelenintensief proces — kan een ontwerper met behulp van het AR-systeem snel een betekenisvol pad definiëren. Dit pad kan vervolgens rechtstreeks worden geïmporteerd in een schets en dient dan als een precieze en bewerkbare basis voor het creëren van nieuwe geometrie. Vanuit deze schets kan in Fusion360 een 'resolve'-operatie worden uitgevoerd om een 3D-referentieobject te genereren. Dat object kan vervolgens worden gebruikt als basis voor verder ontwerp — bijvoorbeeld om een op maat gemaakte bekerhouder te ontwerpen die perfect past op het gescande oppervlak. Op deze manier fungeert het AR-meetsysteem als een lichtgewicht, maar krachtig alternatief voor traditionele 3D-scanning.

Een ander belangrijk scenario is het aanpassen van bestaande parametrische ontwerpen. Wanneer een ingenieur al beschikt over een parametrisch model in Fusion360, kunnen metingen uit de fysieke wereld — zoals gewijzigde afstanden of hoeken — worden geïmporteerd en toegepast als gebruikersparameters. Hierdoor kunnen aanpassingen snel worden doorgevoerd zonder dat het model opnieuw opgebouwd hoeft te worden, wat een brug slaat tussen veranderingen in de fysieke wereld en digitale flexibiliteit. Opvallend is dat deze aanpak geen geavanceerde modelleerervaring vereist, waardoor het systeem ook toegankelijk is voor gebruikers met beperkte kennis van CAD-software.

Tot slot ondersteunt het systeem creatieve workflows door de combinatie van kenmerken uit meerdere fysieke objecten mogelijk te maken. Een ontwerper zou bijvoorbeeld bepaalde geometrische kenmerken uit twee of meer referentiestukken kunnen extraheren en combineren in een nieuw digitaal model. Deze mogelijkheid ondersteunt snel prototypen en hybride ontwerpdenken, waarbij inspiratie uit bestaande componenten direct vertaald wordt naar een nieuw, volledig geïntegreerd CAD-concept.

Naast objectgebaseerde metingen maakt het systeem ook interactie met de omgeving mogelijk. Vlakke oppervlakken in de ruimte van de gebruiker — zoals vloeren, muren of andere architecturale elementen — kunnen worden herkend en gebruikt als referentiegeometrie. Dit maakt het mogelijk om contextuele relaties vast te leggen, zoals het bepalen van de afstand tussen een tafel en een muur. Dit is bijzonder nuttig bij interieurontwerp, installaties, of het aanpassen van producten aan fysieke beperkingen in een ruimte.

Deze gebruikssituaties onderstrepen het potentieel van het systeem als meer dan alleen een meetinstrument: het is een volwaardige ondersteuning bij ontwerpideeën, aanpassingen en optimalisaties. Door te focussen op contextuele kenmerkextractie en naadloze CAD-integratie, ondersteunt het systeem zowel technische nauwkeurigheid als creatieve flexibiliteit.

Efficiëntie van het Systeem

Om de technische prestaties van het ontwikkelde AR-gebaseerde systeem te evalueren, werd een gestructureerde vergelijking gemaakt met traditionele meetmethoden. Deze evaluatie had tot doel de capaciteit van het systeem te beoordelen om geometrische kenmerken nauwkeurig en efficiënt vast te leggen en te beheren in een realistische omgeving. Aangezien het primaire doel was om handmatige gebruikersinteractie te verminderen en menselijke fouten te minimaliseren, werd bewust afgezien van een gebruikersonderzoek. Een dergelijk onderzoek zou te veel subjectieve variabelen introduceren — zoals verschillen in vaardigheidsniveau of vertrouwdheid met CAD-software — wat het moeilijk zou maken om de technische kwaliteiten van het systeem zelf te isoleren.

In plaats van snelheid als belangrijkste maatstaf te hanteren, werd het concept van “acties” geïntroduceerd om de workflowcomplexiteit te kwantificeren. Elke actie verwijst naar een afzonderlijke interactie die nodig is tijdens het meetproces — voorbeelden hiervan zijn het selecteren van een oppervlak, het opslaan van een afstand, of het wisselen tussen tools. Deze benadering sluit nauw aan bij de methodologie van deze thesis, waarin een lager aantal acties wordt gezien als indicatie van een efficiënter en minder foutgevoelig proces.

Uit de evaluatie bleek duidelijk dat er minder acties nodig waren bij het gebruik van het AR-gebaseerde systeem vergeleken met conventionele methoden. Dankzij het direct vastleggen van kenmerken uit de fysieke wereld en de gestroomlijnde export naar Fusion360, konden gebruikers repetitieve toolwissels en manuele gegevensoverdracht vermijden. Deze resultaten wijzen op een workflow waarbij de kans op fouten van nature kleiner is. Bij traditionele methoden vormt elke extra actie immers een nieuw risico op fouten — zoals het verkeerd aflezen van een maat of het foutief invoeren van gegevens in CAD-software. Dergelijke fouten worden in belangrijke mate voorkomen door het AR-systeem, dat in plaats daarvan een nieuwe uitdaging introduceert: de gebruiker moet creativiteit en contextueel inzicht toepassen om te bepalen welke kenmerken het meest relevant zijn om vast te leggen. Deze verschuiving — van mechanische herhaling naar doordachte interactie — vormt een stap richting meer intuïtieve, ontwerpervriendelijke tools.

Inzichten

Het AR-gebaseerde systeem dat in dit onderzoek wordt gepresenteerd, biedt een alternatieve benadering van de modelleringworkflow door handmatige tussenkomst te beperken en in plaats daarvan in-situ kenmerkvastlegging rechtstreeks vanaf fysieke objecten te ondersteunen. Door geometrische kenmerken zoals afstanden en paden vast te leggen en op te slaan, en deze te exporteren naar een CAD-systeem zoals Fusion360, verlaagt de methode aanzienlijk het aantal handmatige acties — en daarmee ook de kans op menselijke fouten. In tegenstelling tot traditionele methodes, waarbij gebruikers vaak moeten schakelen tussen fysieke metingen en digitale invoer, stroomlijnt dit systeem het proces en maakt het de ontwerpbeleving intuïtiever.

De effectiviteit van deze workflow werd geëvalueerd op basis van het aantal acties dat een gebruiker moet uitvoeren om een taak te voltooien. In plaats van de uitvoertijd te meten, richtte de studie zich op hoe elke afzonderlijke stap — zoals het selecteren van een oppervlak, het opslaan van een kenmerk, of het gebruiken van een tool — invloed heeft op de algehele complexiteit van de workflow. Traditionele methodes, die doorgaans meer van deze stappen vereisen, bieden meer kansen op fouten tijdens meting en overdracht. Het AR-systeem voorkomt dit door gebruikers toe te staan kenmerken direct op te slaan in de juiste ruimtelijke context. Tegelijkertijd verlegt dit echter de verantwoordelijkheid naar de gebruiker om te bepalen welke kenmerken relevant zijn, wat een zekere mate van creatief en doelgericht inzicht vereist.

Dat gezegd hebbende, heeft het systeem in zijn huidige vorm enkele beperkingen. Een belangrijk nadeel is de afhankelijkheid van STEP-bestanden als geometrische referentie. Deze bestanden dienen als proof-of-concept, maar gaan ervan uit dat bijbehorende fysieke objecten al beschikbaar én vooraf gemodelleerd zijn. Bovendien is de ruimtelijke uitlijning via ArUco-markers foutgevoelig door de handmatige plaatsing ervan, wat het opstartproces ingewikkelder maakt. Het systeem ondersteunt momenteel ook slechts een beperkte set van basistools zoals directe afstandsmeting, slicing en offsetvlakken, waardoor complexere metingen nog niet mogelijk zijn.

Met het oog op de toekomst zijn er verschillende duidelijke richtingen om het systeem te verbeteren en uit te breiden. Het vervangen van de afhankelijkheid van STEP-bestanden door automatische geometrieherkenning via een 3D-scanner zou de noodzaak voor vooraf gemodelleerde objecten en handmatige uitlijning volledig kunnen elimineren. Dit zou bovendien de bruikbaarheid verbeteren en opstartfouten verminderen. Het uitbreiden van het toolset om meer geavanceerde metingen te ondersteunen zou gebruikers in staat stellen een breder scala aan geometrieën te modelleren, zoals snijpunten en niet-lineaire contouren. Integratie van kunstmatige intelligentie voor suggestie of herkenning van kenmerken zou minder ervaren gebruikers kunnen helpen om belangrijke dimensies en eigenschappen te identificeren, wat het systeem toegankelijker zou maken.

Hoewel tijdens dit project geen gebruikersonderzoek werd uitgevoerd, wordt een dergelijk onderzoek aanbevolen als essentiële volgende stap. Zo'n studie zou inzicht bieden in hoe verschillende soorten gebruikers met het systeem omgaan, waar ze obstakels ervaren of juist uitblinken. Deze feedback is noodzakelijk om de interface verder te verfijnen en te zorgen dat het systeem zich aanpast aan uiteenlopende ontwerpcontexten. Uiteindelijk zou dergelijk toekomstig werk het systeem kunnen transformeren tot een robuust, flexibel en gebruiksvriendelijk platform voor AR-ondersteunde modellering — dat de fysieke en digitale werelden met meer precisie en efficiëntie met elkaar verbindt.

Conclusie

Deze thesis heeft aangetoond dat augmented reality het proces van het meten en overbrengen van afmetingen uit de echte wereld naar CAD-omgevingen zowel eenvoudiger als nauwkeuriger kan maken. Door veel van de manuele stappen die typisch zijn voor traditionele workflows te elimineren, en deze te vervangen door in-situ kenmerkextractie met behulp van een AR-headset, vermindert het systeem veelvoorkomende fouten en verhoogt het de efficiëntie. In plaats van voortdurend te moeten schakelen tussen meetgereedschap en CAD-software of waarden handmatig in te voeren, kunnen gebruikers nu kenmerken rechtstreeks vanuit de fysieke wereld vastleggen en toepassen in Fusion360. De evaluatie toonde aan dat er minder acties nodig waren om modelleringstaken te voltooien, en deze resultaten onderstrepen de voordelen van een meer directe en creatieve benadering van ontwerpen. Hoewel het systeem momenteel nog beperkt is door het gebruik van STEP-bestanden en de noodzaak voor manuele uitlijning via ArUco-markers, vormt het een solide basis voor toekomstige verbeteringen.

De kracht van dit systeem ligt niet alleen in de technische implementatie, maar vooral in de herziening van de manier waarop ontwerpers omgaan met objecten tijdens het modelleren.

Metingen worden niet langer behandeld als geïsoleerde stappen, maar maken integraal deel uit van de ontwerpstroom. Dit maakt het proces intuïtiever en aantrekkelijker, vooral voor gebruikers die zich willen richten op ontwerpen in plaats van op data-invoer. Hoewel er nog enkele beperkingen zijn — zoals de noodzaak voor vooraf gedefinieerde digitale modellen en een beperkt aantal tools — doen deze niet af aan het potentieel van het systeem. Naarmate de technologie zich verder ontwikkelt, kunnen verbeteringen zoals automatische 3D-scanning en meer geavanceerde meetmogelijkheden de uitlijning vereenvoudigen en het gebruik van AR in ontwerp breder toepasbaar maken. Deze benadering vertegenwoordigt een betekenisvolle stap richting het samenbrengen van fysieke omgevingen en digitale modellering op een vloeiende, praktische en gebruiksvriendelijke manier.

Survey

Introduction

This research explores the potential of Augmented Reality (AR) as a tool to improve the precision and efficiency of integrating real-world measurements into digital design workflows. Specifically, it investigates how AR can be applied to streamline the process of transferring geometric features from physical objects into Computer-Aided Design (CAD) environments. The goal is to create a system that allows users to directly capture and reuse real-world spatial information during the digital modeling process, reducing the reliance on error-prone manual entry and simplifying complex measurement tasks.

The core idea stems from the challenges designers and engineers face when attempting to model real-world objects with traditional tools. Taking precise measurements often involves switching between physical measuring tools and CAD software, leading to workflow interruptions and increased risk of inaccuracies. Moreover, subtle spatial features that are critical for accurate modeling can be difficult to detect or capture without extensive effort. This thesis responds to these difficulties by investigating how AR can assist users in taking accurate, in-situ measurements, visualizing spatial relationships, and directly translating that data into the CAD environment. In doing so, the system reduces manual handling, avoids repetitive tool usage, and promotes a more seamless design experience where real-world and digital spaces are tightly interconnected.

Based on these considerations, the thesis formulates a guiding question to explore the potential of AR as a solution to current shortcomings in feature extraction and digital modeling workflows. The central research question addressed in this thesis is: **How can effective measurement and accompanying CAD modeling techniques be devised that leverage the in-situ nature of augmented reality environments?**

To explore the main research question in more detail, the thesis breaks it down into three practical areas of investigation. These sub-questions reflect different angles of the problem and help structure both the development and evaluation of the system in a logical, hands-on way:

- **How can AR be used to support in-situ measurements in a real-world 3D environment?**
  This sub-question examines the use of AR headsets (specifically Magic Leap) to observe and measure physical objects directly in the environment, enabling the user to capture geometric data without switching between tools or manual input.
- **How can captured measurements be transferred efficiently to CAD environments like Fusion360?**
  This investigates seamless data integration between the physical measurement process and digital modeling, aiming to reduce error-prone manual transcription through automation and direct export using STEP files.
- **How can more complex or indirect features, which are not immediately visible**

or measurable, be captured accurately using AR?
This explores indirect measurement techniques such as axis decomposition, offset planes, and slicing, which allow the extraction of less obvious spatial data by computing relationships between geometric elements.

The motivation behind this research lies in overcoming the limitations of current measurement workflows, which are often hindered by sensor inaccuracies, environmental influences, and the need for manual alignment or recalibration. Through a user-centered system design, this thesis demonstrates how AR interfaces can enable intuitive, context-aware feature capturing, improving both the accuracy and usability of CAD modeling processes.

In doing so, the thesis provides a foundation for rethinking how digital design tools can be more tightly integrated with the physical world. By merging real-time spatial data collection with established CAD platforms, this research bridges the gap between manual measurement techniques and digital prototyping, offering insights relevant for designers, engineers, and developers working on next-generation AR-supported design systems.

Related Work

One of the major challenges discussed in earlier research is the uncertainty that arises during the measurement process [1]. This issue becomes particularly tangible in everyday scenarios—for example, when measuring the width of a table and the edge falls between two millimeter markings. It becomes unclear which measurement to trust, and this uncertainty can propagate through the design process, potentially leading to costly errors. Such cases illustrate how small ambiguities in manual measurements, combined with factors like environmental noise, limited tool resolution, or inconsistent user interpretation, can undermine the accuracy of a model. This example serves as a useful reference point when considering how digital tools aim to overcome these practical issues.

Building on these observations, this research draws from a broad spectrum of prior work that addresses challenges in digital measurement, feature extraction, and immersive design tools. A key theme across this literature is the ongoing struggle to achieve both accuracy and usability when transitioning from real-world objects to digital models. While earlier systems provide partial solutions, they often fall short when it comes to supporting more complex or indirect measurements in a flexible and user-friendly way.

Some of the most relevant prior systems have attempted to reduce the measurement uncertainty inherent in traditional workflows. Tools like SPATA [2] and HandSCAPE [3] aim to support users by integrating real-time digital feedback and motion tracking. SPATA facilitates a two-way exchange between physical tools and digital CAD environments, while HandSCAPE embeds sensors in a tape measure to record movement and generate 3D vectors as the user performs measurements. These systems mark significant progress in reducing human error and bridging physical-digital gaps. However, their practical use is still constrained by limitations in setup complexity, hardware dependencies, and lack of support for handling more nuanced or indirect measurements.

In terms of feature extraction, traditional tools rely on user input to identify and document measurements. More recent approaches, such as Immersive Sampling [4], attempt to automate this process by detecting visual and geometric characteristics directly. Some systems also employ gesture-based measurement [5] to extract features. These tools are useful in exploratory design but are often restricted to visible dimensions or coarse approximations. They lack precision and rarely offer a structured workflow suitable for downstream CAD modeling.

A related challenge is identifying which features are meaningful enough to extract. While some tools can detect edges, surfaces, or simple contours, they often overwhelm users with irrelevant information. More advanced approaches like SuperPoint [6] have attempted to improve this by using deep learning to identify robust keypoints across different views and lighting conditions.

However, while these techniques work well for visual matching tasks, they tend to highlight general visual features rather than those relevant to specific modeling goals. Ultimately, identifying which features are truly relevant depends heavily on the user's specific modeling intentions. In practice, meaningful measurements can only be reliably captured when the system considers the user's perspective and task-specific requirements.

Another stream of work focuses on tools in AR and VR that support design and measurement. Applications like pARam [7], DesignAR [8], and ScaffoldSketch [9] illustrate how immersive technology can assist with spatial reasoning, sketching, and prototyping. However, many of these tools fall short when high accuracy is required. For example, measurements based on gesture recognition or hand tracking tend to drift, and most systems do not integrate tightly with established CAD software.

This thesis distinguishes itself by combining the immersive and interactive advantages of AR with the precision required for CAD workflows. Rather than relying on rough measurements or visual approximations, the proposed system uses STEP files for accurate geometry reference, supported by ArUco markers for object identification. Additionally, the system enables users to decompose complex features—such as hidden dimensions or axis-aligned distances—using built-in tools like slicing and offset planes. In doing so, it bridges the gap between exploratory AR interfaces and structured, error-minimized CAD modeling.

The System

The system developed in this thesis builds directly on the limitations identified in related work, offering a practical and integrated approach for accurate feature extraction in Augmented Reality (AR) environments. It is designed to support designers and engineers in capturing real-world geometry in a flexible, precise, and CAD-compatible manner, all from within an AR interface. The system runs on the Magic Leap headset, enabling the user to perform spatial measurements in-situ while maintaining visual context of the physical object.

A foundational element of the system is the use of ArUco markers, which play a crucial role in aligning the virtual and physical environments. These markers are attached to the object being measured and serve as fixed reference points for the digital overlay. Once a stable coordinate system is established using these markers, the user can interact with digital tools that are visually anchored to the object in real space.

The system provides two primary modes of feature extraction: direct and indirect measurements:

- Direct measurements focus on features that are clearly and visibly present on the surface of an object. The user can directly select these features using a virtual pointer to define geometric primitives such as distances, angles or paths. These measurements rely on direct visual access and are executed in real-time.
- Indirect measurements, in contrast, deal with features that are not directly visible or accessible. These include internal structures or spatial relationships that require contextual understanding of the object's geometry.

In addition to the core measurement methods, the system supports a dedicated set of advanced tools—such as axis decomposition, offset planes, and slicing—that allow users to derive more complex measurements. These tools help identify geometric relationships that are not directly visible or easily accessible. Depending on how these tools are applied, the resulting measurements are automatically treated as either direct or indirect. This toolset enhances the user's ability to build a detailed and structured spatial model while maintaining clarity in how each measurement is categorized.

To ensure compatibility with existing CAD workflows, a dedicated plugin was developed for Fusion360. This plugin enables the direct import of captured measurement data from the Magic

Leap system. Once imported, the data is automatically categorized into distances, angles, and paths. Distances and angles are defined as user parameters, allowing for parametric modeling, while paths are directly inserted into a sketch without requiring any manual drawing by the user. This automation significantly streamlines the transition from AR-based capture to CAD modeling.

Overall, this system distinguishes itself by not only offering immersive measurement tools but also by enabling a workflow-oriented experience. It empowers users to move fluidly from the physical to the digital world, supports both basic and advanced feature extraction, and integrates directly with established CAD software—addressing many of the core limitations found in existing approaches.

The Power of Contextual Feature Use

The true power of the system's feature extraction capabilities becomes especially clear when applied in practical design scenarios. Several representative use cases demonstrate how the captured geometric data can be integrated into CAD workflows in ways that go far beyond simple measurement replication.

One particularly valuable use case involves using a captured path as a reference object in Fusion360. Instead of requiring a full 3D scan of an object—often a time-consuming and resource-heavy process—a designer can quickly define a meaningful path using the AR system. This path can then be directly imported into a sketch, serving as a precise and editable foundation for creating new geometry. From this sketch, a 'resolve' operation can be executed in Fusion360 to generate a 3D reference object, which can then be used as a reference to design around—for example, creating a custom cup holder that fits precisely onto the scanned geometry. In this way, the AR measurement system serves as a lightweight yet powerful alternative to traditional 3D scanning

Another important scenario is the adaptation of existing parametric designs. When an engineer already has a parametric model in Fusion360, the measurements captured from the physical world—such as updated distances or angular changes—can be imported and applied as user parameters. This allows for rapid adjustments to the design without rebuilding the model from scratch, bridging real-world changes with digital flexibility. Notably, this approach requires no advanced modeling skills, making it accessible to users who may not be experienced in CAD software.

Finally, the system supports creative workflows by enabling the combination of features from multiple physical objects. For instance, a designer could extract certain geometric features from two or more reference pieces and merge them into a new digital model. This capability supports rapid prototyping and hybrid design thinking, where inspiration from existing components can be translated directly into a fresh, fully integrated CAD concept.

Beyond object-based measurements, the system also enables interaction with the surrounding environment. Flat surfaces in the user's space—such as floors, walls, or other architectural elements—can be detected and used as reference geometry. This allows for capturing contextual relationships, such as determining the distance between a table and a wall, which is particularly useful for interior design, installation planning, or adapting products to real-world constraints.

These use cases highlight the system's potential not only as a measurement tool, but as a meaningful contributor to design ideation, modification, and optimization. By focusing on contextual feature extraction and seamless CAD integration, the system supports both technical accuracy and creative agility.

Efficiency of the System

To evaluate the technical performance of the developed AR-based system, a structured comparison was made with traditional measurement methods. This evaluation was designed to assess the system's capacity to extract and manage geometrical features accurately and efficiently within a real-world setting. Given the primary aim of reducing manual user interaction and minimizing human error, the focus was placed entirely on technical aspects, avoiding a user study. Conducting a user study would have introduced too many subjective variables, such as differences in skill level or familiarity with CAD software, making it difficult to isolate the technical merits of the system itself.

