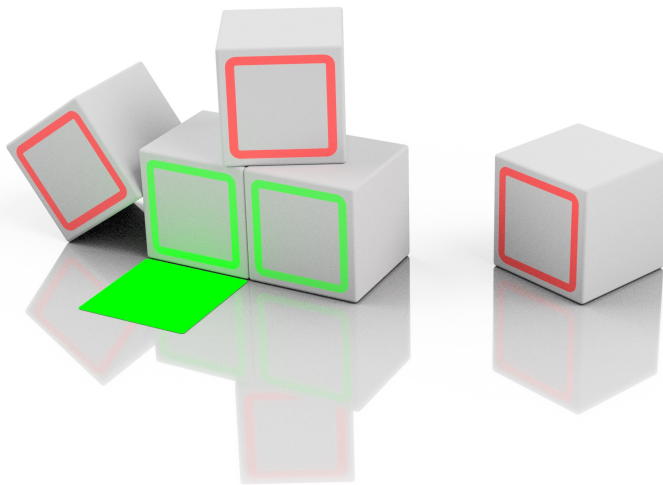




CHALMERS

KANDIDATARBETESRAPPORT



Believe in Me

The Intentional Robot

AMÉLIE ELDH, MARIUS GRIMSTAD, JOHAN LUND,
ANDREAS LÖFMAN, FREDRIK MACINTOSH, SOFIA SUNDIN

Institutionen för signaler och system
Chalmers tekniska högskola
Göteborg, 2016

SSYX02-16-02

Förord

Rapporten är resultatet av ett examensarbete på kandidatnivå, utfört under våren 2016. Examensarbetet utfördes av sex civilingenjörstudenter på Elektroteknik- respektive Automation och mekatronik-programmet vid Institutionen för signaler och system på Chalmers tekniska högskola.

Ett stort tack ges till **Ashfaq Hussain Farooqui, Doktorand, Signaler och system** för all ovärderlig hjälp med C++, ROS och Linux.

Brett Renfer, Director, Experience Design at Collins (www.robotconscience.com) förtjänar gruppens tacksamhet för all hjälp med ofxLibwebsockets.

Handledare **Kristofer Bengtsson, Teknologie Doktor vid Signaler och system** har varit till stor hjälp och delat med sig av sina idéer och kunskaper.

Tack till **Patrik Bergagård, Teknologie Doktor vid Signaler och system** för de värdefulla råden och tankeställarna kring arbetet.

Peter Hegarty, Biträdande professor och studierektor för forskarutbildningen i matematik vid Matematiska vetenskaper tackas för hjälp med vektorteori.

Slutligen förtjänar **Hans Sjöberg, Forskningsingenjör vid Produkt- och produktionsutveckling/Produktionssystem** ett tack för att han bistod med nödvändiga delar och lokal till prototypen.

Sammandrag

De senaste årens snabba utveckling av autonoma maskiner ger fler och fler tillfällen då människor kan träffa på dem i sin vardag. För att kunna utnyttja de möjligheter som dessa maskiner ger, krävs det att interaktionen mellan människa och maskin fungerar på ett smidigt sätt. Denna interaktion bygger i grunden på att kommunikationen mellan människa och maskin fungerar på ett tillfredställande vis. Merparten av den forskning som tidigare gjorts berör hur maskiner skall kunna tolka, eller imitera mänskligt beteende. I det här arbetet ligger istället fokus på hur en maskin skall kunna kommunicera sin intention till en människa med hjälp av visuell kommunikation.

För att undersöka hur väl visuell kommunikation fungerar konstruerades en prototyp som imiterar ett tänkbart kommunikationssystem tillhörande en autonom maskin. Prototypen kan ta till sig information om omvärlden via en djupseende kamera. För att förmedla information använder den sig av en projektor. Genom att ta hänsyn till var objekt i omvärlden befinner sig och hur de ser ut kan prototypen projicera information på valfria ytor utan förvrängning.

Projektionerna användes för att förmedla information till en mänsklig operatör i form av visuella instruktioner. Instruktionerna visar med hjälp av färg och form hur ett användardefinierat antal kuber skall placeras för att konstruera ett klosstorn. De visar även om en kub är rätt eller fel placerad med hjälp av färg.

Från de resultat som insamlades i en undersökning av prototypen visar att den till stor del lyckades förmedla sina instruktioner till människor, men att rum finns för förbättringsarbete vad gäller dess tydlighet.