Rather than using speed as the principal metric, the evaluation introduced the concept of "actions" to quantify workflow complexity. Each action referred to a discrete interaction required during the measurement process—examples include selecting a surface, saving a distance, or switching between tools. This approach aligns closely with the methodology of the thesis, where a reduced number of actions correlates with a more efficient and less error-prone process.

The evaluation clearly showed that fewer actions were needed when using the AR-based system compared to conventional methods. Thanks to direct capture of features from the physical world and streamlined export to Fusion360, users avoided repetitive tool-switching and manual transcription of data. These results demonstrate a workflow where the likelihood of errors is inherently lower. In traditional methods, every additional action presents a new opportunity for mistakes—be it from misreading a measurement or incorrectly inputting values into CAD software. These errors are significantly mitigated in the AR system, which instead introduces a new form of challenge: the user must apply some creativity and contextual judgment to determine which features are most relevant to capture. This shift from mechanical repetition to thoughtful interaction represents a step toward more intuitive, designer-friendly tools.

Insights

The AR-based system presented in this work offers an alternative approach to the modeling workflow by reducing manual intervention and instead supporting in-situ feature capture directly from real-world objects. By capturing and storing geometrical features, such as distances and paths, and allowing their export to a CAD system like Fusion360, the method significantly lowers the number of manual actions and therefore the likelihood of human errors. Unlike traditional approaches, which often require users to alternate between physical measurements and digital input, this system streamlines the process and enables a more intuitive design experience.

The effectiveness of this workflow was evaluated based on the number of "actions" a user must perform to complete a task. Rather than measuring execution time, the study focused on how each individual step—like selecting a surface, storing a feature, or using a tool—impacts overall workflow complexity. Traditional methods, which often include more such actions, inherently present more opportunities for errors during measurement and transcription. The AR system avoids this by enabling users to store features directly in the correct spatial context. However, this also shifts the responsibility toward the user to identify which features are most relevant, requiring a degree of creativity and insight.

That said, the system has several limitations in its current form. A key drawback is the dependency on STEP files for geometric reference. These files serve as a proof-of-concept but assume that corresponding physical objects are available and pre-modeled. Additionally, spatial alignment using ArUco markers is prone to error due to manual placement and can complicate the setup process. The system also supports only a limited set of basic tools such as direct distance, slicing, and offset planes, leaving more complex measurements unsupported for now.

Looking forward, there are several clear directions to improve and extend the system. Replacing

the reliance on STEP files with automatic geometry capture through a 3D scanner could eliminate the need for pre-modeled objects and manual alignment entirely. This would also enhance usability and reduce setup errors. Expanding the feature set to support more advanced measurements would allow users to model a broader range of geometries, including intersections and non-linear contours. Integrating artificial intelligence for feature suggestion or recognition could assist less experienced users in identifying important dimensions and characteristics, making the system more accessible.

While a user study was not conducted during this project, it is recommended as a critical next step. A study would provide insights into how different types of users interact with the system and where they encounter difficulties or excel. This feedback is necessary to further refine the interface and ensure that the system can adapt to a variety of design scenarios. Ultimately, such future work could transform the system into a robust, flexible, and user-friendly platform for AR-supported modeling that bridges the physical and digital worlds with greater precision and efficiency.

Conclusion

This thesis has shown that augmented reality can make the process of measuring and transferring real-world dimensions into CAD environments both easier and more accurate. By removing many of the manual steps typically required in traditional workflows, and replacing them with in-situ feature extraction using an AR headset, the system reduces common errors and increases efficiency. Instead of repeatedly switching between tools or manually inputting values, users can now capture features directly from the physical world and use them in Fusion360. The evaluation indicated that fewer actions were required to complete modeling tasks, and these results highlight the benefits of a more direct and creative approach to design. Although the system is currently limited by its use of STEP files and manual alignment with ArUco markers, it provides a solid base for future improvements.

The strength of this system lies not just in its technical implementation, but in the rethinking of how designers interact with objects during the modeling process. Measurements are no longer treated as isolated steps but are integrated into the design flow. This makes the process more intuitive and engaging, especially for users who want to focus on design rather than data entry. While certain limitations remain, such as the need for predefined digital models and a limited toolset, these do not overshadow the system's potential. As technology advances, improvements like automatic 3D scanning and more advanced measurement tools could further simplify alignment and broaden the use of AR in design. This approach demonstrates a meaningful step toward merging physical environments and digital modeling in a smooth, practical, and user-friendly way.

Contents

Chapter 1

Introduction

The creation of 3D models can be approached in various ways, but the most prevalent method today involves the use of Computer-Aided Design (CAD) software. With CAD, users can adopt a parametric approach to design models, allowing for adjustments to be made via user-defined parameters. However, while this digital approach offers precision and flexibility, designing models that accurately reflect real-world objects is challenging because precise measurements must be taken and transferred into the digital domain. This process is prone to limitations and inaccuracies, as demonstrated in the work of Ramakers et al. [10].

Research has been conducted on techniques to mitigate errors during the measurement process, often by utilizing physical tools. For example, tools like SPATA Tools [2] and HandScape [3] enable the use of digital measurement devices to facilitate the creation of 3D models. This approach minimizes the need for users to frequently switch between measuring and modeling tasks. However, while these tools help streamline the process, they still face some challenges. They depend heavily on precise calibration and sensor accuracy, which can be affected by factors such as lighting, temperature, or even the inherent limitations of the hardware. As a result, capturing fine details can sometimes fall short. Another method, explored by Bokyung et al. [5], involves using body gestures to indicate dimensions such as width or height. While this approach is fast and efficient, it lacks the precision offered by measurement devices.

A more recent approach that has gained popularity involves the use of Mixed Reality (MR) and Augmented Reality (AR). MR blends digital content and the physical world more intimately, allowing virtual objects not only to co-exist with a physical environment, but also to interact with real-world objects in real time. Unlike AR, which simply overlays digital information on a user's view of the world, MR anchors virtual objects to physical locations and renders them responsive to the context of their environment. This combination creates an active and interactive experience where the boundaries between the physical and digital blend seamlessly, enabling applications that respond to both user input and environmental changes. For example, Stemasov et al. [11] utilize MR to generate objects by combining digital and physical components. This approach facilitates the creation of a clamping mechanism for a digital object, ensuring that it remains securely attached to a physical surface, such as a table. Furthermore, this method enables the extraction of precise measurements by directly converting them into quantifiable features. However, a notable limitation is that its accuracy heavily depends on the quality of the scanning process used to capture the physical objects.

Despite this limitation, the increasing capabilities of MR and AR stem from a series of recent technological advancements. Devices today are now equipped with advanced sensors, such as depth cameras and LiDAR, that deliver accurate environmental mapping and spatial awareness. This allows virtual objects to seamlessly integrate into the real world, advancing the immersive experience. In addition, the development of small yet potent processors has enabled complex computations demanded by MR and AR apps to be processed on mobile devices. By integrating

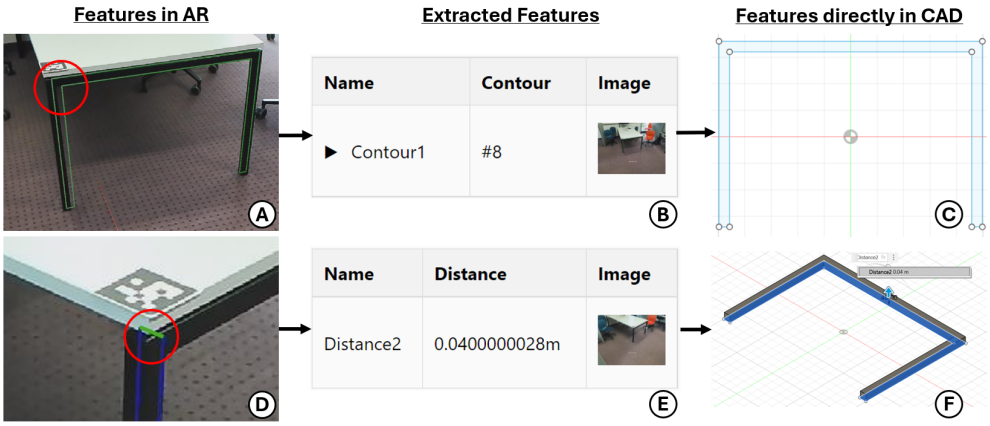

Figure 1.1: Using the system to capture geometric features and apply them in a CAD environment. (A) Capturing the contour/path feature of the table frame in the real world using AR. (B) Exporting the feature to Fusion360 with the related information. (C) Using the captured feature in a sketch so the path matches exactly. (D) Capturing the distance feature representing the frame's width in the real world. (E) Exporting the width feature. (F) Using the feature name during the extrusion of the sketch.

these methods with computer vision techniques, geometrical features from the physical world can be captured and presented to the user. This enables users to assess spatial properties and prioritize those most relevant to their needs.

1.1 Goal of thesis

While recent advancements in MR and AR have significantly improved the way we interact with and interpret spatial data, several limitations still persist in current modeling and measurement tools. Existing solutions—such as SPATA [2] Tools, HandScape [3], and various gesture-based solutions [5]—often struggle with accuracy, reliability, and ease of use, which introduces errors and inconsistencies in the modeling process. This thesis aims to address these shortcomings by exploring a new approach that enhances the precision of measurements in real-world scenarios while simplifying the process for users.

A key research challenge lies in developing a solution that improves the reliability of measurements in dynamic, real-world environments. Current methods often involve complex setups, require manual input, or depend on imperfect sensors, all of which can lead to discrepancies in the final output. The objective of this thesis is to investigate how Augmented Reality (AR) technology, specifically AR headsets like the Magic Leap, can provide a more stable and precise method for capturing real-world measurements.

One area of exploration is the integration of AR headsets with CAD software, such as Fusion360. By enabling users to capture (Figure 1.1A & 1.1D) and transfer geometrical features directly from their surroundings into a 3D modeling environment (Figure 1.1B-C & Figure1.1E-F), this research seeks to eliminate the manual entry of measurements, reducing the potential for human error [10]. Additionally, the challenge is to design a system that can efficiently handle both simple and complex measurements, from visible dimensions to more intricate ones, such as those requiring additional processing (e.g., cross-sections or hidden measurements).

Building on this foundation, the primary objective of this thesis is devising effective measurement and accompanying CAD modeling techniques that leverage the in-situ nature of augmented reality environments. To achieve this objective, the research will investigate various aspects, including:

- Novel in-situ measurement processes for 3D augmented reality environments.

- Streamlining the transfer of captured data to CAD software to reduce errors and improve efficiency.
- Methods for retrieving more complex or indirect dimensions in scenes.

To determine how well the developed system performs, a comparison was drawn with traditional measurement techniques. The result showed that the AR-based system has less user interaction to achieve the modeling goal because measurements could be read and copied directly in position. Such direct workflow reduces the number of manual steps—such as switching tools or re-typing data—which are most prone to human errors. As a result, the system is less prone to errors than traditional methods, where measurements are recorded manually and then transcribed into CAD software.

That being said, there are also drawbacks to the system. It currently supports only a basic set of measurement tools and feature types. Its usefulness could be expanded by the addition of more tools to allow more complex or less obvious features to be recorded in the future. Currently, the system uses STEP files for data transfer, which serves well as a proof-of-concept but leaves room for improvement. Since these files represent virtual 3D models, corresponding physical objects must also be present in the real world to enable meaningful interaction. To achieve spatial alignment between the virtual and physical objects, ArUco markers are used for positioning. However, this still requires a manual alignment step to ensure the virtual object matches the real-world orientation. This manual effort introduces complexity and potential inaccuracies. A possible improvement would be to incorporate a 3D scanner, which can automatically detect and capture the geometry and position of the real object. This would eliminate the need for manual alignment, as the virtual model could be directly matched to the scanned data, thereby improving ease of use and overall system stability.

Despite these limitations, the findings demonstrate how AR measurement can make it easier and better to capture object measurements for modeling. This thesis therefore aims to introduce a more stable, precise, and intuitive approach to creating 3D models from physical measurements. By bridging the gap between physical environments and digital modeling, the system offers an improved workflow for designers, engineers, and other specialists who need precise measurements in their field of work.

Chapter 2

Related Work

The work draws from and builds upon work in "Uncertainty in Measurements", "Feature Extraction", "Prioritizing Meaningful Features" and "Tools in AR/VR". Each of these areas contributes to the foundation of our system, addressing important challenges in measurement and AR.

Uncertainty in Measurements describes errors and variations that can result when users perform measurements. Human interaction, environmental factors, and limitations of measurement equipment can all contribute to inaccuracies. Understanding such causes of uncertainty is crucial in making measurement-based systems more reliable. My system attempts to minimize such issues through the use of AR to guide users and ensure accuracy.

While uncertainty focuses on the reliability of measurements, another important area is how we identify what exactly to measure. This leads us to Feature Extraction, which involves identifying and extracting significant features from objects, such as dimensions and geometric characteristics. In the traditional setting, this is normally a tedious and tool-dependent task. Our system simplifies this task by using AR to identify automatically and extract significant geometrical features from objects in the real world without requiring human intervention and reducing the risk of omitting information.

Once features are extracted, the next challenge lies in determining which of them are actually meaningful. This area is explored in work on Prioritizing Meaningful Features, which focuses on sensing interesting and distinguishing measurements between objects. While average measurements like length and width are everywhere, there are some objects with differentiating qualities that make them more valuable or interesting. My system extends this concept by making it easy for users to take and examine these differentiating characteristics, so the process of measurement becomes not only more informative but also more engaging.

Finally, there is a growing body of work exploring how AR and VR can support creative and technical tasks by bringing design and measurement into immersive 3D environments. These tools aim to make spatial interaction more intuitive. While they have opened up new ways to engage with digital content, many still struggle with issues like tracking drift, limited precision, and the lack of reliable measurement features. My system builds on this foundation, but shifts the focus toward accuracy—combining immersive visualization with exact geometric data to better support high-precision design work.

In the following subsections, we will discuss certain aspects of existing works and compare them to the aspects of my own system.

2.1 Uncertainty in Measurements

The following scenario is a common experience: "Imagine you're in the process of replacing the tabletop on your desk, and you need to measure the width of the table precisely to ensure the new one fits perfectly. You grab your tape measure, line it up carefully, and begin to take the measurement. However, as you near the edge of the table, you notice that the tape measure's millimeter markings fall right between two numbers. The edge of the table doesn't align with any specific marking."

This is a situation many people face during measurement tasks. You're trying to get an accurate reading, but factors like the precision of your tools, the alignment of the measurement, and even your vantage point can introduce slight inaccuracies. These small discrepancies can matter a lot, especially when you're working on something where precision is key, like replacing a tabletop.

Such challenges are not only common but have been studied to better understand how people approach measurements and the errors that can arise. One paper that explores this issue in depth is by Jeeeun Kim et al. [1], which investigates how people deal with measurement uncertainty—the ambiguity and imprecision that naturally occur when translating physical measurements into digital models. Through a series of studies in 3D modeling and fabrication contexts, they analyze how users experience, respond to, and compensate for uncertainty during measurement. Their work highlights how uncertainty can impact downstream processes, and they propose interface strategies that accommodate rather than ignore these imprecisions.

While understanding the causes and impacts of measurement uncertainty is essential, it's equally important to explore how interactive systems can help users avoid or reduce such errors altogether. In that light, Ramakers et al. [10] identified 10 measurement patterns—design strategies embedded in existing interactive systems that support more effective and reliable measurement activities. Rather than focusing on uncertainty itself, their work reveals how these often-hidden techniques can enhance accuracy and ease during measurement tasks.

They categorized these patterns into three distinct groups (Figure 2.1):

- Measurement Support: Focuses on the act of measuring specific dimensions accurately.
- Design Assistance: Aims to streamline the handling of measurements during design workflows.
- Craft & Fabrication Activities: Describes how measurement activities during crafting and fabrication phases can be simplified or even avoided.

By surfacing and systematizing these patterns, their work offers practical insights into how the design of tools and interfaces can directly influence the precision and reliability of everyday measurement tasks. Among these, the category of Measurement Support stands out as particularly relevant for my system, as it focuses on improving the accuracy of specific dimensional measurements through digital means.

2.1.1 Measurement Support

The first category, Measurement Support, is fully integrated into my system. It emphasizes the significant advantages of digital measurement tools over traditional analog methods, effectively eliminating scenarios like the one described earlier. Digital tools provide precise and reliable measurements, minimizing human error during the initial measurement process. For example, a digital caliper provides exact numeric readings whereas a manual vernier caliper requires the user to interpret the scale, which can lead to misreadings. In contrast, manual tools often introduce more potential for error, as they rely on the user's alignment and interpretation of measurements. Consequently, this dependence on human judgment can lead to inaccuracies that digital tools, with their automated precision, are designed to avoid.

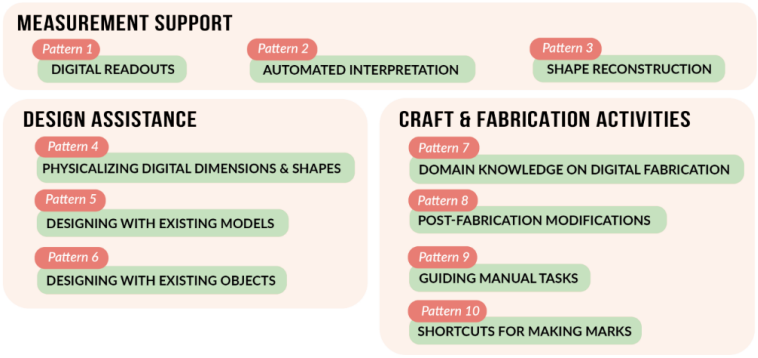

Figure 2.1: A summary of the 10 Measurement Patterns, organized into three groups: "Measurement Support", "Design Assistance" and "Craft & Fabrication Activities".

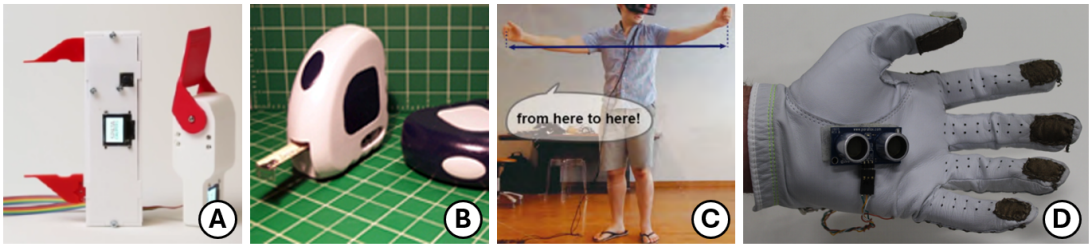

Figure 2.2: An overview of the Measurement Devices, starting with (A) SPATA tools, followed by (B) HandSCAPE, (C) BodyMeter and (D) the Ultrasonic Glove, illustrating their progression in bridging physical and digital design workflows.

Another issue arises when transferring measurements to another program or document, which introduces additional opportunities for errors. In traditional workflows, the manual process of inputting measurements into a new system or software is prone to mistakes. By integrating physical measurements directly into digital design environments, the system significantly reduces the errors associated with this manual data transfer. The integration of measurement tools with digital systems not only enhances efficiency but also ensures a higher level of accuracy throughout the design process.

Examples of systems that exemplify this integrated approach includes SPATA tools [2] and HandSCAPE [3]. These systems effectively connect physical measurements to digital models, creating a seamless bridge between the real world and the digital environment.

SPATA Tools

SPATA Tools (Figure 2.2A) exemplify this integrated approach by providing digitally enhanced measurement instruments that support a bidirectional exchange between the physical and digital design spaces. They are capable of both capturing precise measurements from real-world items and transmitting digital dimensions to physical outputs. This dual capability allows a designer to measure an object and see its dimensions immediately reflected in their CAD software, or conversely, measure a virtual shape and feel its size physically through the actuated SPATA tools. This integration creates a more intuitive, hands-on design experience that minimizes errors and streamlines the creative process.

Furthermore, these tools are designed to integrate effortlessly with fabrication-centric platforms such as mechanical CAD, mesh modeling, and 2D design environments. They support common tasks like creating and modifying geometric primitives (e.g., boxes, cylinders) by allowing users to input measurements through direct physical interaction, thus eliminating the need for manual data entry or frequent tool switching. While current prototypes are tied to computers and may

lack advanced features like high-precision depth probes, they represent a significant step forward in merging digital and physical design workflows, particularly when referencing established real-world dimensions.

HandSCAPE

Similarly, HandSCAPE is essentially a digitally enhanced measuring tape (Figure 2.2B) designed to bring traditional field measuring tasks into the digital age. By embedding an IMU sensor that tracks movement and orientation into a regular tape measure, HandSCAPE allows users to capture 3D spatial data with familiar motions. As users pull out and position the tape, the system records both distance and orientation, enabling it to calculate vectors and build accurate 3D models of physical objects or environments in real time. For example, an architect on a construction site can use HandSCAPE to measure room dimensions. As they measure, the tape automatically tracks the length, angle, and position of walls, sending the data wirelessly to a device to create an accurate 3D model of the room. However, HandSCAPE has some limitations. Errors can occur when users push the button multiple times in quick succession, leading to inaccurate 3D models. Additionally, nearby magnetic fields can interfere with the orientation sensors, affecting accuracy.

In addition to these systems, there are systems designed for more exploratory or coarse prototyping, such as BodyMeter [5] and the Ultrasonic Glove [12]. These tools leverage gesture-based inputs to measure the dimensions of objects like furniture, providing a more flexible and accessible approach to measurements without the need for highly precise instruments. This can be especially useful in prototyping or early-stage design phases.

BodyMeter

BodyMeter is a computer system that allows users to establish furniture dimensions using their body as a reference measure (Figure 2.2C). Instead of typing numbers, as in normal typing or advanced design programs, BodyMeter allows users to create full-body movements to convey dimensions. For example, the users can stretch their arms to show the size of a table or measure the height of a chair by assuming their typical posture. The system uses cameras and motion sensors to recognize the movements and translate them into actual measurements. One of the features of BodyMeter is the addition of a head-mounted display with real-time, full-scale feedback enabling users to see their design in virtual space similar to a physical prototype.