Abstract

The rapid development of autonomous machines over the last few years gives more and more opportunities for people to interact with them in their everyday life. To be able to take advantage of the possibilities that machines provide, the interaction between people and machines needs to work in a smoothly fashion. This interaction is based on how the communication between humans and machines works in a satisfactory way. Most of the studies involves how machines would interpret or imitate human behaviour. In this study, however, the main focus lies on how a machine could communicate its intentions to a human with the help of visual communication.

To examine how well visually based communication works, a prototype were constructed, the prototype imitates the possible communication system of an autonomous machine. The prototype collects information from its surroundings through a camera with depth perception and uses a projector to mediate information to humans. By taking the position and rotation into consideration, the prototype can project information on various surfaces without distortion.

The projections are used to deliver information to a human operator in the form of visual instructions. The instructions convey how a user defined quantity of boxes should be placed to construct a tower of blocks, by using colours and shapes. The projections also show whether a cube is placed incorrectly or not, with the use of colours.

The results gathered from a survey regarding the prototype shows that it managed to relay its intentions in a mostly understandable way, but that there are still room for improvements regarding its clarity.

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Kommunikationsmöjligheter	2
1.3	Syfte	2
1.4	Utmaningar	2
1.5	Avgränsningar	3
1.6	Rapportens upplägg	3
2	Prototypen BelieveMe	4
2.1	Konceptöversikt	4
2.2	Systemet i sin helhet	4
2.3	Kinect-kamera	5
2.4	Freenect	5
2.5	ROS - Robot Operating System	6
2.6	Projektor	6
2.7	Förflyttning av BelieveMe	7
2.8	AssembleMe - Interaktion mellan människa och BelieveMe	7
2.9	Krav och specifikationer för BelieveMe	8
3	Projektion av intention	9
3.1	Projection mapping	9
3.1.1	openFrameworks	10
3.1.2	Beskrivning av funktion	10
3.2	Mappning av 3D-koordinater till ett 2D-plan	10
3.2.1	Teori	10
3.2.2	Förskjutning av projektionsplan	12
3.3	Visualisering av intention	13
3.3.1	Test av färger med hjälp av projektor	13
3.3.2	Val av visualisering för att skapa tilltro och trygghet	14
3.3.3	Undersökning om hur BelieveMe kan skapa tilltro och trygghet	15
3.3.4	Sammanställning av färgval	16
3.3.5	Visualisering vid AssembleMe	16
4	Programvaran bakom BelieveMe	18
4.1	Input med hjälp av Kinect	18
4.2	Rviz	18
4.3	Personidentifiering	18
4.4	Navigation	20
4.5	AssembleMe	22
4.5.1	Kontroll av lådas position	23
4.5.2	Marköridentifiering	24
4.5.3	Publicering av hörn	25
4.6	Kommunikation mellan ROS och projection mapping	26
4.6.1	Websocket	26
4.6.2	Rosbridge suite	26
4.7	Projection mapping	27

5	Utvärdering av BelieveMe	28
5.1	Verifiering av krav på BelieveMe	28
5.2	Förutsättningar inför AssembleMe	28
5.3	Test av AssembleMe	28
6	Slutsats	32
7	Förslag för fortsatt arbete	33
A	Undersökning av hur färger, former och autonoma fordon uppfattas	Appendix A1
B	Undersökning av AssembleMe	Appendix B1
C	Poängsättning vid test av projektor	Appendix C1
D	Rotation av kvaternion	Appendix D1

Förkortningar som används

HRI	Human Robot Interaction
ROS	Robot Operating System
SLAM	Simultaneous localization and mapping

Ordlista

AssembleMe	Interaktion mellan människa och färdigställd prototyp, BelieveMe, i ett konstruktionsexperiment där man skall bygga ett klostorn
Avbildning	Att överföra koordinater ifrån ett koordinatsystem till ett annat.
BelieveMe	Namnet på prototypen.
FPS	Bilder per sekund.
Callback	En funktion som startas av en viss händelse, till exempel för att ta hand om inkommande data.
JSON	Ett språkoberoende format för att strukturera upp och utväxla data av olika typer.
Kvaternion	En utvidgning av de komplexa talen som kan användas för att beskriva rotation för 3-dimensionella objekt.
Marker	Ett virtuellt 3-dimensionellt objekt som används för att visa till exempel en låda eller person.
Markör	Ett väldefinierat, 2-dimensionellt mönster lämpat för avläsning och igenkänning med hjälp av en kamera.
Nod	Ett program som körs inom ROS, kan prenumerera och publicera.
Odometri	Att med hjälp av data från rörelsesensorer, eller via bildanalys, bestämma position.
Overhead	Ytterligare informationsdata som fungerar som bilaga till den data som ska skickas.
Pose	En datatyp som innehåller ett objekts position och rotation i alla led.
Projection mapping	Projicering som sker på en icke-plan yta och tar hänsyn till dess 3-dimensionella ojämnheter.
Prenumerera	Lyssna på data från ett topic.
Publicera	Skicka data till ett topic.
Punktmoln	En mängd punkter i ett koordinatsystem. I det här fallet en mängd koordinater i ett 3-dimensionellt rum.
Topic	Ett unikt namn för en dataström som godtyckligt antal noder kan publicera eller prenumerera på.
Cluster	En samling av punkter som en del av ett större punktmoln.