This approach to digital fabrication offers a more personalized, user-centric design process, particularly for users who are inexperienced with the traditional CAD tools. By utilizing gestures and simple voice commands like "this wide" or "from here to here," users engage with the system in an embodied, intuitive way that is directly related to their environment. BodyMeter helps users make adjustments and fine-tune their designs in real time so that their furniture perfectly suits their individual needs and available space. This design strategy builds an iterative design cycle, wherein users can experience and test their design instantly and provide feedback on it, producing a more intuitive and flexible mechanism for body-size adjustment and context-aware space adjustment of furniture.

Ultrasonic Glove

Another gesture-based system is the Ultrasonic Glove, it is a wearable input device designed for AR systems, utilizing ultrasonic transducers and a tilt sensor in the palm (Figure 2.2D). The transducers emit ultrasonic waves to measure distances, enabling intuitive, gesture-based interactions for modeling, numeric entry, and affine transformations. By using natural gestures, users can create virtual models of physical objects. The system combines distance data from the transducers with orientation information to build dimensional vectors and construct accurate models. These interactions allow for easier manipulation of virtual objects and are designed to be more intuitive than traditional input methods like the keyboard or mouse.

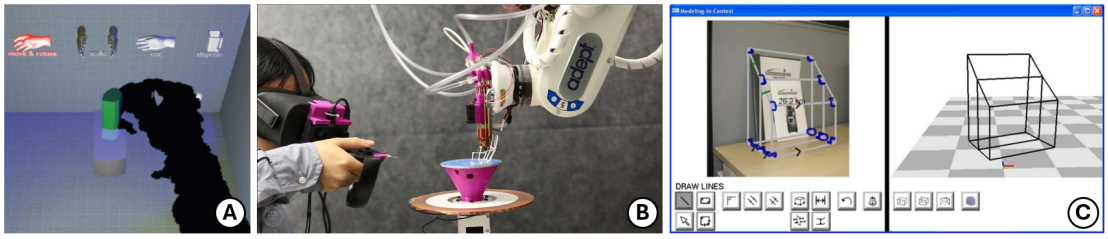

Figure 2.3: An overview of the Design Assistance Tools, starting with (A) MixFab, followed by (B) RoMa and (C) Modeling-in-Context, showcasing their capabilities in integrating physical objects into digital design workflows through AR and real-time interaction.

The Ultrasonic Glove stands out not merely for replacing analog tools with digital ones, but for reimagining how measurement and modeling can be performed altogether. Rather than relying on external tools or manual input, it allows users to define dimensions directly through spatial gestures, bringing a more embodied, expressive approach to digital design. This method reduces the cognitive load typically associated with translating real-world dimensions into virtual models. By embedding measurement into natural movements, the glove enhances user engagement and lowers the barrier for novice users who may be unfamiliar with technical design tools. It shifts the design experience from abstract and number-driven to physical and intuitive, making spatial interaction more accessible in wearable AR environments.

A challenge that arises in traditional workflows is the manual transfer of measurements into other programs or documents, a process that is prone to human error. Systems like SPATA, HandSCAPE, BodyMeter and Ultrasonic Glove tackle this by integrating physical measurement tools with digital environments, drastically reducing the potential for mistakes. My own system builds on this approach by establishing an even more efficient interface with Fusion360, ensuring that measurements captured in the physical world are reflected in the digital design. This integration streamlines the design process, reducing errors and increasing the accuracy and efficiency of design workflows.

Furthermore, the system addresses the alignment challenges associated with using traditional tools, such as tape measures, which require careful positioning to avoid errors. In contrast, the system leverages STEP files to retrieve accurate object dimensions directly. By referencing these standardized 3D representations, users can instantly access precise measurements without the need for manual alignment. This approach further reduces the potential for mistakes, ensuring that measurements are not only accurate but also contextually relevant in the digital modeling environment.

2.1.2 Design Assistance

For the second Measurement Pattern, Design Assistance, the development of virtual models is usually derived from measurements of real-world objects. This "Designing with Existing Objects" tradition enables virtual models to have very similar dimensions and features to real-world objects. The precision of measurements cannot be overstated because minute differences can dramatically affect the final design.

MixFab

Some of the existing methods that illustrate how designers can incorporate physical objects into digital workflows include MixFab [13], which leverages AR to create immersive experiences whereby users preview and design digital models in context with real objects. By scanning real-world objects, users can import their dimensions directly into the digital space, such that the models they create will be contextually relevant (Figure 2.3A). This approach enables designers to easily incorporate the physical world surrounding them into their design workflow, bridging

the gap between virtual and real-world design. However, despite its strengths, MixFab also has notable limitations. For example, the accuracy of the scanned objects depends heavily on the resolution and quality of the depth sensors used, which can result in imprecise or crude representations, especially for objects with fine details. Additionally, while the system supports gestural interaction, this form of input lacks the precision of traditional CAD tools and can make tasks such as alignment, measurement, or detailed modeling more challenging.

RoMa

Another instance is RoMA [14], which enables designers to work in a mixed-reality environment using AR headsets and controllers to communicate with a robotic arm that prints features in real-time (Figure 2.3B). This integration of design and fabrication within the same space allows for real-time adjustments based on physical limitations, enhancing accuracy and fostering innovation. As designers sketch digital components in AR, the robotic arm prints them instantly, creating an interactive and hands-on workflow. Designers can rotate the platform to view printed parts and continue building upon them. A key feature of RoMA is its ability to design directly on existing objects. For example, a designer can place a toy on the printing platform and add accessories, such as a cape, by sketching on its surface with AR controllers. The system captures the object's shape and generates a digital patch, which the robotic arm prints with precision. This approach eliminates the need for pre-scanned models, allowing for direct interaction with real-world objects.

Modeling-in-Context

Likewise, Modeling-in-Context [15] provides an interesting method where users sketch 3D models over a 2D photo of their surroundings (Figure 2.3C). The user sketches and subsequently specifies geometric constraints between the lines and the length of some edge, which immediately becomes an accurate 3D model. The method enables users to create and visualize objects within the context of their surroundings and is particularly appropriate for applications where spatial relationships and physical constraints are important. However, some limitations of this method remain. It assumes that users can provide accurate geometric constraints and dimensions, which may not always be feasible for novice users or informal design tasks. Additionally, the system requires a certain level of precision in sketching and constraint specification, potentially limiting its accessibility for users without prior experience. While the method benefits from contextual awareness, it may struggle with scenes containing complex lighting, occlusions, or ambiguous geometry in the photo background.

An improvement over these existing systems, my system is a new solution. Using AR and STEP files, the system allows users to capture precise measurements of real-world objects in their environment. The measurements are stored in a virtual inventory, where they can be accessed at a later stage in the modeling process. This allows users to focus initially on capturing the necessary measurements of real-world objects, without the pressure of needing to jump right into the full modeling process.

2.1.3 Craft & Fabrication

The last Measurement Pattern focuses on what happens after the design stage: the fabrication process itself, where errors may still occur despite careful measurement and modeling. In this project, the primary focus is on capturing and handling measurements, it does not address errors that may occur during the fabrication process itself. To tackle this, Kim et al. [1] has proposed several post-fabrication modification techniques that can assist during manufacturing. These techniques, such as buffer insertion or the replacement of minimal parts, aim to reduce the number of iterations required during fabrication. In the future, such techniques could potentially be combined with the Fusion360 integration to create a comprehensive system that supports the entire workflow—from measurement and design to fabrication—ensuring a streamlined and error-minimized process.

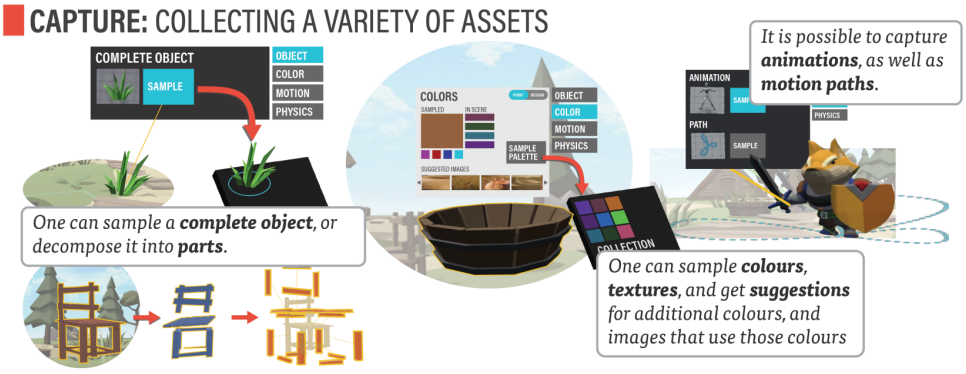

Figure 2.4: Capturing step in Immersive Sampling: From the selected object (plant, chair, character, etc.), properties of the object can be captured. Some of these properties include color, movement, separate parts and more.

2.2 Feature Extraction

The next major capability of the system is Feature Extraction: the automated identification of key object characteristics that eliminates the need for manual measurement. By simply observing objects through a device, the system can capture essential features such as dimensions, angles, paths, and other geometric attributes. This process significantly simplifies measurement tasks, as users no longer need to manually measure or input data themselves. Instead, they can specify the exact regions or aspects of an object from which they want to extract geometrical features, allowing for a more flexible and tailored approach to analysis.

Feature extraction in Virtual Reality (VR) and Augmented Reality (AR) has been studied extensively, with one notable contribution by Stemasov et al. [4] that introduces the concept of Immersive Sampling (Figure 2.4). This innovative approach leverages VR to allow users to explore, modify, and reuse various features of digital models—much like accessing and borrowing materials from a public library. By enabling creators to design new virtual environments from pre-existing spaces, Immersive Sampling not only saves significant development time but also fosters creative experimentation and iterative design.

At its core, Immersive Sampling offers a robust framework where users interact with models to reveal intrinsic features—such as color, animation, sound, and component details. These features are then stored in a centralized inventory, which can be systematically retrieved and recombined to generate novel designs. This process eliminates the necessity to build virtual objects from scratch and encourages a modular, mix-and-match approach to design.

A central aspect of this methodology is its adoption of an object-oriented approach [16] for feature management (Figure 2.5). This paradigm draws a strong parallel to object-oriented programming (OOP), where classes encapsulate both attributes and methods that can be instantiated as discrete objects. In the context of Immersive Sampling, each digital model is conceptualized as an object, containing a set of features that define its properties—such as dimensions, color, material, and additional component details. This organization facilitates the rapid reuse and modification of features across different models. For example, when a user interacts with a virtual chair, the system might expose key attributes like material, height, and armrest dimensions, which can later be applied to other objects to create varied or entirely new designs.

Stemasov et al. [11] expanded the concept of feature extraction beyond virtual environments by utilizing AR to incorporate features from the physical world. Instead of creating entirely virtual spaces, they demonstrated how features from real-world objects can be used to generate new models. Their approach addresses a key limitation of traditional 3D modeling: it requires significant technical expertise and often involves time-consuming adjustments to physical ob-

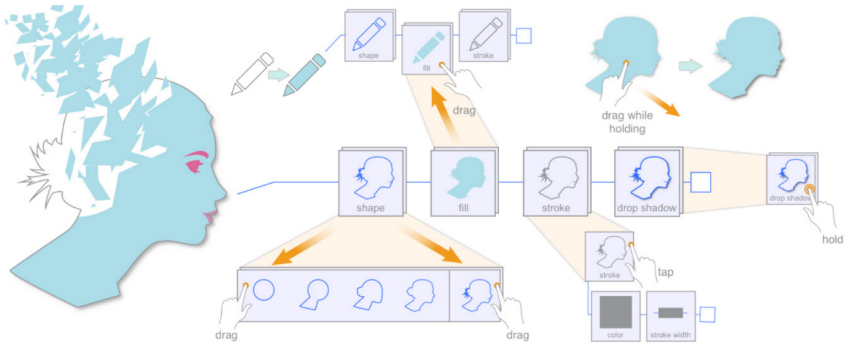

Figure 2.5: Object-Oriented Drawing interactions by Xia et al. [16]

jects. To tackle this, Stemasov et al. introduced a pick-and-choose paradigm, enabling users to browse a repository of existing models and adapt them using constructive solid geometry (CSG) operations. Moreover, their system allows for real-world objects to be scanned into 3D models, from which features can be extracted. For instance, a digital model could be aligned to the thickness of a real-world table to create a clamp that fits perfectly. This integration of AR and feature extraction bridges the gap between the physical and digital worlds, enabling users to blend real-world characteristics into their designs with minimal effort and expertise.

Unlike the system by Stemasov et al. [11], my system is not a modeling tool but rather a capturing tool. This distinction means that it does not utilize a fully object-oriented approach. Instead, my system focuses on allowing users to extract features from real-world objects and store them in an inventory for later use in CAD applications. This enables users to access relevant information quickly during the modeling phase, reducing reliance on physical tools to manually acquire these features. In its current state, my system empowers users to selectively capture features from their surroundings, such as dimensions or shapes, without requiring extensive manual effort. For example, a user might extract the diameter of a cylindrical object or the thickness of a surface directly into their inventory. In the future, the approach of Stemasov et al. [11] could be incorporated to further enhance the system's capabilities. This would allow for even faster and more comprehensive feature extraction, reducing the need for users to model objects manually within a CAD environment. By integrating these advanced techniques, the system could eventually evolve into a more robust tool that supports both capturing and modeling workflows seamlessly.

2.3 Prioritizing Meaningful Features

While feature extraction simplifies the process of identifying characteristics from objects, it presents a significant limitation: objects often contain an overwhelming number of features. To avoid overwhelming the user with unnecessary information, it is critical to extract only the most relevant features within the context of a specific application or design goal. These relevant features are not solely determined by the type of object but are strongly influenced by how the object will be used. For instance, general features of a table might include surface dimensions or leg height, but when designing a clip-on accessory, precise details such as the thickness or edge shape of the tabletop become critical. Similarly, in other design contexts, different features may become important depending on the functional requirements of the intended solution.

To investigate how systems can be designed to extract contextually relevant features, we examine two different approaches that demonstrate how the identification of key features depends on usage and user input. These approaches highlight the importance of aligning feature extraction with the specific goals of the application or design task.

Sehgal et al. [17] developed a system for automatically measuring body dimensions using 2D

images. This was achieved by initially collecting data from test subjects to establish a set of averages for key body measurements. For a new 2D image, the system determines the person's height and midline, then uses the averages to map horizontal levels relative to the midline. These levels correspond to specific body regions, such as knees, hips, and shoulders. Key marker points are placed at these levels, enabling precise measurements based on these reference points. This system demonstrates the importance of identifying contextual features that are relevant to a specific domain. For body measurements, the focus is on anatomical landmarks, and the system tailors its approach to capture dimensions that align with this context.

In contrast, UniPose [18] offers a more versatile system that works for any object type. Instead of relying on pre-defined markers, it selects key points based on user input. These points are highly adaptable to the type of object being analyzed. For example, a table might have points at each corner of its tabletop and the bases of its legs, while a human might have points at joints or facial features like the eyes. UniPose utilizes both textual and visual input to determine these key points. Through textual input, the user provides a description of where the points should be placed, such as "corners of the table" or "elbows and knees of a human figure." Visual input, on the other hand, uses a reference image of a similar object with annotated key points. The system analyzes this information and applies it to the user's object. This process is enabled by an encoder-decoder architecture:

- The encoder processes the input data and encodes it into a representation containing all relevant information.
- The decoder generates output based on this encoded representation, identifying and applying key points to the object provided by the user.

As demonstrated by systems like those of Sehgal et al. [17] and UniPose [18], human input remains essential in identifying context-relevant features. The relevance of specific features is highly dependent on the object and the user's goal. For instance, a designer working on a chair might focus on structural points, while an animator might prioritize aesthetic or movement-related features. Consequently, no automated system can reliably determine the most interesting features without some form of contextual guidance provided by the user.

In my system, the user plays an active role in identifying the features of interest by interacting with specific points on an object. This approach leverages the user's contextual knowledge, as the user is best positioned to know which features are important for their particular use case. When the user interacts with a specific part of the object, the system highlights and presents features relevant to that interaction. For example, if a user is designing a replacement tabletop, interacting with the tabletop's surface might reveal geometrical features such as dimensions and material thickness. On the other hand, interacting with the leg of the table might provide information about its height, diameter, and connection points. By enabling users to focus on specific parts of the object, the system ensures that only relevant features are displayed, reducing information overload and increasing efficiency.

This interaction-based approach not only simplifies the process for the user but also ensures flexibility across various contexts and object types. While fully automated systems like UniPose [18] rely heavily on pre-defined algorithms, my system empowers the user to guide the feature selection process, making it adaptable to a wider range of use cases. In the future, incorporating advanced techniques such as those from UniPose [18] or Sehgal et al. [17] could further streamline feature selection, allowing for even faster and more precise identification of context-specific features.

2.3.1 Measurements using Visual Computing

In addition to user-driven methods, there are several visual computing techniques that can be employed to identify interesting features in objects. One prominent method is the Scale-Invariant Feature Transform (SIFT), which is widely used in computer vision for detecting and describing local features in images. The SIFT is an algorithm that identifies and extracts

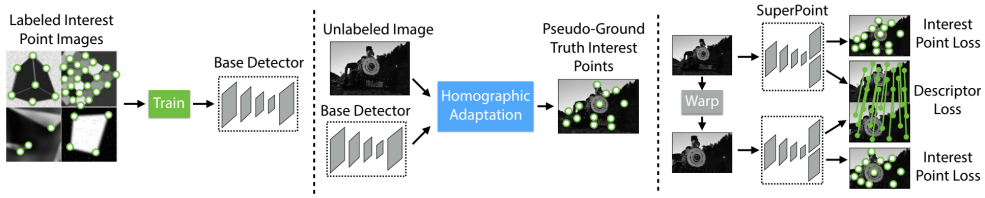

Figure 2.6: The architecture of the SuperPoint technique.

distinctive features from images, regardless of scale, orientation, or lighting conditions. While it is a proven technique for matching features across images taken under varying conditions, its robustness primarily applies to image comparison and alignment. This technique is primarily used to determine whether two images correspond or match with one another. It could, however, also be adapted to identify features within images. In this case, the extracted features would primarily be based on the need to compare and align images rather than directly identifying object-specific features.

A limitation of traditional computer vision techniques like SIFT is that they rely on mathematical methods, which restrict their adaptability when dealing with complex variations in object features or environments. SuperPoint [6] addresses this issue by leveraging deep learning, enabling the system to adapt and learn to recognize more intricate and varied patterns. SuperPoint achieves this through a two-part architecture (Figure 2.6): the MagicPoint Convolutional Neural Network (CNN) and Homographic Adaptation. The MagicPoint CNN is a neural network trained on simple geometric shapes, such as squares, circles, and cylinders. It efficiently identifies key points on these basic structures, acting as a foundational model for feature detection. The Homographic Adaptation component builds upon this by introducing transformations to the images, such as rotations or perspective shifts. This allows the network to generalize and learn to detect features from different viewpoints, improving its ability to handle complex and diverse objects.

While this advanced technique offers significant benefits, such as robust keypoint detection and adaptability to variations, it has limitations for our system's specific use case. SuperPoint primarily focuses on identifying visible, external features of objects. In our system, however, the objective is to allow users to select features relevant to their needs—features that may not always be directly tied to visual keypoints. Although SuperPoint offers a highly advanced way of detecting keypoints using deep learning, its inability to provide context-aware, user-defined features makes it less suitable for my system. The focus of my tool is to allow users to extract features directly relevant to their specific design tasks, rather than relying on an automated system that highlights only general points of interest.

Although automated keypoint detection systems like SuperPoint provide powerful capabilities, their limitations highlight the need for more interactive and user-centric approaches to feature selection. This is where immersive technologies such as Augmented and Virtual Reality offer compelling alternatives.

2.4 Tools in AR/VR

AR and VR technologies have found their way into all kinds of domains—design, education, training, data visualization, and more. What makes them especially interesting is how they allow users to interact with digital content in a spatial, often intuitive way. Instead of working through flat screens, people can engage directly in three-dimensional environments, which opens up new possibilities for understanding, creating, and collaborating. Because of that, a wide range of tools have been developed over the years to explore what AR and VR can offer.

A lot of this work has focused on tasks like prototyping, sketching in 3D, or aligning digital elements with the physical world. These tools often aim to enhance creativity, make spatial

reasoning easier, or support collaboration between remote users. Many are designed to be quick and interactive, letting users experiment freely within an immersive space. That said, while there's been plenty of innovation in how users interact with content, one thing that hasn't been explored as much is the ability to measure things—specifically geometric features like distances, angles, or spatial relationships—within these environments. Even though AR and VR are all about space and scale, tools that let users take precise measurements are still surprisingly rare.

In the next part of this section, we'll take a closer look at four tools—pARam, DesignAR, Scaffold Sketch, and RefAR—that show some of the different directions this space has taken. Each of them offers a unique perspective on how AR or VR can be used to support creative or technical work. At the same time, these examples help highlight a common limitation across many systems: the lack of built-in support for accurate, user-friendly spatial measurement. It's a gap that continues to shape how effective these tools can really be in practice.

We begin with pARam, a tool that stands out for how it blends parametric modeling with an immersive, real-world interface.

2.4.1 pARam

pARam [7] is a mixed-reality prototype built for Microsoft's HoloLens 2 that brings the power of parametric design out of the CAD suite and into your living room. Rather than staring at sliders and numerical fields on a flat monitor, you choose from a curated set of fourteen model families—everything from lampshades and vases to side tables and shelving units—and then place a live, fully manipulable preview directly into your physical space. Once it's there, you can tweak virtually any dimension on the fly: pull up 2D sliders for coarse adjustments, grab and drag handles on the model itself for more tactile control, or even sketch free-form Bézier curves in mid-air to reshape contours. Every interaction triggers an immediate visual update, so you get instant feedback on how your design evolves in context.

When you need to anchor your virtual object to a real-world measurement, pARam offers intuitive in-air "pinch" gestures: bring thumb and forefinger together to mark a point, or stretch two fingers apart to span a distance across your environment. The system reads those inputs and imports them as exact values into your model. You can then run a quick "drop test," which simulates lowering the virtual object onto the scanned mesh of the room. By doing so, pARam checks for balance and collision issues—basically, it tells you whether your design will sit flush on the surface or if it might tip over, jam against a wall, or float unnaturally. A basic point-light shader even sketches in rough shadows so you can get a sense of how lighting and occlusion will interact with your surroundings.

Despite its impressive capabilities, pARam still carries a few inherent trade-offs. Those mid-air measurement gestures feel remarkably natural, but the HoloLens 2's spatial tracking can introduce slight deviations—typically between two and ten millimetres. That level of inaccuracy might be fine for sketching new table forms or experimenting with lampshade curves, but it becomes a liability when you require engineering-grade tolerances. Similarly, while the built-in library of parametric generators lets you jump straight into design, it's a fixed collection: to try out a brand-new generator, you must first build it in an external tool (such as Archimatix) and then import it manually into pARam.

By contrast, our system takes a fundamentally different approach to spatial accuracy. Rather than relying on gesture-based tracking or headset SLAM to capture measurements in real time, we integrate directly with high-precision 3D STEP models—standard files that already encode every dimension, angle, and surface relationship with millimetre-level exactness. Because we extract data straight from the CAD-grade geometry, there's no drift or tracking error to contend with. The result is a workflow that not only lets you visualize and manipulate designs in situ, but also guarantees the kind of dimensional fidelity required for technical review, fabrication, and rigorous spatial analysis.

While our system emphasizes dimensional precision through CAD integration, other approaches focus on enhancing spatial interaction and user experience through immersive input and visualization methods.

2.4.2 DesignAR

DesignAR [8] transforms a regular design workstation into an immersive modeling environment by seamlessly combining a tilted, high-resolution multi-touch and pen display with a head-mounted AR headset. Instead of limiting 3D work to a flat screen, users can view their models floating in space above the display, while the surface itself provides precise pen and touch input. This "augmented display" setup blends the hands-on control of a tablet with the spatial depth of augmented reality, making it easy to sketch, browse, and manipulate 3D objects in a more intuitive way.

Building on that setup, the system supports a wide range of creative workflows. Users can start from a simple sketch or traced outline and quickly turn it into a full 3D shape, or choose from a built-in library of primitives and past designs. Models can be adjusted and refined on the fly, with real-time updates and multiple floating viewpoints that make it easier to understand geometry from all angles. Menus and tool palettes can be moved into the AR space around the display, keeping the main workspace clear while still offering easy access to functions. Once a model takes shape, live previews can be placed into the real-world environment, updating instantly with each change to help users evaluate their designs in context without breaking their workflow.

While the experience is fluid and well-integrated, there are a few technical trade-offs. Spatial tracking can drift by a few millimetres, the headset's narrow field of view can occasionally cut off parts of the model, and interacting with the display through overlaid 3D content can feel awkward. However, to support greater accuracy, the system relies not only on spatial positioning but also on its pen-based display and dashboard interface, which allow for more controlled and deliberate modeling. Although it doesn't use STEP files like our system, this structured input approach still offers a more dependable way to define and edit geometry than purely gesture-driven tools.

Other systems address the challenge of precision in immersive design by drawing on familiar analog workflows, translating 2D drafting techniques into spatial environments to support more accurate freeform creation.

2.4.3 Scafold Sketch

ScaffoldSketch [9] is a VR tool that helps industrial designers create accurate 3D sketches by adapting a familiar two-step drawing process from traditional 2D design. Instead of trying to draw precise shapes freehand in space—which can be challenging and error-prone—users first draw simple scaffolding lines that act as construction guides. These lines are then automatically corrected by the system to snap into proper alignment, preserving relationships like parallelism, perpendicularity, and equal lengths. Once the scaffold is in place, users can add shape curves that define the final form, which are also auto-corrected to follow and flow smoothly through the scaffold's key points.

This approach gives designers more control and structure, letting them produce clean, geometrically consistent 3D sketches that are both technical and expressive. Rather than prioritizing freeform creativity, the system emphasizes precision—enabling features like straight edges, correct angles, and symmetrical forms. While it doesn't use imported 3D models like STEP files, it shares a similar goal with our system: improving spatial accuracy in immersive design environments. By leveraging a pen-and-paper-like workflow and combining it with intelligent auto-correction, ScaffoldSketch makes complex 3D drawing more approachable and accurate without requiring traditional CAD skills.

While ScaffoldSketch improves accuracy through structured drawing techniques, other systems take a more context-driven approach—using real-world geometry as a guide for creating precise models in space.

2.4.4 RefAR

The RefAR system [19] offers a novel approach to 3D modeling in Augmented Reality (AR). It is designed to help users, especially those who are not as comfortable with 3D modeling, to sketch shapes by sketching directly in the air and using real-world objects and recorded environment point clouds as on-the-fly references. It is basically enabling you to "draw" your 3D model around things that are present. The system employs a Microsoft HoloLens 2 to capture these point clouds, then refines the user's original 3D strokes by aligning them with this data, yielding a more accurate 3D model.

RefAR also uses a neural network to predict the shapes the user had in mind for their sketches, which helps to automate the refinement process and makes the whole interaction feel more natural. There are some limitations to be aware of, however. Most significantly, the quality of the final model still relies on the correctness of the user's initial mid-air sketching, and the quality of the environmental data captured also counts. This means that incorrect sketching or omitted environmental scans can lead to flawed results.

In contrast to RefAR's method, which is based on the accuracy of the user's freehand sketching and the fidelity of real-time environmental capture, our system, as explained previously, does it differently in a bid to attain accuracy. We focus on inherent accuracy by bridging directly from high-precision 3D STEP models. These generic CAD files inherently possess highly accurate dimensional and relational information. By extracting the data directly from this geometric data, we exclude the risk of drift and imprecision inherent with AR systems like RefAR that rely on real-time tracking. This inherent difference makes it possible to have a workflow that can support in-situ visualization and manipulation as well as guaranteeing the level of dimensional precision required for high-end technical applications such as design reviews, manufacturing, and fine-scale spatial analysis.

In summary, the four tools—pARam, DesignAR, ScaffoldSketch, and RefAR—show how AR and VR can make 3D work feel more natural. pARam lets you grab and tweak shapes in mid-air, DesignAR puts your model on a touch screen that you can lift into space, ScaffoldSketch uses simple guide lines to clean up your freehand drawings, and RefAR snaps your sketches to real-world scans so they look more "correct." Each one brings a neat trick to the table for making digital objects feel life-like and easy to shape.

But there's a catch: small tracking errors or a limited view might be fine for quick ideas, yet they become a real problem when you need exact measurements. A few millimetres of slipping or a half-hidden corner can derail a design that's supposed to fit together perfectly or match a factory's specs.

That's where our system comes in. Instead of guessing sizes from hand motions or headset tracking, we load full CAD models that already know every dimension—every length, angle, and curve is exact. You still see and move your design right in your room, but now you can trust that what you're looking at matches the real, build-ready object down to the last millimetre. It's the best of both worlds: an easy, hands-on experience with the accuracy you need to actually make things.

Chapter 3

Concept

As previously mentioned, there are numerous ways in which users can make mistakes during the measurement process. To address this, I have developed a system designed to guide users and minimize the risk of such errors. This system allows users to perform measurements by observing and interacting with objects. Through the use of various tools, both visible and hidden features can be selected, enabling more comprehensive and accurate data collection.

To extract these features from objects, it was first necessary to develop a method for obtaining a 3D model of the object. Initially, we considered using a 3D scanner integrated into the AR headset. However, this approach introduced several limitations: for instance, the user would need to scan the entire room before any measurements could be taken. As a result, we opted for a proof-of-concept approach that assumes the existence of a 3D STEP file for each object. By leveraging ArUco markers, the system can detect which object the user is observing and retrieve the corresponding STEP file.

Once the necessary measurements have been taken, the collected data can be transferred to Fusion360. This allows users to seamlessly integrate the measurements into their workflow and use them as a basis for creating new designs.

In this chapter, we begin with a comprehensive, step-by-step guide that demonstrates how the system operates, outlining the specific actions users need to take to activate and use each tool effectively. This foundational walkthrough is designed to help users navigate the system with confidence. Following this, we cover the process of recognizing new objects in the room by using ArUco markers, which enables the system to accurately map and interact with the physical environment. We then provide a detailed overview of all available tools, explaining their individual functions, key features, and how they contribute to the overall workflow. To conclude, we present several practical examples that showcase real-world use cases, illustrating how the system can be successfully applied to common design tasks and offering insights into its practical value.

3.1 Walkthrough

In this section, we will walk through a practical example to demonstrate how key features can be used effectively. The example project involves creating a clamp that holds a spool of filament for a 3D printer. This task has been chosen because it allows us to cover most of the basic features in a simple and meaningful way. As we build the clamp step by step, you will become familiar with the essential tools and how they work together. In later sections, we will explore more advanced tools through additional real-world examples.

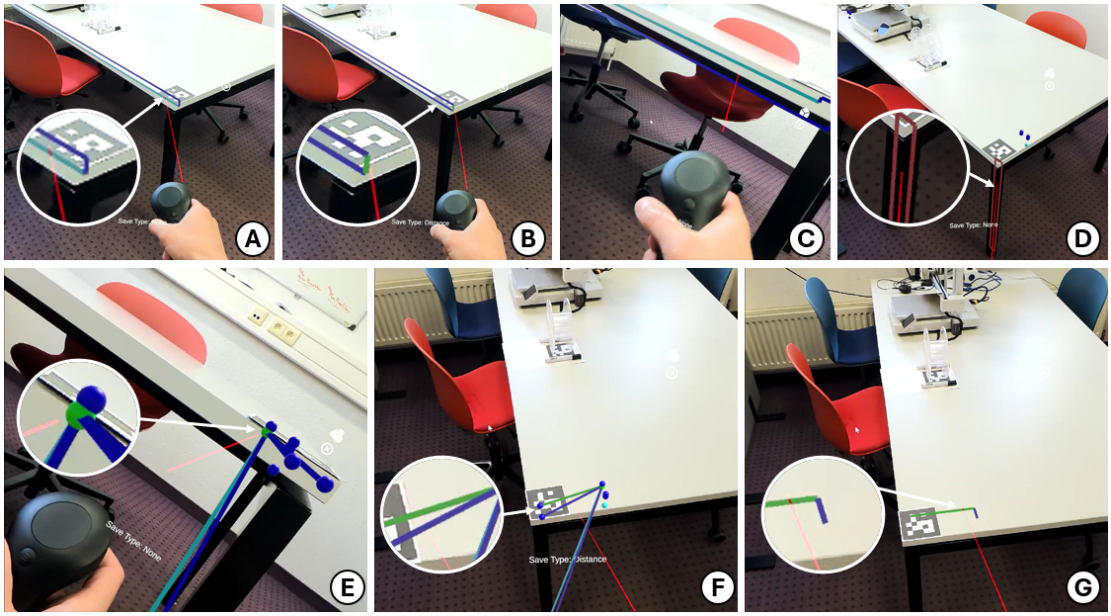

Figure 3.1: Step-by-step guide for extracting table dimensions: (A) Selecting side of tabletop. (B) Highlighting the height. (C) Selecting the surface of the frame. (D) Choosing second surface. (E) Picking a vertex. (F) Pick out the right feature. (G) Performing Axis Extraction on F.

3.1.1 Step 1: Extracting Table Dimensions

To create the required clamp, the necessary dimensions must first be collected in order to design it. Since the clamp will be attached to the tabletop, the most critical measurement is its thickness. To begin, the user must first select a surface—specifically, the side of the tabletop—by pointing the controller at it and confirming the selection with the trigger button. This surface selection is required to let the system know where the measurement will take place, and visual feedback is provided to confirm the user's choice (Figure 3.1A). After this step, the system automatically switches to Direct Tool mode. In this mode, the user can store important features such as distances, angles, or paths by selecting relevant edges—these are the geometrical features that are directly visible to the human eye on the real-world object. As the user points the controller at different edges, they are highlighted in light blue to indicate which edge is currently being targeted. To measure the tabletop's thickness, the user points the controller at the edge representing the vertical height, selects it using the bumper button (Figure 3.1B, the edge becomes green when it is selected), and then stores the distance feature by swiping upward on the touchpad. This intuitive interaction allows for quick and accurate measurements, improving both efficiency and the overall user experience. Additionally, all stored features are saved and made accessible in the inventory, so the user can easily reference them later during the design process.

After obtaining the tabletop thickness, the user notices a structural bar mounted underneath the table. This could present a problem when designing the clamp, as the bar might obstruct it and prevent proper attachment. To avoid this, it is useful to also measure the depth between the tabletop and the bar to determine the maximum usable space. The user starts by selecting the surface of the bar below the table (Figure 3.1C). Then, to define the spatial relationship, a second surface is selected—namely, the side of the table leg (Figure 3.1D). Because two surfaces have now been selected, the system automatically switches to Indirect Tool mode. In this mode, features that are not directly visible can be derived through calculated geometry. The user is prompted to select a vertex on the first surface (the bar). Once this vertex is selected (Figure 3.1E, the green sphere), the system calculates Euclidean distances from that point to all vertices

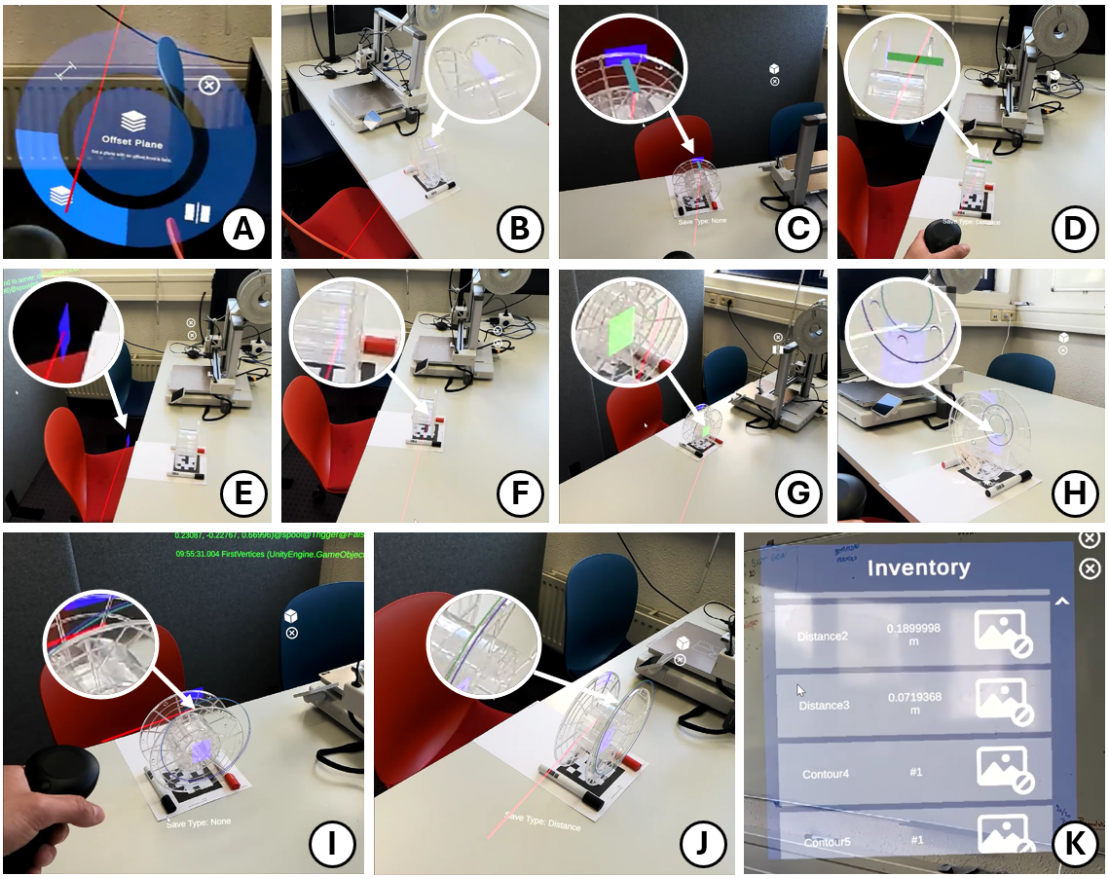

Figure 3.2: Step-by-step guide for extracting spool dimensions: (A) Activating specific tools. (B) Creating a plane. (C) Using the Straight Distance tool. (D) Select/Save feature from C. (E) Creating and moving plane. (F) Move plane from F to center of spool. (G) Utilize the Slicing tool. (H) Select/Save feature from G (I) Selecting upper face of spool. (J) Select/Save feature from I. (K) Viewing features in Inventory.

on the second surface (the table leg). The user then selects an edge based on these calculations (Figure 3.1F, the green one), but in this case, the chosen edge is angled and not perpendicular to the tabletop. Since the goal is to obtain the true vertical distance, the user performs axis extraction by using a swipe-down gesture on the touchpad. This action decomposes the angled edge into its X-, Y-, and Z-axis components, which are then visually displayed in the scene (Figure 3.1G). The user can then select the correct, perpendicular edge and store it, ensuring that the measurement accurately reflects the available vertical clearance for the clamp.

3.1.2 Step 2: Extracting Spool Dimensions

Once the necessary features of the table have been captured, the user proceeds to gather features required for positioning the spool onto the clamp. One of the key measurements needed is the width of the spool. This can be obtained using various tools, but in this case, the user chooses the Straight Distance Tool. Before this tool can be used, a plane must first be created. The user activates the Plane Tool by holding the trigger and selecting it from the menu (Figure 3.2A). After activation, the user is prompted to select a flat surface—specifically, one side of the spool—by pointing with the controller and confirming the selection with the bumper. Once selected, the plane becomes visible in the scene (Figure 3.2B). The user then activates the Straight Distance Tool in the same way: by holding the trigger and selecting the tool from the menu. This tool requires a plane to be selected first, followed by a point elsewhere on the object

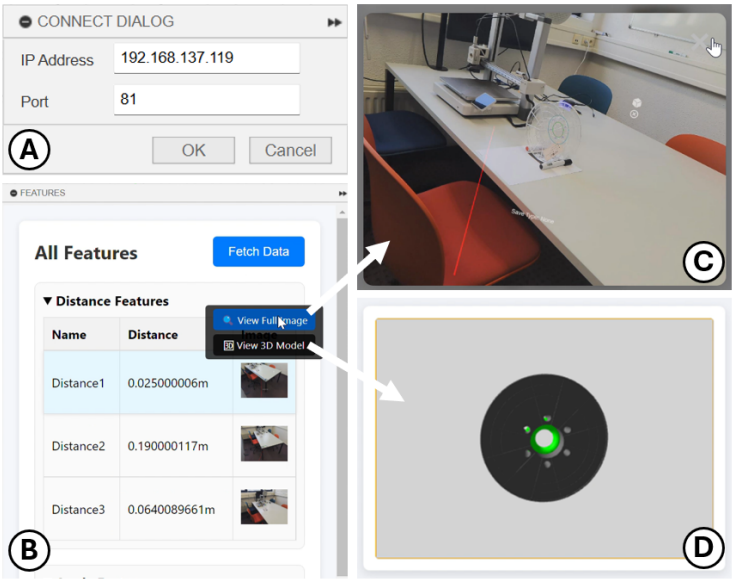

Figure 3.3: Interface palettes for (A) connecting to the Magic Leap and (B) fetching data. Once the data is retrieved, each feature provides contextual information through (C) an image or (D) a 3D model.

to calculate the perpendicular distance from that plane to the chosen point. The user selects the previously defined plane using the bumper, and then points to the opposite surface of the spool, confirming the selection again with the bumper. As a result, an edge representing the straight distance between the two sides is visualized in the environment (Figure 3.2C). This edge, like the previous features, can be stored using the familiar upward swipe gesture on the touchpad, ensuring the spool's width is accurately captured for the clamp design (Figure 3.2D).

In addition to the width measurement of the spool, the user may also want to capture the paths that represent both the outer and inner contours of the spool. These can be extracted in two different ways. To retrieve the inner path, the user makes use of the Slicing Tool. Similar to the previous tools, this also requires a plane to function. Therefore, the user first creates a new plane in the same way as before—by holding the trigger and selecting the Plane Tool from the menu. Once the plane has been created, it must be repositioned so that it lies in the middle of the spool. This is done by pointing at the plane with the controller, holding the trigger to grab it, and moving it into position (Figure 3.2E). With the plane properly aligned (Figure 3.2F), the user then activates the Slicing Tool, again by holding the trigger and selecting it from the menu. The tool requires the user to first select a plane and then select an object to be sliced. In this case, the repositioned plane is selected using the bumper (Figure 3.2G), followed by selecting the spool object in the same way. Once confirmed, the object is sliced, and the path representing the inner contour of the spool becomes visible (Figure 3.2H). The user then selects this path with the bumper, and before storing it, changes the save type from distance to path, ensuring that the feature is saved correctly as a contour.

Finally, to capture the outer contour of the spool, the user makes use of the Direct Tool, which is automatically activated upon selecting a surface. The user points with the controller to an accessible part of the upper side surface of the spool and selects it using the trigger (Figure 3.2I). Once the surface is selected, the visible edges are highlighted, allowing the user to identify the outer contour. This contour is then confirmed with the bumper (Figure 3.2J). As with the inner circle, the user changes the save type from distance to path before storing, ensuring the feature is correctly saved as a closed path.

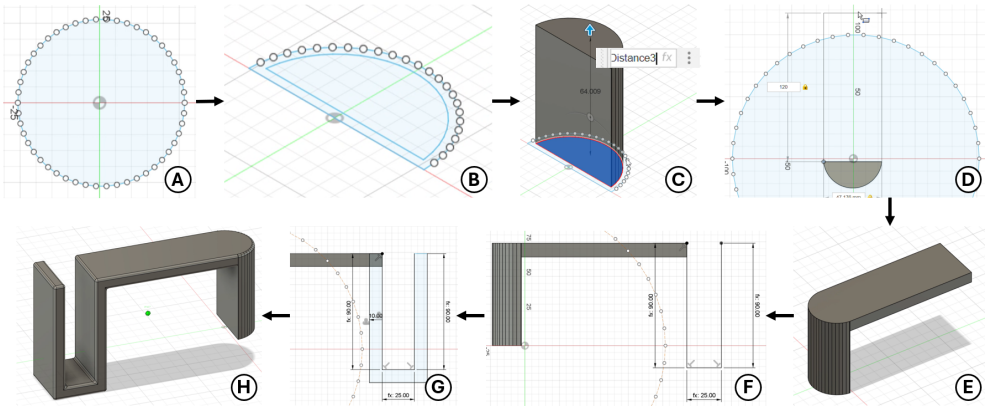

Figure 3.4: Step-by-step guide for modeling with the captured features: (A) Using the inner path feature. (B) Preprocessing. (C) Extruding using the width feature. (D) Defining the height of the clamp. (E) Extruding the height. (F) Shaping the rest of the clamp. (G) Preprocessing. (H) Final model using all the captured features.

3.1.3 Step 3: The Design Process

Now that the user has captured and stored all the necessary features, they can begin designing the clamp in Fusion360. Before the design process can start, however, the collected data must be exported from the Magic Leap system to Fusion. This is done using a custom-built plugin specifically developed for this workflow. To initiate the export, the user launches the plugin in Fusion360, which then connects to the Magic Leap device (Figure 3.3A) and fetches the stored data (Figure 3.3B). Once the retrieval is complete, the system automatically organizes the features into three distinct categories: distances, angles, and paths.

Each exported feature includes contextual information to help the user understand what it represents. The plugin provides two ways to access this context: either through a screenshot (Figure 3.3C) of the recorded feature or via a 3D model (Figure 3.3D) in which the feature is clearly highlighted. Both methods allow the user to quickly recall the purpose and location of each measurement. With all relevant data and visual references available, the user is fully prepared to begin modeling the clamp in Fusion360 with accuracy and confidence.

With all the necessary information at hand, the user can now start designing the actual clamp. The first step involves creating the holder for the spool. To do this, the user makes use of the previously saved inner path. By creating a new sketch and simply clicking on the stored feature corresponding to the inner path, the shape is automatically inserted into the sketch (Figure 3.4A). This ensures that the exact geometry is reused, providing a perfect fit for the real-world object. Once the path is added, some adjustments are made to the sketch (Figure 3.4B), after which it is extruded. During the extrusion process, the user references one of the saved distances—specifically, the width of the spool—to determine the correct extrusion depth. Because this measurement is stored as a named parameter, the user can simply enter its name, and the correct value is applied instantly (Figure 3.4C). This results in a highly precise model that seamlessly fits the physical spool.

With the spool holder complete, the next step is to design the clamp that will attach the entire structure to the table. The user begins by determining the required height of this clamp—ensuring that it will be tall enough to accommodate the full size of the spool without obstruction. To assist with this, the previously saved outer path feature is used. By clicking on this feature, the corresponding shape is automatically drawn into a new sketch, providing a clear visual reference for the minimum height required. This makes it easy for the user to design a clamp that extends slightly higher than the spool itself, guaranteeing a proper fit and allowing for some tolerance in the physical setup (Figure 3.4D). Finally, the shape is extruded to a thickness chosen by the user, forming the structural base of the clamp that will ultimately

be fixed to the tabletop (Figure 3.4E).

With the overall height defined, the user can now proceed to create the remaining part of the clamp. A new sketch is created, this time focusing on defining the depth of the clamp. As previously noted, a structural bar is located underneath the table surface, which imposes a constraint on how deep the clamp can extend. To avoid any collision with this bar, the user references the previously stored distance between the tabletop and the bar as a maximum limit. This value is then slightly reduced by a margin chosen by the user to ensure the clamp fits safely in the available space. Additionally, the thickness of the tabletop is also taken into account by referencing its stored value, which is entered by name to guarantee precision (Figure 3.4F).

Once this part of the geometry is completed, the overall shape of the clamp begins to take form. However, further refinements are necessary to finalize the design. One important consideration is to ensure that the body of the clamp does not interfere with the space required for the spool itself. To achieve this, the user overlays the previously saved outer path onto this new sketch. This visual reference helps define an appropriate boundary, ensuring that the clamp's wall thickness remains outside of the spool's operational area (Figure 3.4G). With all these constraints in place, the user completes the sketch and moves on to the final processing steps to produce a functional and spatially optimized clamp design (Figure 3.4H).

3.1.4 Recognizing new objects

As we saw during the walkthrough, features can easily be accessed by interacting with the objects and the controller. However, before such interaction becomes possible, a virtual model must first be available in the AR environment.

To make this possible, the user must first place the respective ArUco markers on the target objects within the real world. It is necessary to pay attention to every marker's orientation such that upon scanning, the respective digital object fits perfectly with its real-world counterpart. This correct alignment not only ensures technical correctness but also significantly improves the general user experience since it allows for smoother and more natural interaction with the upgraded environment. This alignment step is currently done manually, which suffices for this proof-of-concept. However, in future releases, this process could be automated through the use of object recognition technology or 3D scanners. Once all markers are placed in their correct positions, the user is then ready to use the system for accurate measurement tasks.

Upon system boot, the user must do a scan of all ArUco markers that were placed using the Magic Leap device. This allows the system to load and align all virtual models of the actual objects. Now that the virtual elements are correctly registered in the virtual world, the user can begin taking measurements and interacting with the system.

3.1.5 Interface Controls

To facilitate these interactions, a clear understanding of the interface controls is crucial. Therefore, before concluding the walkthrough, we will briefly summarize the button layout and explain which buttons are used for which features, as illustrated in Figure 3.5. This retrospective overview helps clarify the user interactions described in the previous steps and reinforces understanding of the workflow.

As shown in Table 3.1, the two main tools—direct and indirect—share the same core functionality. The next three tools—offset, slicing, and straight distance—also rely on a similar set of buttons to perform their actions. This consistency across tools makes the system intuitive and easy to learn, helping to reduce confusion for users as they begin working with it. With the button layout now clarified, the following section provides a short explanation of what each button does in the context of the different tools.

Figure 3.5: Mapping of the AR controller inputs of the Magic Leap

Tool	Ray	Bumper	Trigger	Touchpad	Menu
Direct Tool	✓	✓	✓	✓	✗
Indirect Tool	✓	✓	✓	✓	✗
Offset Planes	✓	✓	✓	✗	✗
Slicing Tool	✓	✓	✗	✗	✗
Straight Distance	✓	✓	✗	✗	✗
Axis Extraction	✗	✗	✗	✓	✗
Inventory	✓	✗	✓	✗	✓
Tool Menu	✗	✗	✓	✗	✗

Table 3.1: Mapping of Interface Buttons to System Tools

Ray

The ray functions as a pointing device in nearly all tools, allowing users to aim at or interact with visible objects and features. It essentially acts as the virtual equivalent of a computer mouse, making it a central part of the interaction process.

Bumper

The bumper button is generally used for making selections. It lets users indicate which edges or planes should be used within a given tool. For example, in both the direct and indirect tools, the bumper selects edges that will be saved to the inventory. In tools like offset, slicing, and straight distance, it helps carry out sequential selections—such as first choosing a plane, and then using the ray to click on a specific point on an object.

Trigger

The trigger is primarily used in the main tools to confirm surface selections. A user may first choose a surface with the trigger, then either use the bumper to select edges or pull the trigger again to switch to the indirect tool. The trigger is also involved in the offset planes tool, where it allows users to toggle through different modes for plane detection. Additionally, the trigger can be used to select a stored feature from the inventory and display it in the relevant context.

Touchpad

Because most of the physical buttons are already in use, the touchpad offers an alternative method for saving features. It recognizes specific gestures—for instance, a swipe-up motion triggers the system to store the currently selected edges as a feature in the inventory. Additionally, when a single edge is selected, performing a swipe-down gesture initiates axis extraction. This decomposes the edge into its X-, Y-, and Z-axis components, allowing users to isolate and store only the relevant features.

Menu

Finally, the menu button is used to open the inventory. Once it is open, users can employ the ray and trigger to select a feature and view it again in its appropriate context.

3.2 Features

Now that we have explored a complete walkthrough of the system in action—from setup and measurement to integration with Fusion360—it's important to take a step back and examine the building blocks that make this workflow possible. This section offers a detailed look at the individual tools and features that power the system. Each tool has been designed to support specific user needs, ranging from basic measurements to advanced spatial interactions. By understanding the role and functionality of these features, users can more confidently and effectively leverage the system in a variety of design scenarios.

3.2.1 Direct

The Direct Tool is one of the two fundamental components that will consistently be used within the system. It enables users to extract measurable features—such as distances, angles, or paths—directly from an object based solely on its visible geometry, without needing to reference other surfaces or separate objects. The second tool, known as Indirect, builds upon the foundation laid by the Direct Tool and will be discussed in a later section. Together, these tools form the core of the interaction model, but it is the Direct Tool that serves as the user's first and most frequent point of contact when gathering measurement data.

To begin the measurement process with the Direct Tool, the user initiates an interaction by selecting a surface on the target object using the controller. Ideally, this surface is one where the user expects a relevant feature—such as a useful edge—to be located. Once a surface is selected, the system provides visual feedback by highlighting it, allowing the user to visually confirm their choice. From there, the user can select specific edges of the surface by clicking on them. When the user hovers over an edge, it is highlighted in light blue to indicate that it can be selected. Once clicked, the edge turns green, confirming that it has been actively selected for measurement. This method allows for intuitive and visually guided feature selection, eliminating the need for complex referencing or alignment with other geometry. The user interacts directly with what they see, streamlining the process and making it accessible even to those with limited experience in CAD environments.

The type of measurement feature that gets stored depends on how many edges are selected:

- One edge results in a distance.
- Two edges define an angle.
- More than two edges are interpreted as a path composed of connected lines.

This automatic classification of measurements based on edge count helps reduce user error and simplifies the workflow, allowing the focus to remain on the design task rather than on manual configuration. Additionally, the user can choose which feature should be saved. They can either follow the system's suggestion or choose to store the selection as a path. Selecting

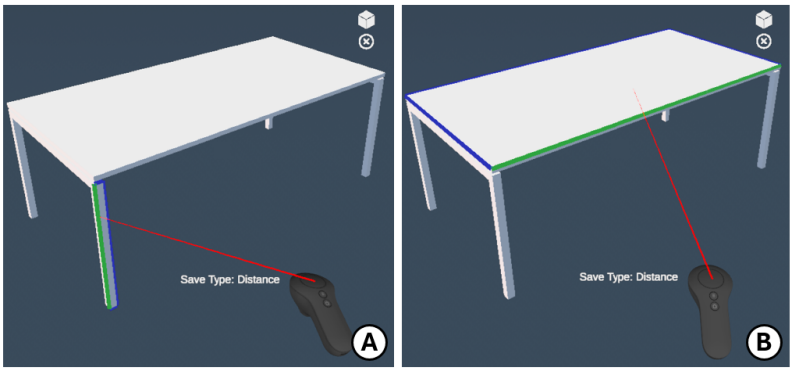

Figure 3.6: Illustration of the Direct Tool in use: (A) selecting the vertical edge to measure the height of a table leg, and (B) selecting a horizontal edge to determine the width of the tabletop.

an angle is not possible: if only one edge is selected, there is no angle, and if more than two edges are selected, all corresponding angles would have to be included—making the process inefficient.

To make this process more tangible, consider a scenario where a user wants to determine the height of a table leg. Using the controller, the user would begin by selecting a surface on the leg—typically a vertical face—and then select the vertical edge that spans from the floor to the underside of the tabletop (Figure 3.6A). This edge directly represents the height and can be stored as a distance. In another example, if the user wishes to measure the width of the tabletop, they would instead select a horizontal surface of the tabletop and then select the edge that runs across its full width (Figure 3.6B). Again, this results in a distance measurement, but the context and chosen geometry are different.

This system of context-sensitive edge interpretation not only makes the process more efficient but also reduces cognitive load for the user. Instead of requiring explicit selection of measurement types beforehand, the system interprets intent based on the user's actions—helping to make the tool both powerful and user-friendly. As such, the Direct Tool provides a flexible yet structured approach to geometry capture that fits seamlessly into the broader measurement workflow.

In addition to its role as a standalone measurement method, the Direct Tool is also tightly integrated into the functioning of other tools within the system. It often operates behind the scenes, triggered automatically as part of more complex interactions. For example, when a user engages with tools that analyze indirect relationships between objects, the system may first use the Direct Tool to establish base features. In this way, the Direct Tool underpins much of the system's capability, serving both as a manual utility and as a foundational layer for more advanced functionality.

3.2.2 Indirect

The Indirect Tool is the second of the system's core tools and is conceptually derived from the Direct Tool. While the Direct Tool allows users to extract measurements directly from a single surface, the Indirect Tool extends this functionality by enabling users to measure relationships between different surfaces—either on the same object or across multiple objects. This makes it possible to capture more complex spatial relationships that cannot be derived from a single surface alone, such as diagonals or inter-object distances.

The interaction begins similarly to the Direct Tool: the user selects a surface on the object using the controller, which the system then highlights. Instead of immediately selecting an edge, however, the user continues by selecting a second surface, either from the same object or

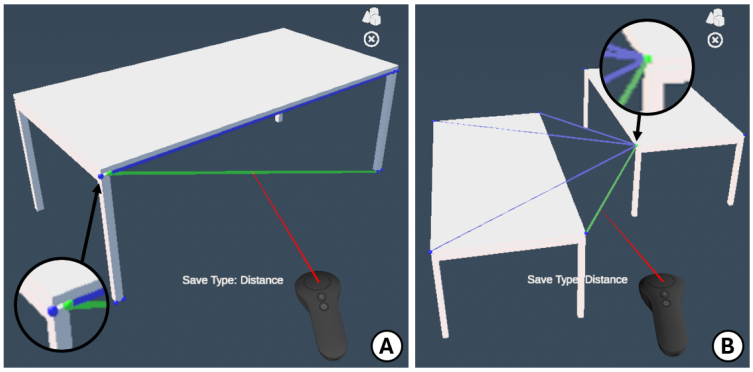

Figure 3.7: Illustration of the Indirect Tool in use: (A) selecting the diagonal edge between 2 table legs, and (B) selecting the distance between 2 vertices of different objects.

another nearby object. This action signals the system to activate the Indirect Tool.

Once both surfaces have been selected, the user chooses a vertex on the first surface. At that point, the system automatically visualizes all virtual Euclidean edges connecting that vertex to every vertex on the second selected surface. These represent all possible connections between the two surfaces, giving the user the freedom to select specific edges that define the desired measurement. This enables a broader range of measurements that go beyond the limits of a single face or object, while still maintaining an intuitive selection process.

To better understand how this works in practice, consider a scenario where a user needs to measure the diagonal between two table legs. The user would start by selecting a vertical face on the first leg, followed by a vertical face on the second leg. After choosing a vertex on the first face, the system displays possible connections to vertices on the second face. By selecting the appropriate connection, the diagonal can be quickly and accurately defined (Figure 3.7A). The same logic applies to measurements between separate objects—select two surfaces, choose a vertex, and select the resulting edge that represents the measurement (Figure 3.7B).

Just like with the Direct Tool, the type of feature that gets recorded depends on how many edges the user selects. A single selected edge is treated as a distance, two edges define an angle, and three or more edges form a path. However, it's important to note that, unlike in the Direct Tool, paths created using the Indirect Tool may span across different planes. Because these edges may connect vertices located in entirely different orientations, the resulting path does not necessarily lie flat in one direction. This introduces additional complexity in how paths are interpreted and visualized, a detail that is discussed further in the Implementation section.

The Indirect Tool plays a key role in expanding what users can measure and how they can interact with the geometry around them. It builds on the interaction model introduced by the Direct Tool but adds a layer of spatial flexibility that allows for more complex and meaningful design data to be collected. Whether measuring across components of a single structure or between entirely different objects, the Indirect Tool provides the precision and adaptability needed to support a wide range of design and analysis scenarios.

3.2.3 Offset Planes

Now that the two primary tools for extracting features—Direct and Indirect—have been introduced, we can explore a set of supportive tools that help users obtain features that are not immediately visible to the naked eye. These tools are essential for working with implicit or spatially abstract geometry that cannot be directly derived from visible edges or surfaces. One such tool is the Plane Offset Tool. While this tool does not store measurement features itself, it serves as an important intermediate step that enables other tools to function correctly.

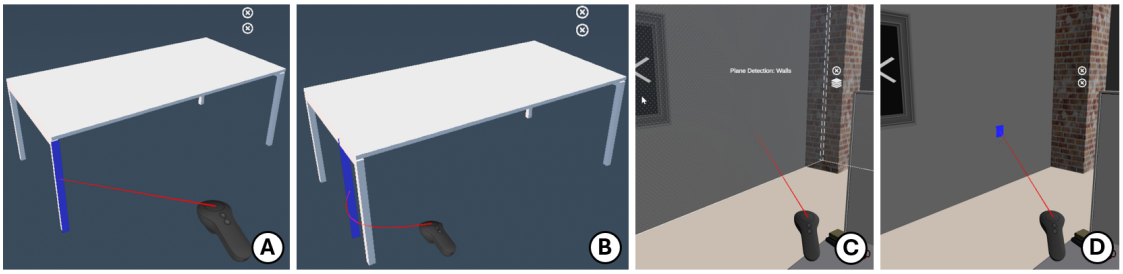

Figure 3.8: Illustration of the Offset Planes Tool in use: (A) adding a plane by selecting a surface of an object, (B) same as A but now it is moved a little, (C) scanning the room with plane detection, and (D) adding a plane by clicking on the detected wall of C.

Planes can be created in two different ways. The first method involves selecting a flat surface of an object. When the user selects such a surface, the system generates a plane with the same orientation as the surface (Figure 3.8A). However, this method is only valid for surfaces that are perfectly flat—it does not support curved or irregular geometry. This limitation ensures that the resulting plane maintains geometric consistency and can be used reliably in subsequent interactions.

The second method is based on plane detection, which is particularly useful in scenarios where reference geometry is not part of the object data—such as walls, floors, or ceilings. Since architectural elements like walls are typically not included in STEP files, the Plane Offset Tool provides a way to incorporate them into the measurement process. In this mode, the user can switch between different detection modes: None, Wall, Floor, and Ceiling. When one of the detection modes is selected, the system scans the surrounding environment and highlights the detected flat surfaces (Figure 3.8C). The user can then use the controller to click on one of these surfaces and generate a plane aligned to it (Figure 3.8D). This approach allows environmental elements to be integrated into the measurement workflow, greatly enhancing flexibility.

Regardless of how the plane is created, users also have the option to adjust the position of the plane (Figure 3.8B). This is particularly helpful when a reference plane is needed at a certain distance from the original surface rather than directly on it. The adjustment is restricted to the normal direction of the plane—ensuring that the plane retains its original orientation while being repositioned along its axis. This makes it possible to place planes in precisely controlled spatial locations, whether for aligning new measurements or aiding in visualization.

By enabling the creation and positioning of reference planes, the Plane Offset Tool bridges the gap between visible object features and abstract spatial reference frames. It acts as a flexible foundation for other tools that rely on spatial orientation or environmental context, without imposing rigid constraints on how or where geometry must exist. In this way, it provides users with the control and adaptability needed to navigate both object-based and spatially-aware measurement tasks.

3.2.4 Slicing

With the ability to create planes now established, these planes can be leveraged by the Slicing Tool. Up until this point, only features visible to the naked eye could be stored and measured. However, there are often important features hidden within objects—features that aren't visible from the outside. The Slicing Tool enables users to reveal internal structures by cutting through the object geometry. A clear example would be slicing through a table to expose hidden steel bars that provide support beneath the surface—elements that would otherwise remain invisible.

The inspiration for this tool comes from the concept of a 3D slicer, commonly used in additive manufacturing. Once a designer has created an object, a slicer breaks it down into thin,

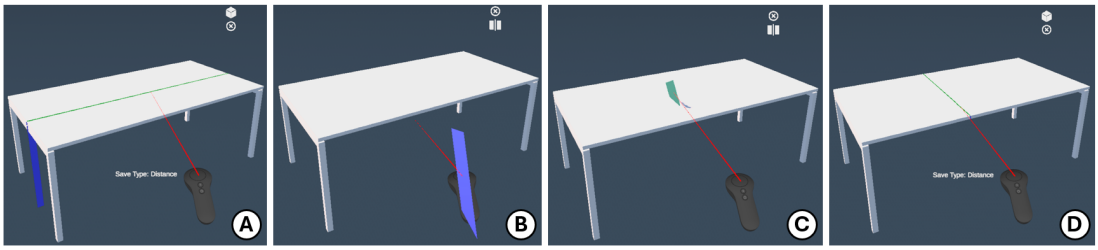

Figure 3.9: Illustration of the Slicing Tool in use: (A) slicing an object using a plane created via the Offset Plane Tool, (B) initiating a slice using the freely placed plane (virtual knife), (C) user positions the slicing plane manually, and (D) slicing the table based on the freely positioned plane.

horizontal layers. The designer can scroll through these layers to inspect the internal structure and assess whether it meets their expectations. Similarly, the Slicing Tool in this system provides a way to examine cross-sectional geometry, revealing new features and design details that may not be apparent from external surfaces alone.

To use the Slicing Tool, the user is provided with two main methods to define the slicing plane. The first method gives the user complete freedom to place a plane anywhere in space, at any orientation. In this mode, the controller is used almost like a virtual knife. The user moves and rotates it to define the slicing position. Once the user is satisfied, the plane can be confirmed and selected as a slicing tool, after which they choose the object they want to slice (Figure 3.9B-D).

The second method involves selecting a plane that was previously created using the Plane Offset Tool (Figure 3.9A). Because placing a freehand plane can be challenging to align precisely, the Plane Offset Tool provides structured, orthogonal planes that are accurately aligned with object or environmental surfaces. Instead of creating a new plane from scratch, the user simply selects one of these existing reference planes as the slicing surface. This ensures geometric accuracy and repeatability, especially when working with standardized or architectural elements.

Once an object has been sliced, the newly exposed edges become interactable, and the system immediately places the user in Direct Tool mode. This means the Direct Tool is automatically activated within the Slicing Tool context, allowing the user to select and extract features from the freshly revealed geometry. If desired, the user can then switch to Indirect Tool mode to capture more complex relationships between surfaces or objects. This seamless integration ensures that slicing not only reveals hidden features but also provides immediate access to the tools needed to work with them.

By enabling users to slice through geometry and inspect cross-sectional features, the Slicing Tool greatly expands the depth and flexibility of the system. It empowers designers to work not only with external form but also with the internal logic and structure of their objects—providing a richer, more comprehensive approach to measurement and analysis.

3.2.5 Straight Distance

In previous tools—particularly the Indirect Tool—measurements are derived by connecting vertices from one surface to those of another. However, this method relies heavily on the assumption that the objects being measured are properly aligned with each other. In real-world scenarios, this alignment is often not present, making it difficult or even impossible to retrieve a true perpendicular distance between two elements using vertex-to-vertex comparisons alone.

The Straight Distance Tool solves this problem by enabling the user to measure orthogonal distances between elements, regardless of their spatial alignment. Instead of requiring two

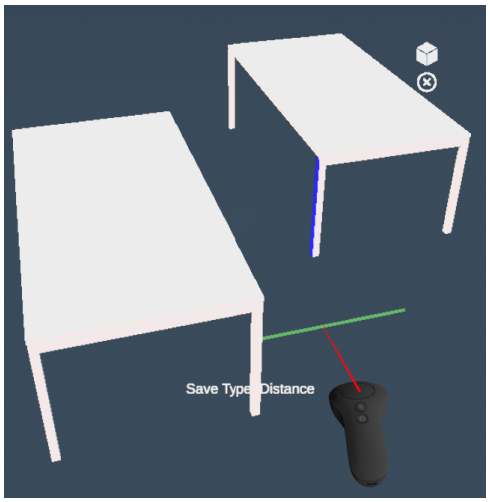

Figure 3.10: Illustration of the Straight Distance Tool in use: The user has selected the plane of the right table and a point on the table leg of the left table.

vertices to be aligned, this tool allows the user to select a plane and a single point, and then calculates the shortest possible distance from that point to the plane. This is achieved by projecting the point onto the selected plane, resulting in a straight edge that represents the perpendicular measurement.

To use the tool, the user must first define a reference plane. This is typically done using the Plane Offset Tool, which allows planes to be placed on any flat surface of an object—such as the side of a table or the face of a cabinet. Once the plane has been created and selected, the user can then pick a point on another object. The system immediately projects that point onto the plane and visualizes the connection between the original point and its projection, forming a clear and accurate straight-line distance.

As a practical example, imagine the user wants to know the distance between two tables that are placed unevenly in a room. By creating a plane on the vertical surface of one table and selecting a point on the second table, the tool projects that point back to the reference plane. This process generates a perpendicular edge that accurately represents the true distance between the two pieces of furniture, even if they are offset or rotated relative to one another (Figure 3.10).

This tool is especially useful in architectural or room-scale setups where objects are often measured relative to fixed elements like walls or floors. Since the Plane Offset Tool can also detect and generate planes on architectural surfaces—such as walls, floors, and ceilings—the Straight Distance Tool becomes a valuable asset for spatial analysis. For example, it allows a user to determine how far a cabinet stands from the wall behind it, even if the wall isn't part of the original 3D model.

Once a point is projected and a distance edge is created, the system automatically transitions into Direct Tool mode, just like with the Slicing Tool. This means the edge is stored as a feature that can be further used in the design workflow. If needed, the user can also manually switch back into Indirect mode to relate the projected vertex to other features, objects, or reference geometries.

3.2.6 Axis Extraction

The final tool we'll discuss is the axis extraction tool. As the name implies, this tool is designed to extract the X, Y, and Z axes from a selected feature within a 3D model. This tool was introduced not because the Indirect Tool is faulty, but because users often need precise, straight-

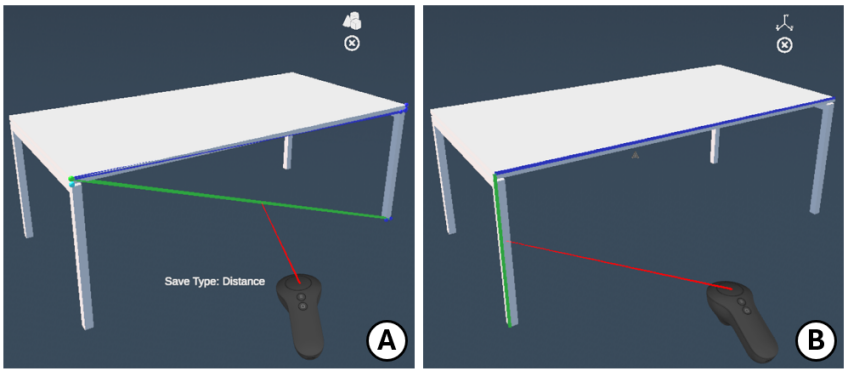

Figure 3.11: Illustration of the Axis Extraction Tool in use: (A) The user has selected a feature from the Indirect Tool, (B) where the Axis Extraction is applied to it.

line measurements aligned with the primary axes. The indirect approach will tend to form edges by connecting a chosen vertex to all other vertices, which can result in edges that are skewed or angled relative to the object's principal directions. While a good method for most purposes, this will not in general produce the clean, axis-aligned distances that users sometimes require for accurate modeling or analysis.

To address this, the axis extraction tool performs the reverse by isolating a single edge—one that consists of precisely two points that are on the same straight line—and attempts to decompose it into its X, Y, and Z components. Remember that the tool tries to extract these components, but cannot always extract all three if the edge is perfectly aligned along one or two axes. For example, if the two points are at the same height, the tool will not extract a Z-axis component. This selective extraction avoids the introduction of spurious or irrelevant data, maintaining the integrity of the resulting vectors.

Working with the tool is straightforward but requires some preparation. The user must select a valid feature, which can be created using any of the tools described in the previous subsections. The only requirement is that the selected feature must be a single edge with exactly two vertices so that it will be an individual straight line. Once this condition is met, the user performs a swipe-down gesture on the touchpad, and the system decomposes the feature into its X, Y, and Z axes. The user can then choose one of these axes to select and save for further operations. This operation facilitates easy and efficient extraction of clean, axis-aligned directions, which can be critical for precise measurement, alignment, or downstream model pipelines.

In practical scenarios, a user might initially select a slanted feature generated using the Indirect Tool (Figure 3.11A). While this feature may appear usable, its angled nature can lead to inaccurate distance predictions, potentially causing errors in the design phase. To correct this, the user can turn to the Axis Extraction tool by performing a swipe-down gesture. When this action is taken, the selected feature is decomposed into its components (Figure 3.11B), allowing the user to choose the axis-aligned feature that does represent the correct straight-line distance.

3.3 Design Environment

Now that we've discussed all the features included in the system, this section takes a closer look at how geometric features are stored and subsequently used within a CAD environment. This integration is made possible by exporting the relevant data directly into the CAD system.

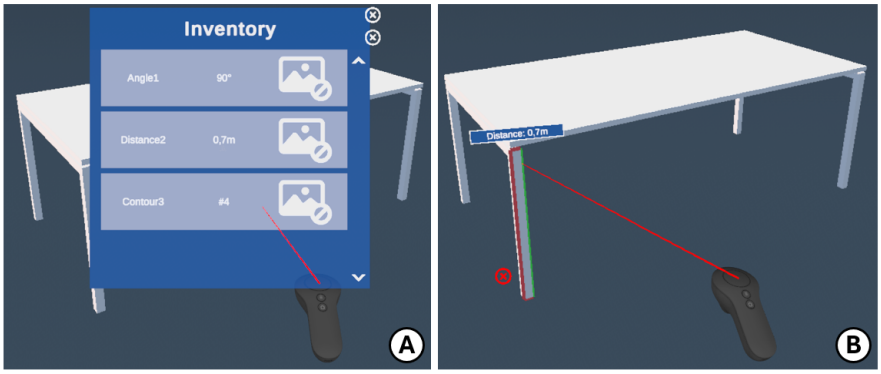

Figure 3.12: Illustration of the Inventory in use: (A) The general look of the inventory, and (B) the view when a feature is selected and the controller is hovering over an edge.

3.3.1 Inventory

Using the tools described earlier, users are now able to store various geometric features. These features can include distances, angles, or paths, depending on how the user interacts with the objects. Once the user selects one or more edges using either the Direct or Indirect Tool, the system automatically determines the type of feature to store based on the number of selected edges. All saved features are collected and displayed in the user's personal inventory.

The inventory provides an organized overview of every saved feature, presenting key information such as the feature's name, its value (in meters for distances, degrees for angles, or the number of edges in a path), as well as a snapshot to give visual context (Figure 3.12A). While the image helps to identify where and how the feature was created, it may not always offer a clear enough view—especially for more complex or less visible selections. For that reason, users have the option to revisit any saved feature by simply clicking on it in the inventory list.

When a feature is selected for review, all active tools are temporarily disabled, and the system highlights the geometry involved in the selected feature. This allows users to fully understand where the feature was taken from and how it fits into the overall model. It's a helpful way to verify what has already been captured, and to identify which measurements still need to be made.

While inspecting a saved feature, users can also hover over individual edges using their controller (Figure 3.12B). Doing so reveals the exact length of each edge, making it easier to interpret complex paths or verify specific segments within a feature. In this way, the inventory becomes more than just a list—it acts as a powerful reference tool for navigating, validating, and re-engaging with saved measurements during the design workflow.

3.3.2 Fusion360

While the system allows users to extract and store measurements directly in an internal inventory, the ultimate goal is often to use these features in a CAD environment for further design work. To support this workflow, the system includes an export function that allows users to transfer stored features directly to Fusion360. As discussed in the related work section, manually transferring measurements can be error-prone and time-consuming, which is why a custom Fusion360 plugin has been developed to automate the entire process and reduce human error.

This Fusion360 plugin is designed to seamlessly retrieve the stored data from the system and organize it into three distinct categories: distances, angles, and paths. Each feature is displayed as a separate row within the appropriate table, including the same core details shown in the Magic Leap inventory—such as the feature's name, numerical value (e.g., in meters or degrees),

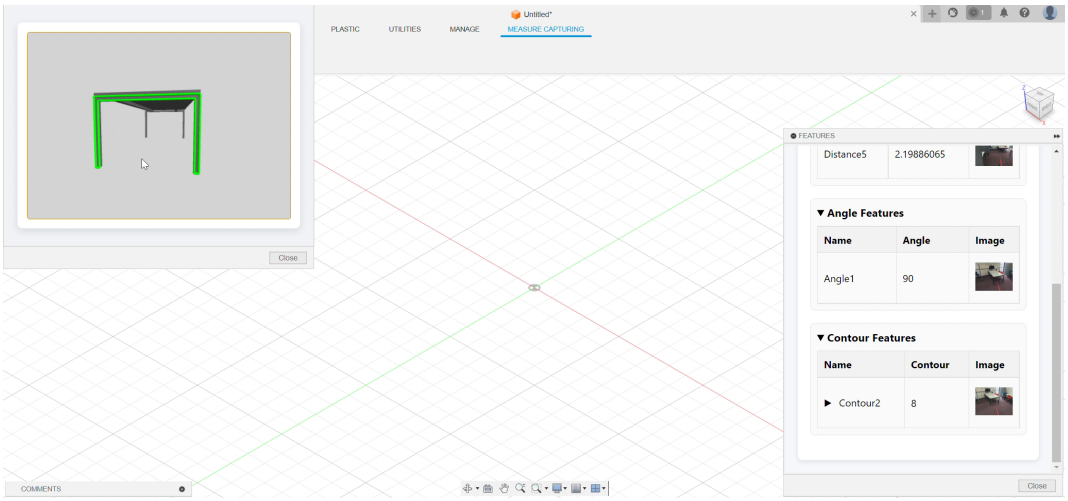

Figure 3.13: Illustration of Fusion360 in use: The general look of the Fusion360 plugin.

and a visual snapshot (Figure 3.13). This structured overview ensures that users can quickly identify relevant features and incorporate them into their ongoing design workflow without confusion.

Once the features are imported, they are fully integrated into the Fusion360 environment. All distances and angles are automatically defined as user parameters, which the designer can reference throughout their modeling process. This means that if a dimension needs to be reused or modified, it can be done consistently and centrally via parameter control, enhancing both precision and flexibility.

For path-based features, the plugin offers a different kind of functionality. When a user selects a specific path from the list, it is automatically sketched into the active workspace, eliminating the need to manually redraw complex contours or reference them from static images. This feature enables users to quickly continue their design work based on real-world measurements, accelerating the transition from concept to CAD.

Beyond the data itself, the plugin also enhances visual context. In addition to the still image shown for each feature, users can also view a fully rotatable 3D representation of the selected object and the feature applied to it (Figure 3.13 left upper corner). This makes it easier to recall exactly where a measurement was taken and how it relates to the geometry of the object, especially when returning to a project after some time.

Lastly, for stored paths, the plugin automatically decomposes the contour into its individual segments, giving designers the ability to extract and reuse sub-measurements directly. This can be especially helpful when a specific edge length was not explicitly stored but is part of a larger contour. If a required measurement is missing entirely, users can simply capture it again in the Magic Leap interface and refresh the Fusion360 view to keep their design process moving forward.

3.4 Practical Examples

Now that we are familiar with some of the core functionalities of the system, we will demonstrate a few additional real-world examples to highlight its versatility and power. Unlike the previous walkthrough, these examples will not follow a step-by-step format. Instead, they will focus on how the system's capabilities can be applied creatively to solve practical design challenges.

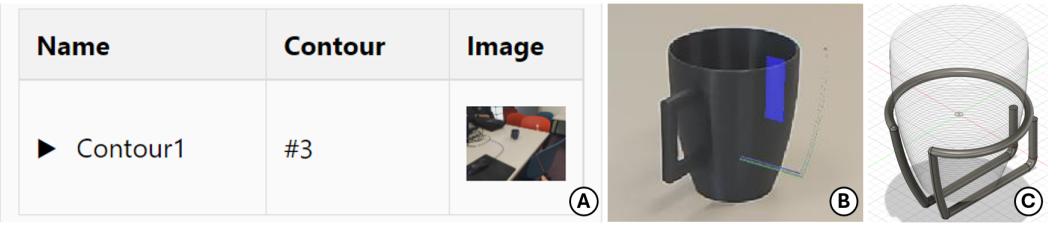

Figure 3.14: Creating a cup holder: (A) The captured feature. (B) The visible feature represented as a path. (C) The final result, where the captured feature was used for the cup model, and the holder was designed around it.

3.4.1 Cup Holder

In this first example, the user wants to create a custom cup holder tailored specifically to a particular mug. To begin, the user needs to capture the shape of the mug. However, since the mug is a round object and does not have clearly defined edges, it is not possible to directly select edges for storing a path. To work around this, the user starts by using the Slicing Tool. The first step is to create a plane using the Plane Tool. In this case, the plane is initially placed at the handle of the mug. The user then moves the plane to the center of the mug to ensure an accurate slice. Alternatively, the plane could also be created against a nearby wall or surface and repositioned afterward. Once the plane is correctly positioned, the user selects it with the Slicing Tool and then selects the mug object. This triggers the system to slice the mug, revealing the edges that represent its cross-sectional shape (Figure 3.14B).

These visualized edges can now be selected and saved by the user as a path feature in the inventory. Once this path is stored, the user has all the necessary data to proceed with designing the cup holder (Figure 3.14A). The next step is to switch to Fusion360, where the stored features can be imported and used to begin the actual modeling process.

To begin designing the cup holder, the user first visualizes the stored path within a sketch by simply clicking on the saved feature. Once the path is visible, the user selects half of the sketch to perform a revolve operation. This action generates the full 3D shape of the mug, which can now serve as a basis for designing a holder that fits around it (Figure 3.14C). In this way, the saved feature acts as a reference, allowing the user to recreate the object's shape without requiring a full 3D scan. By using this workflow, users can capture the form of various objects in Fusion360 without needing to perform complex steps or rely on expensive scanning equipment. Working with a digital representation of the object directly in Fusion enables the user to model around the visualized sketch or shape with ease, offering a powerful and accessible design solution.

3.4.2 Back Table Extension for Extra Workspace

A second example involves the user designing an additional table to serve as extra workspace. In this case, the available space in the physical environment must be taken into account, as the new table will be placed between a wall and an existing table. Since the wall cannot be treated as a standard object—due to the fact that walls vary greatly in size and structure—it needs to be recognized dynamically by the Magic Leap system. This is achieved using Plane Detection, a feature that allows the Magic Leap to identify flat surfaces in the real world by analyzing visual and spatial data from its environment. The system detects planes by mapping consistent, continuous surfaces and recognizing patterns in the spatial mesh, making it possible to define boundaries and available space without predefined models.

Now that we know walls and other flat surfaces can be recognized, we still need to gather a few important measurements. The most crucial dimensions for creating this additional table are the distance between the wall and the existing table, the height of the existing table, and finally the width of the table. Once these three measurements have been obtained, we can design a

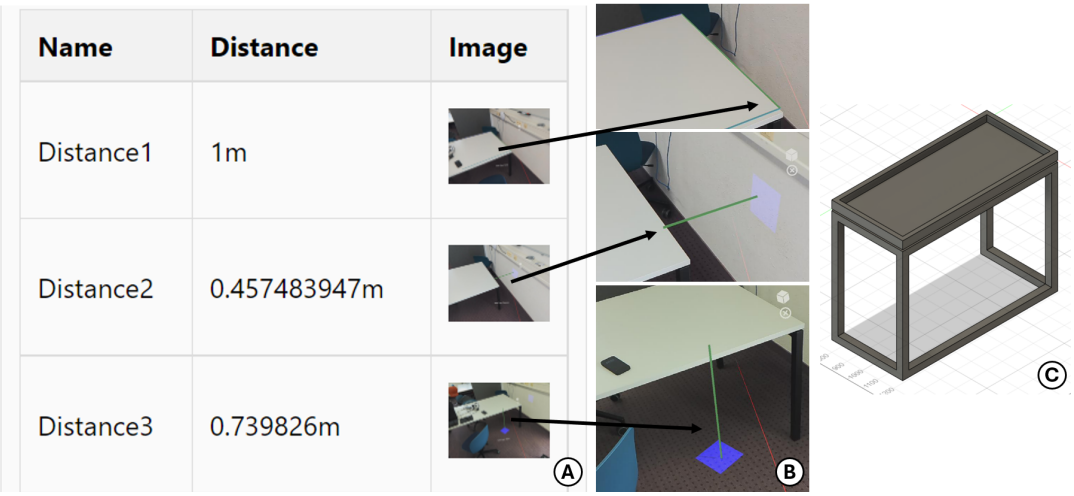

Figure 3.15: Creating the extra table: (A) The captured features. (B) The visible features represented as distances. (C) The final result.

new table that matches the height and width of the current one but fits perfectly in the space between the wall and the existing table (Figure 3.15C).

The width of the existing table can be easily obtained by using the Direct Tool and selecting the tabletop as the reference surface (Figure 3.15B, first). To determine the height, several methods are available. However, to showcase the system's capabilities, the surrounding environment is used instead. This is done through the Offset Plane Tool. When activating this tool, the user can press the trigger to enable Plane Detection. As a result, the user sees the planes detected by the Magic Leap, and by clicking on one of these detected planes with the bumper, a new plane is created that inherits the same properties. This newly created plane can then be used in combination with the Straight Distance Tool to accurately measure the height. Additionally, this generated plane can be used throughout the system with other tools—just like any plane created on a flat surface—allowing for flexible and consistent integration in the measurement and design workflow.

To obtain the height of the table, Plane Detection is used to create a plane on the floor. This plane can then be used with the Straight Distance Tool to measure the table's height. The measurement is taken by first selecting the generated plane and then selecting the top surface of the table. In doing so, the system calculates the straight distance from the floor to the top of the table (Figure 3.15B, last). The same method can be used to determine the distance between the wall and the table. In this case, however, Plane Detection is used to generate a plane on the wall, which is then combined with the side surface of the table to measure the straight distance between the two (Figure 3.15B, middle).

Lastly, the obtained features can then be used within Fusion to streamline the modeling process. Since all the saved features represent specific distance values, they can be directly referenced in Fusion by entering the name of the feature, which automatically retrieves its corresponding value (Figure 3.15A). This makes it easy to reuse precise measurements without the need to manually input or recalculate them. These features can be integrated into sketches, ensuring accurate dimensions, and can also be applied in modeling operations such as extrusions, fillets, and other parametric functions, increasing both efficiency and consistency in the design workflow.

3.4.3 Phone Holder

As a final example, a real-world scenario is presented where the user wants to design a phone holder based on the design of their existing iPad holder. In this case, it becomes important to accurately capture and reuse the specific angle from the original object. By using the

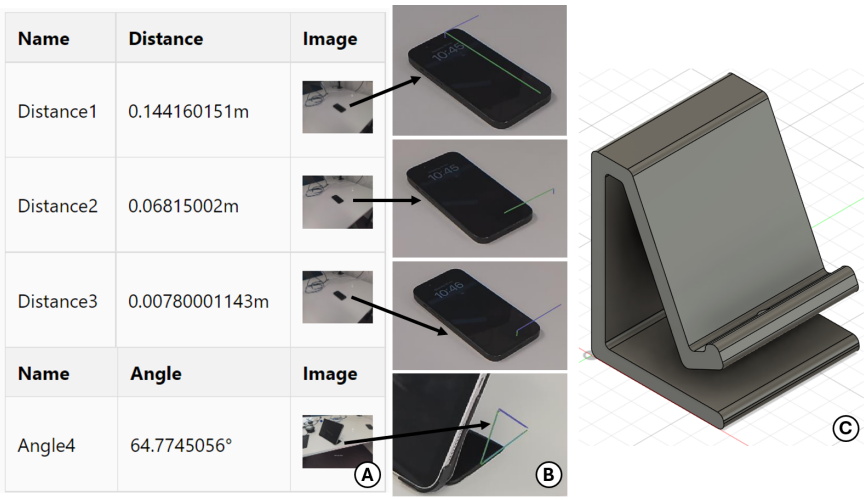

Figure 3.16: Creating a phone holder: (A) The captured features. (B) The visible features represented as distances and one angle. (C) The final result.

standard functionality of the system, angles can be saved as features when exactly two edges are selected—allowing the user to replicate the precise inclination used in the iPad holder for the new phone holder design.

The most important features needed for designing the phone holder are the dimensions of the phone itself—namely, its width, length, and thickness. These measurements can be obtained in various ways, but those methods are not discussed further here. Another key feature is the angle of the iPad holder, which can be captured using the Direct Tool. By pointing at the side of the holder and pressing the trigger, the user can visualize the edges of the model. Then, by selecting the two edges that form the desired angle (figure 3.16B, last), and saving the feature, the angle is stored for later use.

For the design of the phone holder itself, an existing model with user parameters can be used as a base (Figure 3.16C). Once the features have been retrieved from the Magic Leap and imported into Fusion (Figure 3.16A), the user parameters of the base model can be updated with the values obtained from the Magic Leap. This eliminates the need for the user to design a new model from scratch, enabling them to adapt existing models using the saved features quickly and efficiently.

As we've seen in the previous examples, the system can be used in powerful and flexible ways. For instance, users can quickly and easily reconstruct symmetrical shapes that serve as useful references in the design process. This allows them to build around these shapes with greater ease and precision.In addition, the system can take the user's surrounding environment into account during measurement. This means the entire space can be used as a reference to gather measurements, making it possible—at least in theory—to reconstruct the shape of a room. Finally, we've seen how measurements from multiple objects can be combined into a single final design. This removes limitations for the user, giving them full creative freedom. It also means that it's always possible to start from a basic parametric model and simply update it using the captured measurements, allowing for a fast and efficient design workflow without needing to build everything from scratch.

Chapter 4

Implementation

In this chapter, we will take a closer look at the technical aspects of the system. We will focus primarily on the hardware that is being used, as well as the overall structure of the system in terms of software. This includes how different software components are integrated and how they communicate with each other.

4.1 Hardware

For the development of this system, the Magic Leap (Figure 4.1) was chosen as the primary hardware platform. This decision was based on a combination of technical requirements, development flexibility, and practical usability. Since the goal was to build an Augmented Reality (AR) system that enables users to interact with 3D features in real-world space, selecting the right headset was a key factor in achieving a reliable and intuitive user experience.

One of the key advantages of the Magic Leap is its untethered design. Unlike many other AR or VR devices that require a connection to a high-performance computer, the Magic Leap is self-contained. This allows for a much more portable and flexible experience, making it well-suited for use in diverse environments such as workshops, offices, or field applications. Users are not restricted by cables or the proximity of a workstation, which contributes significantly to the immersive experience.

Another important benefit is the headset's access to onboard cameras and sensors. This access is crucial for capturing the spatial and visual context of features selected in the real world. For this system, we needed the ability to gather environmental data to anchor features accurately within a given context. Many competing devices, like the Meta Quest or earlier versions of the HoloLens, either limit or fully restrict access to this type of data, which would have severely limited the system's capabilities. The Magic Leap, in contrast, provides developers with the necessary tools to work with the physical environment at a detailed level.

In addition, the Magic Leap supports the use of ArUco markers—a lightweight and widely-used method for spatial tracking and identification in Augmented Reality. This functionality made it possible to precisely detect and register specific positions or reference objects within the physical world. It can be especially useful for tasks that require repeatable and accurate positioning or for linking virtual features to physical markers in dynamic settings.

The device also includes support for advanced interaction techniques such as hand tracking, which—although not the primary interaction method used in this prototype—opens up promising directions for future development. Instead of relying solely on controllers, users could eventually interact directly with objects using hand gestures, creating a more natural and seamless interface. The presence of these capabilities ensures that the system remains extensible and adaptable as new use cases arise.

Figure 4.1: The Magic Leap headset.

Finally, when designing an AR system, the choice of visual delivery method plays a major role. In general, there are two types: video pass-through, where the user views the real world through a camera feed, and optical see-through, where digital content is projected onto transparent lenses that allow direct sight of the real world. The Magic Leap uses optical see-through, which maintains true-to-life depth perception and minimizes visual lag, making it ideal for precision interactions with virtual content aligned to physical surfaces.

Throughout the development process, no major issues were encountered with the Magic Leap. It provided a stable and capable platform that met all necessary hardware and software requirements for building a context-aware AR system. Its support for environmental sensing, marker detection, and flexible input methods made it a strong foundation for both current functionality and future extensions.

4.2 Software

4.2.1 Introduction

Now that the choice of headset has been explained, along with the reasoning behind it, the next step was to begin the actual implementation of the system. For this project, the decision was made to build the MR application using Unity. This engine was chosen because it offers robust support for virtual and augmented reality development and is widely used within the XR community. One of Unity's major advantages in this context was its integration with the Magic Leap Hub, which allowed for development and testing without constantly needing to wear the headset.

Thanks to the Magic Leap simulator provided through the Hub, it was possible to test positioning, interface behavior, and certain environmental interactions directly from the computer. This significantly improved development efficiency and enabled iterative testing in a more comfortable and controlled environment. Being able to simulate inputs and environments made debugging faster and allowed for rapid prototyping, which is essential in early stages of application development.

However, Unity does have some limitations, particularly when it comes to interacting with complex 3D geometry. While it excels in general scene building and rendering, it lacks native support for extracting or analyzing geometric features from 3D models. For instance, identifying specific edges, surfaces, or planes from a mesh is not straightforward unless custom shaders or workarounds are used. These methods can be time-consuming to implement and are not always

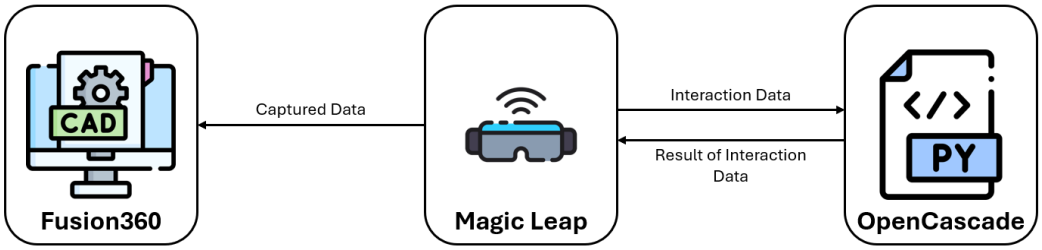

Figure 4.2: The full integration of the whole VR application.

reliable or scalable.

To address this challenge, the project made use of OpenCascade, an open-source CAD modeling library designed for high-level geometric operations. OpenCascade is especially powerful when working with boundary representation (B-Rep) models, as it allows for precise access to individual geometric elements such as faces, edges, and vertices. This level of control is crucial for an application that needs to allow users to select and manipulate specific features of objects in 3D space.

By integrating OpenCascade into the system, it became possible to analyze 3D models in a more meaningful way, enabling the detection of planes, slicing of objects, and even measuring distances—all based on real geometric properties rather than just visual meshes. For the purposes of this application, OpenCascade was essential in bridging the gap between raw 3D models and the feature-level interactions that users would perform through the AR interface.

Finally, a Fusion360 plugin was also developed to enable the transfer of saved data from the Magic Leap directly into Fusion360. This integration allows users to seamlessly move from capturing data in the AR environment to working with that data in a professional CAD setting. By doing so, users can immediately begin designing or refining their models in Fusion360 based on the features they identified and recorded within the MR application. This direct connection between the two systems enhances workflow efficiency and supports a smoother transition from spatial interaction to precise digital modeling.

4.2.2 Integration

While OpenCascade offered the necessary geometric capabilities to support precise feature-level interactions, integrating it into the Unity environment was not straightforward. The two systems operate in different programming ecosystems—Unity is based on C#, while OpenCascade is written in C++. This posed several technical challenges that had to be addressed in order to enable seamless communication between them.

One common approach to bridging such systems is by creating a custom C# wrapper. This wrapper acts as a translator between Unity and OpenCascade, exposing the C++ functions of the OpenCascade library in a way that Unity's C# environment can understand and use. However, developing such a wrapper is a complex and time-intensive process. It requires deep knowledge of both languages, careful memory management, and constant maintenance to ensure compatibility with both environments.

While technically possible, writing such a wrapper would have been an extremely time-consuming task and fell outside the main focus of my thesis. It would have added a great deal of overhead without offering any real benefit to the goal of the project. That's why I chose a different and more efficient approach: using sockets to handle the communication between Unity and OpenCascade.

Sockets are a method of enabling communication between different programs, even if they're written in different programming languages or running on different devices. Think of it as a

digital phone line—one application sends a message, the other receives it, processes it, and replies. In this case, since OpenCascade was already being used through Python, I set up a Python-based socket server on my computer.

Here's how the system works: whenever a user performs an action in the Magic Leap environment, that action is sent from Unity to the socket server. The server then processes the request using OpenCascade and performs the necessary geometric computation. Once the result is ready, it's sent back through the socket to the Magic Leap, where Unity takes over again to visualize or use the result in the MR application.

This approach allowed the two separate systems—Unity and OpenCascade—to communicate seamlessly, even though they are based on completely different technologies. It also kept the architecture modular and made it easier to debug or extend individual components in the future.

As previously mentioned, a Fusion360 plugin was also developed, and this plugin similarly relies on socket-based communication to receive data from the Magic Leap. In this setup, the Magic Leap acts as the socket server, while the plugin itself functions as the client. This configuration makes it possible for multiple Fusion360 applications—potentially operated by different users—to connect to the Magic Leap and work with the captured data simultaneously. This opens the door to collaborative design workflows, where several designers can begin creating or modifying models based on the same spatial information collected in AR. The plugin itself was also implemented in Python, which further streamlined the integration process.

Finally, as shown in Figure 4.2, the entire system integration can be visualized. At the core of this architecture is the Magic Leap, which serves as the central point of interaction. From there, data can either be sent directly to Fusion360 for use in further design processes, or routed through the socket server to OpenCascade. In the latter case, the transmitted data—captured through user interaction within the Magic Leap—is processed by OpenCascade to perform geometric operations or computations. Once a result is obtained, it is sent back to the Magic Leap, where Unity handles its visualization and integration within the mixed reality environment. This flow highlights how each component communicates seamlessly to support a responsive and modular design experience.

4.2.3 Interaction Objects

With the system architecture in place and communication between the core components established, the next important consideration is how users actually interact with the content in the mixed reality environment. As mentioned earlier, this is a proof-of-concept application, and for the time being, it relies on pre-existing STEP files rather than scanned models. This choice was made for two important reasons. First, current 3D scanning technologies are not yet accurate enough to capture precise geometries, especially when fine details are involved. This could lead to unreliable interactions or misinterpretations of object features. Second, expecting users to perform a full and accurate scan of each object in their environment introduces a significant usability barrier. In real-world situations, it's not always practical—or even possible—for users to scan every surface or hidden feature, particularly with smaller or more complex items.

As a result, the assumption is made that, in future iterations of this system, high-resolution 3D scans will already be available for the relevant objects. These scans could come from manufacturers, databases, or be generated ahead of time using more controlled and precise methods. For now, to simulate this behavior and ensure consistency during testing and development, we load the STEP files manually and place them within the AR environment using ArUco markers. This approach allows for a stable and predictable setup, where each marker corresponds to a specific 3D model. The markers not only identify which model to load, but also determine the object's initial position and orientation in the space. While it may require some manual alignment between the marker and the physical object, this process is usually quick and easy to perform. Additionally, using markers ensures a more robust and reproducible interaction

setup—an important consideration when validating the technical functionality of the system. In the future, this process could be further streamlined by integrating object recognition algorithms, removing the need for markers entirely while preserving spatial accuracy.

4.2.4 Tool-Based Algorithms

Now that the integration process has been fully explained, we will take a closer look at how specific tools actually work and where OpenCascade plays a key role. To do this, we'll walk through the underlying algorithms and clarify which tools rely on which processes. This will give a better understanding of what each tool is really doing behind the scenes.

Clicked Point

A commonly used algorithm is triggered when a specific point is clicked within the MR environment. The system first checks whether the clicked point lies on an object that was loaded using an ArUco marker. If that's the case, key data about the interaction point and the associated model is sent to OpenCascade to run the corresponding algorithm. This may include whether a reference plane needs to be retrieved.

When OpenCascade receives coordinate data from the socket server, it first needs to adjust this data to match its own coordinate system. Unity, which sends the data, uses a left-handed coordinate system where the X-axis points to the right, the Y-axis points upward, and the Z-axis points forward (out of the screen). OpenCascade, in contrast, uses a right-handed coordinate system, where the X- and Y-axes have the same orientation, but the Z-axis points in the opposite direction — backward (into the screen). The structural difference lies in the spatial relationship between the axes: in a left-handed system, the Z-axis forms a mirrored orientation compared to the right-handed system. This distinction directly impacts how 3D objects are positioned and rotated.

To ensure consistent interpretation of spatial data between the two systems, a coordinate transformation is required. This is typically done by inverting one axis — commonly the X-axis — which effectively mirrors the coordinate system and changes it from left-handed to right-handed. Additionally, because Unity uses meters and OpenCascade operates in millimeters, all coordinates are scaled by a factor of 1000. This combined transformation preserves the correct scale, position, and orientation of 3D models, avoiding issues such as flipped geometry or misaligned rotations during data transfer.

After this conversion, we can use the transformed parameters to load the STEP file—whose name was provided—into OpenCascade, and apply the correct translation and rotation. This ensures that the object appears in the exact same location in OpenCascade as it does in Unity. Next, we identify the surface that lies closest to the clicked point. To do this, we use the OpenCascade function TopExp_Explorer, which lets us retrieve all surfaces of the object. We then calculate the distance from each surface to the clicked point and keep the one that is closest.

Once we've identified the correct surface, we use a similar approach to retrieve all the edges of that surface. These edges are then analyzed, and the precise points that form those edges are converted to Unity's coordinate system, stored in JSON format, and sent back to Unity. Unity then uses these points to visualize the correspondign edges.

This algorithm is primarily used for the Direct Tool, where the user can click on a surface and the corresponding edges are displayed, allowing the user to make a selection. It is also used when the user clicks on a second surface that's different from the first, which is part of the Indirect Tool. However, when a second surface is clicked, not only are the edges of that surface displayed, but also the vertices of the first surface. These vertices are automatically retrieved from OpenCascade, since they are already needed to visualize the edges.

The remaining logic for the Indirect Tool is handled within Fusion360 itself. It works by

identifying which vertex is closest to the ray, and that vertex is then highlighted. The user can then press the bumper to select a vertex, which will automatically connect to every vertex on the second surface by creating edges.

The algorithm is also used when creating a plane. This process follows the same steps as before, but with an additional check to determine whether a plane needs to be created. If that's the case, essential information about the selected surface is sent back to Unity in JSON format—without the edges. This allows Unity to generate a plane based on the received information.

Slice Object

The slicing algorithm also relies on the user clicking a point. Before the user can actually use the Slicing Tool, a plane must first be defined—unless the user chooses to use the virtual knife. In cases where a plane is needed, the user will first use the Offset Plane Tool, which expects the user to click on a point on an object. When that happens, the same algorithm described in the previous section is executed.

Once a plane has been defined, the user can proceed with the Slicing Tool. At this point, all necessary data is gathered before being sent over the socket server. The user first selects the plane that will be used for slicing, followed by the object to be sliced. Relevant information about the selected object and slicing plane is then sent to OpenCascade for processing.

OpenCascade begins by converting from a left-handed to a right-handed coordinate system, and then applies the correct translation and rotation to the provided STEP model. This ensures the object appears in the same location as it does in Unity. Next, the plane's position and normal vector are used to perform the slicing operation, which is done using the BRepAlgoAPI_Section command. The result of this operation is a set of vertices and edges, which are then converted to Unity's coordinate system, analyzed, and sent back to Unity in JSON format.

Once Unity receives the result, a preprocessing step is performed to visualize the detected edges. At that point, the user is automatically returned to the Direct Tool, where they can select and optionally save the edges.

Straight Distance

For the Straight Distance tool, a plane must first be created using the Plane Offset tool. In this case, the clicked point algorithm is triggered once again. Once a plane is available, the necessary data can be collected to enable the tool. The user first selects a plane and then clicks a point. Relevant information about the selected plane and the clicked point is then sent to proceed with the operation.

Once OpenCascade receives this data, it calculates the orthogonal projection to determine the second point, which then allows the distance between the two points to be computed. All this information is analyzed and formatted into a JSON object, which is then sent back to the Unity. In this case, there's no need to convert between left-handed and right-handed coordinate systems, since we're working directly with point coordinates. When Unity receives the data, it processes and visualizes it in the environment. At that point, the user is once again working within the Direct Tool, allowing further interaction.

4.2.5 Axis Extraction

For the final tool, OpenCascade is not used and instead everything is calculated directly within Unity. When a user selects an edge in either the Direct or Indirect tool, that edge can be broken down into its three components. This is done by first checking if the selected edge is compatible. In this case, that means the edge must be a straight line consisting of only a start and an end point. If this condition is met, the direction vector of the line is determined and then split into the three local axes (X, Y, and Z) of the selected object. These components are

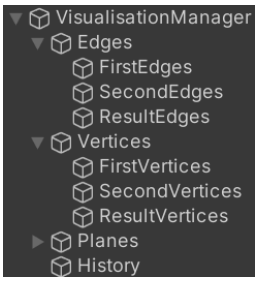

Figure 4.3: The Unity structure that is used to keep the visible edges to show the user.

visualized by projecting the edge onto each of the axes, giving the user a clear understanding of the individual directional components of the selected edge.

4.2.6 General Implementation

Now that we've discussed how each tool works and how OpenCascade plays a role in them, let's talk about the overall system implementation. As we already know, OpenCascade is primarily used when specific tools are activated by the user. The rest of the system's implementation is mainly handled within Unity. This was done by creating several managers, each responsible for a specific part of the implementation. These managers are essentially C# scripts that continuously monitor the system's current state.

There are two key statuses that are constantly tracked:

- The first tracks which main tool is currently active, since this determines which features can be saved. It also tracks whether the user is viewing a history or not. As a result, the user can be in one of five states: None, Direct, Indirect, Axis Extraction or History.
- The second status tracks the specific tool being used at any given moment, with the exception of Direct, Indirect and Axis Extraction, which are tracked in the first status. Here as well, there are four possibilities: Cancel, Offset Plane, Slicing, and Straight Distance.

By using these statuses, each script can independently execute its corresponding functionality based on the current state. For example, the ControllerInput Manager constantly checks the system's status to determine which action should be performed when a button is pressed. As we know, the tools often use the same buttons, but depending on the current tool, a different selection may be expected for the tool to function properly.

To ensure that the resulting data received via the socket connection is processed correctly, there is also a specialized Visualization Manager. This manager ensures that whenever a method from the script is called, the provided data is visualized correctly. Because both single-surface (Direct) and two-surface (Indirect) data need to be stored, separate empty GameObjects have been created in Unity to hold this data.

For example:

- When the user clicks a surface in Direct mode, the edges and vertices are stored in the Edges and Vertices GameObject (First).
- When the user clicks a second surface in Indirect mode, the corresponding edges and vertices are stored in the Edges and Vertices GameObject (Second).
- When the user selects a vertex during Indirect mode, the resulting edges are calculated and stored in the Edges and Vertices GameObject (Result).

This structure allows data to be stored efficiently and easily removed when needed (Figure 4.3).

There are a few other managers involved as well, most of which are related to GUI functionalities. However, one other important manager is the Feature Manager. This manager handles user interaction with vertices, edges, and planes. It mainly manages the user's selection actions by constantly checking which vertex, edge, or plane is closest to the ray, and highlighting it. This approach means the user doesn't have to be precisely on top of a vertex, edge, or plane to select it—they can select from a small distance instead. This design choice helps compensate for the difficulty of precise navigation in AR, where users rarely have a perfectly steady hand.

Finally, there are two socket scripts responsible for establishing network connections. In this system, the Magic Leap acts both as a server and a client, depending on the situation. It acts as a server to allow one or more CAD environments to connect and retrieve saved data, and as a client to connect to OpenCascade. This setup also allows multiple Magic Leap devices to connect to OpenCascade simultaneously if needed.

4.2.7 Path Conversion

As mentioned earlier, paths can also be stored within the system. These paths can be either open or closed. Once saved, they can later be retrieved in Fusion360 to be loaded and used in that environment. However, before this is possible, we first need to convert 3D coordinates into 2D coordinates. Additionally, the stored coordinates must be made relative to a specific reference point. In this section, we'll explain how this transformation is handled and why it becomes more challenging when a path is stored using the Indirect Tool.

When a user saves a path using either the Direct or Indirect tool, each selected edge is stored by recording its coordinates. If the edge is a straight line, only the start and end points are saved. For more complex curves—like arcs or splines—the line is broken up into smaller segments, and points are saved at regular intervals along the shape. So, in general, every edge in a contour is stored through a list of coordinates that define its shape.

Once we've collected all these 3D coordinates, we need to make sure that this contour can be placed anywhere in the virtual world without losing its shape. At this stage, however, all points are still defined in global world space. That means they contain absolute positions, making it harder to reuse or reposition the shape elsewhere. To resolve this, we select a reference point and convert all the other coordinates into relative positions based on this point. In our implementation, we use the contour's center point as this reference, essentially moving it to the origin point (0, 0) of our 2D space.

Let's make this clearer with an example. Suppose we have a 2D square defined by the following coordinates: (2, 3), (2, 6), (5, 6), and (5, 3). If we want to shift this square so that one of its corners becomes the new origin (0, 0), we subtract (2, 3) from all the points. The resulting relative coordinates become: (0, 0), (0, 3), (3, 3), and (3, 0). This makes the shape easier to manipulate and ensures consistency regardless of its location in the original 3D world. See figure 4.4 for a visual representation.

Now that we've converted the points into relative coordinates, the next step is to transform the 3D coordinates into 2D. This step depends heavily on the nature of the plane the path lies on. There are two main scenarios: one where the entire path lies flat on one of the standard coordinate planes (like XY, XZ, or YZ), and another where the plane is arbitrarily tilted in 3D space. The first case is relatively straightforward—you can simply remove the depth coordinate (for example, drop the Z value for an XY-aligned plane) and you're done.

The second case, however, is more complex. When the plane is not aligned to one of the standard axes, the depth of each point varies across the surface. That means we can't just drop one coordinate, we need a more robust approach to flatten the 3D shape into 2D without losing accuracy or distorting the geometry. To solve this, we use a method called Principal Component Analysis (PCA).

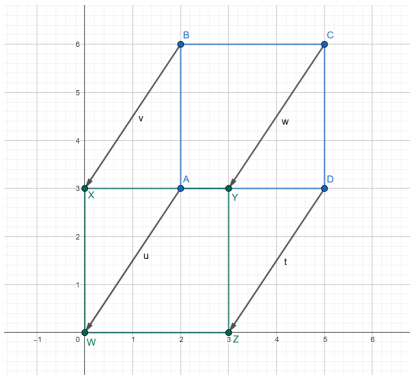

Figure 4.4: Converting from absolute to relative coordinates: The image shows two squares, one with absolute coordinates (A, B, C, D) and the other with relative coordinates (W, X, Y, Z). The vectors from A to W, B to X, and so on represent the shift needed to adjust each point relative to the reference point, illustrating the transformation of the shape into a new coordinate system.

PCA allows us to find the best-fit plane for a set of points in 3D space. Here's how it works in simplified terms: we start by computing the centroid of all the points, which gives us the average position. Then we center the point cloud around this centroid, effectively shifting all points so the mean becomes the origin. With this normalized data, we apply Singular Value Decomposition (SVD), which is a mathematical method that gives us the principal directions along which the points are most spread out.

These directions form a new coordinate system: the first two principal components represent the plane that best fits the shape, while the third component gives us the normal vector, which is perpendicular to that plane. Using these, we create a rotation matrix that allows us to project every 3D point onto the new 2D plane. Once projected, we extract only the X and Y components to get the final 2D representation.

The only issue arises when a user saves a path using the Indirect Tool — finding a suitable projection plane isn't always straightforward. If the selected edges can't all be projected onto the same plane, the resulting 2D points will be incorrectly projected. Unfortunately, there's no reliable way to reconstruct an accurate 2D path from this, so it's best to avoid saving paths using the Indirect Tool whenever possible.

That said, even if such a path is saved, it's unlikely to be useful in real-world applications. These paths typically consist of edges connecting one vertex to all others (Euclidean distances), which doesn't actually form a meaningful shape. That's why, in this context, distances and angles are far more important than the path itself.

Chapter 5

Evaluation

Now that we have familiarized ourselves with the entire system and how it works, this chapter will cover the evaluation. Instead of conducting a user study, which is typically for ascertaining usability and user satisfaction, this evaluation focuses on the technical performance and efficiency of the system. Because this project remains in the proof-of-concept stage, the primary goal is to validate whether the system offers precise and effective measurement compared to traditional methods. A technical evaluation enables an objective assessment of performance metrics, independent of the variability caused by different user behaviors. Once the system is more established and stable, subsequent work can include long-term user studies to establish usability and user experience in actual use contexts.

To assess this technical performance and efficiency, we compare the system's results with those obtained using traditional methods. These conventional approaches typically involve extracting geometric features from objects by, for example, measuring distances or angles, or manually tracing the contours of objects. The comparisons are based on the examples referenced in this thesis: the walkthrough for creating a clamp for a spool, as well as the three real-world use cases — the cup holder, the side table extension, and the phone holder.

Having established which examples will be used for comparison, the next step is to define how the evaluation will be conducted. One might consider measuring efficiency based on the time it takes to complete a task. However, time is an unreliable variable, as it can vary significantly from person to person. For instance, measuring the length of a table can take longer for some users due to difficulties with tool alignment or reading measurements accurately. These issues can occur with both traditional methods and within the proposed system itself — such as when a user selects the wrong tool or makes an incorrect selection. Because numerous factors can influence speed and because it varies between individuals, time is not the most suitable metric. Instead, a more robust and consistent variable is the number of "actions" required to extract a geometric feature. In this context, an "action" refers to any step that a user must take to achieve the desired measurement or result. By using action count as the primary variable, we can compare both methods in their optimal forms, without external factors interfering. This allows us to make a cleaner and more objective comparison of the system's efficiency.

With this evaluation method in place, we can now proceed to the actual comparison of the examples. As mentioned, this will be done by analyzing the number of actions required in the optimal scenario to successfully complete the walkthrough or real-world use cases. By comparing these action counts between the traditional methods and the system presented in this thesis, we aim to demonstrate how much manual effort can be reduced through the use of our system — and, by extension, how this reduction in manual steps may also decrease the likelihood of user errors.

Before we begin, it's important to clearly define what we mean by an "action" for both methods.

This definition has already been introduced in a general sense, but now we will refine it further by specifying what constitutes an action in each individual case. In the case of traditional measuring, an action refers to any measurement or manual step required to obtain the final geometric feature. There are three specific cases to consider:

- Distance: Aligning the measuring tool, reading the value, and writing it down. This results in a total of 3 actions.
- Angle: Also requires 3 actions, following the same reasoning as with distance.
- Path: Taking a picture of the object and using that image to trace the path. This involves 2 actions in total.

However, each of these cases also requires an additional action to transfer the result into Fusion. This could be a point of debate, as values can technically be entered directly into Fusion360. Nevertheless, for consistency with the approach used in our own system, we assume that the same workflow is followed here as well. In this workflow, all geometric features are first obtained and then collectively used in Fusion. To mirror this process with traditional methods, we assume that all measurements are written down first and only then transferred into Fusion one by one. This approach also minimizes context switching, just as it does in our system. Therefore, the total number of actions needed is 4, 4, and 3 respectively.

For my own system, the definition of an "action" is slightly different. In this case, an action refers to the selection of edges made after using a specific tool. This is because after each tool is used, the system visualizes the edges that can be used as features. Again, we consider three cases:

- Distance: Selecting the correct edge and saving it. This results in 2 actions.
- Angle: Here, 3 actions are needed, since two edges must be selected.
- Path: Although the traditional method allows a path to be traced over an image, this still requires several manual steps. To keep the comparison consistent, we will also consider this as 2 actions: selecting the desired path and saving it.

One major advantage of my system is that the export of all features can be done at once. This means that, in total, the required number of actions becomes 2, 3, and 2 respectively, with just 1 final action added to export everything at once.

Now that we have clearly defined what an "action" means in both contexts, we can move on to comparing the walkthrough and use cases between the traditional method and my system. The results of this comparison are summarized in the table below.

Example	Traditional	Capturing System
Walkthrough	(3D * 4) + (2P * 3) = 18	(3D * 2) + (2P * 2) + 1 = 11
Cup Holder	(1P * 3) = 3	(1P * 2) + 1 = 3
Table	(3D * 4) 12	(3D * 2) + 1 = 7
Phone Holder	(3D * 4) + (1A * 4) = 16	(3D * 3) + (1A * 3) + 1 = 13

Table 5.1: Comparison of how many action need to be taken with D (Distance), A (Angle) and P (path).

As shown in Table 5.1, it can be observed that my system generally requires fewer actions compared to the traditional method. However, the difference in action count is not always substantial. What becomes clear, though, is that the more features are captured, the more noticeable the gap becomes. This is mainly due to the fact that in my system, all features can be exported in a single step.

Efficiency in my system isn't limited to just exporting. Another major advantage is the elimination of manual errors, which are still possible in traditional methods. So even in cases where

the number of actions is roughly the same, it's important to remember that each manual step in traditional measuring introduces the possibility of a mistake. In fact, the number of potential errors in the traditional method is roughly equal to the number of actions taken. With my system, as mentioned earlier, such errors are avoided entirely—although it does require the user to think creatively about how to extract certain features using the available tools.

Overall, we can conclude that my system is quite efficient and significantly reduces the likelihood of measurement errors. The trade-off, however, is that users must learn to work flexibly and creatively with the different tools available to achieve the same results.

Chapter 6

Discussion

This chapter provides a critical reflection on the results of the project, assesses the overall performance of the developed system, and positions its contributions relative to traditional approaches. The primary objective of this research was to explore whether an augmented reality (AR) system could improve the process of extracting geometric features from physical objects and transferring them into a CAD environment, with a particular focus on reducing manual effort and minimizing errors. Based on a series of test cases—including a guided walkthrough and three real-world use scenarios—the system has proven to be both functional and efficient in achieving these goals.

The results clearly demonstrate that the system enables users to extract key dimensions with fewer "actions" compared to traditional measuring tools. Furthermore, it reduces the likelihood of common human errors such as incorrect tool alignment, misreading measurements, or mistakes during manual data transfer. In doing so, the system streamlines the early stages of the design process and improves reliability without requiring the user to possess advanced measuring skills.

6.1 Error Reduction

Traditional measurement workflows are inherently prone to human error. A common example is the misalignment of measuring tools—such as rulers or calipers—which can result in inaccurate distance readings. In this system, such alignment issues are avoided entirely because the AR interface highlights measurable edges automatically, based on the selected tool and the loaded STEP file. Users no longer need to physically position tools; instead, they simply select the relevant features from an augmented overlay.

Another frequent source of error in traditional methods is the manual reading and transcription of values. It is not uncommon for users to misread a ruler or to accidentally enter an incorrect value into CAD software. The developed AR system bypasses this process by directly calculating distances or angles based on user-selected features, using geometric data from the STEP file. The result is then transferred automatically to Fusion360 via a socket connection, ensuring consistency and accuracy throughout.

Additional difficulties arise when users attempt to digitize more complex geometries, such as curves or contours. Conventional approaches often involve tracing shapes manually—either by sketching on paper or using tablets connected to CAD software. These methods not only require more time and skill but also introduce variability depending on the user's drawing precision. By contrast, the AR system allows users to select the edges that define a contour or path, which can then be imported as an accurate sketch in Fusion with no additional input.

In some cases, specific measurements may be entirely inaccessible using physical tools, either

because they are too small to measure accurately or located in hard-to-reach areas. The AR system addresses this limitation by offering tools such as virtual cross-sections, which provide insight into the internal structure of an object without requiring physical modification.

Despite these improvements, the system is not without its own challenges. One such limitation is that users must often decide for themselves which tool to use in a given situation. Since multiple tools can sometimes be used to extract the same feature, this decision-making process can introduce inefficiencies, particularly for novice users. Determining the most appropriate approach is not always straightforward and can lead to a trial-and-error workflow.

To summarize everything, the table below highlights the key differences between traditional measurement methods and my own system.

Error	Traditional	Capturing System
Misalignment	✓	✗
Misreading	✓	✗
Wrong Transfer	✓	✗
Shape Reconstruction	✗	✓
Creativity	✓	✓

Table 6.1: Summary of where the errors can occur in traditional methods vs the implemented system.

6.2 Measuring Efficiency

To evaluate the system's efficiency without relying on time-based metrics—which can vary significantly between users—this research used the number of user actions as an objective measure. An action refers to any manual step needed to extract and transfer a measurement, such as tool alignment, reading values, or entering data into Fusion360.

As explained in detail in the evaluation chapter, this metric allows for a fair comparison between traditional methods and the AR-based system, independent of user experience or skill. The results showed that the AR system consistently requires fewer actions due to automated calculations, direct feature selection, and seamless data transfer. By minimizing these manual interactions, the system not only accelerates the workflow but also reduces the likelihood of human error.

This supports the broader conclusion that the system offers a more streamlined and reliable design process, particularly in scenarios where precision and consistency are essential.

6.3 Limitations

Although the system provides a streamlined alternative to traditional measuring workflows, it still presents several limitations that affect its general usability and scalability in professional settings. These constraints range from technical dependencies to usability bottlenecks during detailed tasks.

First of all, the current system relies on the use of STEP files and ArUco markers. This approach requires users to manually prepare each object before the system can be used effectively. For every object the user wants to work with, a corresponding STEP file must be available. This file then needs to be manually loaded and aligned using an ArUco marker. This alignment process is crucial for ensuring a correct user experience, but it also introduces a significant barrier, as the user is solely responsible for performing it accurately.

Additionally, the system assumes that a digital CAD model (in the form of a STEP file) is available for each object. This assumption limits the system's applicability to cases where such a file exists or can be created. However, this is not always practical, especially for reverse engineering or when working with unknown or handmade objects.

Another limitation is that the system currently supports only a limited number of tools. While the Direct and Indirect tools allow users to define various types of measurements and paths, they can be inefficient in certain contexts. For example, when many edges are present, it becomes difficult to quickly select the correct one. This leads to extra time spent navigating the interface and interpreting geometry, especially when working with detailed or complex surfaces. Furthermore, users can currently only create planes on flat surfaces, which limits flexibility in more complex scenarios. In contrast, CAD tools like Fusion360 allow planes to be defined using points, lines, or intersections. This level of versatility is not yet supported in the current system.

Smaller objects can also pose challenges. Because AR-based interaction often lacks the precision of traditional input devices, fine-grained feature extraction can be difficult—particularly for objects with small details or complex geometry. Selecting the correct edge, point, or surface can be hard when the object is physically small or partially obstructed. This limits the system's usefulness in situations where high accuracy is essential.

Lastly, the conversion of measurements into Fusion does not happen in real time. At present, users are expected to capture all the required features at the beginning of the process, after which these features are used within Fusion. However, if the user forgets to measure something or makes an error during the measurement process, they must return to the object, re-capture the feature, and import it again into Fusion360. This fragmented process breaks the design flow and introduces unnecessary repetition.

6.4 Future Work

To address the limitations outlined above, several opportunities for future improvement can be identified. These include both technical upgrades and workflow enhancements aimed at improving flexibility, accuracy, and user experience.

A significant improvement would be to reduce or eliminate the dependency on pre-existing STEP files. Currently, users must have a CAD model of the object beforehand, which limits the system's usability in reverse engineering or informal prototyping contexts. One promising direction is to implement object recognition, allowing the system to automatically identify the correct object and retrieve or generate the corresponding digital model. This could simplify setup dramatically and make the system usable in more dynamic environments.

An even more flexible approach would be to replace pre-existing CAD files entirely by using 3D scanning or image-based reconstruction. Although commercial 3D scanners still struggle with precision and ease of use, they offer the potential to generate digital models without manual modeling effort. In this context, the work by Noeckel et al. [20] is particularly relevant. They propose a method to reconstruct manufacturing-ready CAD models directly from photographs, without relying on traditional 3D reconstruction techniques. Instead, their system identifies individual components and how they connect, effectively enabling reverse engineering of unknown physical objects. Techniques like this could be adapted to serve as an automatic frontend to the current system, eliminating the need for STEP files altogether and opening the door to more fully automated feature extraction workflows.

In addition, future work could focus on enhancing the current toolset. At the moment, users are limited to creating planes on flat surfaces and must manually determine which edges or faces are relevant. Expanding the system to support more advanced constructions—such as planes based on multiple points, axes, or surface intersections—would greatly increase its flexibility. The Direct and Indirect tools could also be refined to better handle complex geometries, or even

be extended with AI-powered suggestions based on previously selected features. As discussed in the Evaluation chapter, many errors in traditional measurement are related to manual selection and transfer. By predicting which features are likely to be needed next, the system could guide the user through the measurement process more efficiently.

Another avenue of improvement is user interaction for small or detailed objects. Currently, the precision of AR-based selection can be limiting. A possible solution would be to implement a zoom or "magnify" function that allows users to scale up their view of the object virtually, making it easier to select features accurately. This would be particularly beneficial when working with intricate shapes or when occlusion becomes a problem.

Lastly, the integration with Fusion360 could be made more seamless by introducing real-time synchronization. At present, features must be collected and then exported manually. A live link between the AR system and Fusion would allow features to be updated instantly as they are selected or modified. This would support a more fluid design workflow and reduce the need to repeatedly switch between environments. Furthermore, exporting the Fusion model back into the AR space would allow users to validate their designs in context—overlaying the virtual model on the real object to check fit, alignment, and scale before moving to fabrication.

Chapter 7

Conclusion

Developing a system that allows users to extract geometric features from the real world into Augmented Reality (AR) and import them directly into a CAD environment is no small feat. What started as a technical investigation quickly grew into a much broader question—one that looked at how humans interact with digital tools and how design workflows could be made more intuitive and precise. This thesis illustrates that AR can indeed reduce errors and speed up the design process but also reveals some essential challenges yet to be addressed.

One of the key findings of this study is that the system significantly reduces the types of errors inherent in traditional measurement methods—like misreading measurements or misaligning tools. Automating the feature extraction and data transfer not only improves efficiency, but also accuracy and consistency of results. That said, it's also clear that there remains a very real role for the user to play. While the system lowers the barrier to entry, it still requires the user to engage actively—especially in cases of challenging shapes or complex geometries.

In relation to the central research question—how AR can improve measurement accuracy and simplify CAD modeling—this thesis offers strong answers:

- New in-situ measurement methods were developed and tested with success. By using AR headsets in combination with ArUco markers and pre-loaded STEP files, users can capture precise measurements directly from physical objects, all within an intuitive and guided workflow.
- The transfer of data to CAD software was streamlined through a custom-built Fusion360 plugin. This makes it possible to export measurements and geometric features automatically, complete with helpful visual context, cutting down on both time and errors.
- To support more advanced use cases, tools such as axis extraction and slicing were added. These allow users to capture dimensions that are not directly visible, expanding the system's usefulness for more technical or layered design challenges.

No system is perfect, obviously. Using STEP files and ArUco markers ensures a high degree of accuracy, but with some complexity too. The users who are new to preparing the files or don't know how to position markers accurately can find the initialization a bit tedious. However, this compromise is relatively unavoidable in shifting towards higher precision.

The learning and refinement of this system have been rewarding and a humbling experience. Each phase introduced new technical hurdles that forced me to make tough decisions, under not-always-ideal conditions. Developing within these constraints—such as the lack of real-time sync or the need for better integration—forced me to put solutions first that were not only functional, but also pragmatic for real-world users. Consequently, this project was not merely a technical exercise, it was a learning experience in creative problem-solving, user-focused thinking, and persistence.

The final result provides a strong foundation to build upon. Future improvements like automatic object detection, a more diverse toolset, and zoom functionality for fine-grained control are logical next steps. But beyond the system itself, I hope this thesis highlights a broader point: that technology is most valuable when it adapts to the users who use it—not the other way around. I believe the future of AR in design lies in balancing precision with usability, and this project aims to take a small but meaningful step in that direction.

Additionally, this work shows that AR has strong potential to reduce human error during the measurement process. By guiding users visually and interactively, AR helps prevent common mistakes—such as misalignment or misreading—and creates a more reliable foundation for design from the very beginning.

Bibliography

[1] J. Kim, A. Guo, T. Yeh, S. E. Hudson, and J. Mankoff, "Understanding uncertainty in measurement and accommodating its impact in 3d modeling and printing," in *Proceedings of the 2017 Conference on Designing Interactive Systems*, ser. DIS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1067–1078. [Online]. Available: https://doi.org/10.1145/3064663.3064690

[2] C. Weichel, J. Alexander, A. Karnik, and H. Gellersen, "Spata: Spatio-tangible tools for fabrication-aware design," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 189–196. [Online]. Available: https://doi.org/10.1145/2677199.2680576

[3] J. Lee, V. Su, S. Ren, and H. Ishii, "Handscape: a vectorizing tape measure for on-site measuring applications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 137–144. [Online]. Available: https://doi.org/10.1145/332040.332417

[4] E. Stemasov, D. Ledo, G. Fitzmaurice, and F. Anderson, "Immersive sampling: Exploring sampling for future creative practices in media-rich, immersive spaces," in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, ser. DIS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 212–229. [Online]. Available: https://doi.org/10.1145/3563657.3596131

[5] B. Lee, M. Cho, J. Min, and D. Saakes, "Posing and acting as input for personalizing furniture," in *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, ser. NordiCHI '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2971485.2971487

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," 2018. [Online]. Available: https://arxiv.org/abs/1712.07629

[7] E. Stemasov, S. Demharter, M. Rädler, J. Gugenheimer, and E. Rukzio, "param: Leveraging parametric design in extended reality to support the personalization of artifacts for personal fabrication," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, ser. CHI '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3613904.3642083

[8] P. Reipschläger and R. Dachselt, "Designar: Immersive 3d-modeling combining augmented reality with interactive displays," in *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ser. ISS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 29–41. [Online]. Available: https://doi.org/10.1145/3343055.3359718

[9] X. Yu, S. DiVerdi, A. Sharma, and Y. Gingold, "Scaffoldsketch: Accurate industrial design drawing in vr," in *The 34th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 372–384. [Online]. Available: https://doi.org/10.1145/3472749.3474756

[10] R. Ramakers, D. Leen, J. Kim, K. Luyten, S. Houben, and T. Veuskens, "Measurement patterns: User-oriented strategies for dealing with measurements and dimensions in making processes," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581157

[11] E. Stemasov, T. Wagner, J. Gugenheimer, and E. Rukzio, "Mix&match: Towards omitting modelling through in-situ remixing of model repository artifacts in mixed reality," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–12. [Online]. Available: https://doi.org/10.1145/3313831.3376839

[12] T. N. Hoang, R. T. Smith, and B. H. Thomas, "Ultrasonic glove input device for distance-based interactions," in *2013 23rd International Conference on Artificial Reality and Telexistence (ICAT)*, 2013, pp. 46–53.

[13] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen, "Mixfab: a mixed-reality environment for personal fabrication," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 3855–3864. [Online]. Available: https://doi.org/10.1145/2556288.2557090

[14] H. Peng, J. Briggs, C.-Y. Wang, K. Guo, J. Kider, S. Mueller, P. Baudisch, and F. Guimbretière, "Roma: Interactive fabrication with augmented reality and a robotic 3d printer," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–12. [Online]. Available: https://doi.org/10.1145/3173574.3174153

[15] M. Lau, G. Saul, J. Mitani, and T. Igarashi, "Modeling-in-context: user design of complementary objects with a single photo," in *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, ser. SBIM '10. Goslar, DEU: Eurographics Association, 2010, p. 17–24.

[16] H. Xia, B. Araujo, T. Grossman, and D. Wigdor, "Object-oriented drawing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 4610–4621. [Online]. Available: https://doi.org/10.1145/2858036.2858075

[17] R. Sehgal, R. Gupta, and N. Anand, "Automatic extraction of 3d body measurements from 2d images of a female form," in *Proceedings of the IOSR Journal of Engineering (IOSRJEN), Volume 5, Issue 3*, 07 2018.

[18] J. Yang, A. Zeng, R. Zhang, and L. Zhang, "Unipose: Detecting any keypoints," 2024. [Online]. Available: https://openreview.net/forum?id=v2J205zwlu

[19] K. Wu and Z. Cheng, "Refar: 3d sketch-based modeling with in-situ references," in *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2022, pp. 507–511.

[20] J. Noeckel, H. Zhao, B. Curless, and A. Schulz, "Fabrication-aware reverse engineering for carpentry," 2021. [Online]. Available: https://arxiv.org/abs/2107.09965