1 Inledning

1.1 Bakgrund

Automation är inget nytt begrepp; redan i antika Greklands mytologi talas det om hur Hephaestus, smidets gud, tillverkade en automaton, en självgående maskin, för att hjälpa honom i smedjan [1]. Några århundraden före Kristus fanns förstas inte möjligheten att konstruera något genomgripande autonomt, än mindre en humanoid sådan, men konceptet var fött.

Utvecklingen av autonoma system som tar egna beslut utifrån insamlad data från omvärlden går framåt i rasande fart. Ett exempel är Googles självkörande bilar som redan trafikerar vägar i USA [2].

Med det ökade antalet robotar och mer eller mindre autonoma processer är behovet av tydlig kommunikation mellan människa och maskin viktigt. Även innan robotar blev en verklighet lekte dock filosofer och författare med tanken. Den som kan sägas var först med att ställa upp och problematisera robotars interaktion med människor var Issac Asimov, som redan på 40-talet formulerade tre lagar för att kontrollera robotars beteende i förhållande till människor:

- ”1. En robot får aldrig skada en människa eller, genom att inte ingripa, tillåta att en människa kommer till skada.*
- ”2. En robot måste lyda order från en människa, förutom om sådana order kommer i konflikt med första lagen.*
- ”3. En robot måste skydda sin egen existens, såvida detta inte kommer i konflikt med första eller andra lagen” [3].*

Dessa tre lagar må vara science fiction men har bidragit till att förändra synen på robotar och autonoma system till den nyttobaserade syn som råder idag [4]. Synen på robotar som farliga är dock något som levt kvar, även om kulturella skillnader ibland blir uppenbara [5]. Ofta hanteras problemet med säkerhetssystem där människor och robotar arbetar avskilda från varandra, till exempel i en så kallad robotcell i en fabrik.

I takt med att robotar blir vanligare och deras tillämpning blir mer vardaglig och närmare människan är det inte längre möjligt eller önskvärt att separera maskin och människa. Detta leder till att människor befinner sig och agerar i närheten av robotar med de faror det innebär. Det finns idag på en del fordonssystem, så kallade *Emergency Automatic Braking* [6] för att skydda bland annat fotgängare. Dessa system ämnar dock endast till att skydda människa från system, vilket såklart är viktigt, men kommunikationen mellan människa och maskin måste förtydligas. I en framtid där människor kan förväntas arbeta tillsammans med robotar, kommer mer att krävas. Möjligheterna är något som tydligt illustreras i David Bournes artikel [7] ”My boss the robot”, där det på ett vardagligt sätt återberättas hur ett arbetslag bestående av både människor och robotar, givet fungerande kommunikation kan prestera påfallande effektivt.

Hur människor och olika autonoma maskiner i form av robotar interagerar med varandra är något som har studerats tämligen ingående. Fokus i undersökningarna har oftast varit på hur robotar tolkar och uppfattar människors agerande. Med en kontinuerligt ökande andel autonoma maskiner i kontakt med människor har dock intresset för hur människor uppfattar robotar ökat. Som exempel har roboten Kismets [8] förmåga att kommunicera med omvärlden studerats av Breazeal et al. [9]. Deras slutsats är att människor gärna tolkar in intentioner i beteenden, även där det inte finns någon medveten intention. I en annan studie av Breazeal [10] utvecklas dessa tankar till en mer allmän bild av att tydliggörande av kommunikationen mellan människa och robot starkt ökar de funktioner robotar kan fylla i samhället, även om just den studien fokuserar på att lära robotar tolka människor.

Mer relevant för denna studie, är studier som tar upp både projektion som kommunikationsmedel för roboten, och hur människor reagerar på denna. En sådan har gjorts av Chadalavada et al. [11]. Där visas att om en autonom maskin kan förmedla sin intention tydligt, ökar tilliten bland de människor som interagerar med den.

1.2 Kommunikationsmöjligheter

Att robotar agerar på ett pålitligt sätt är viktigt för att människor skall kunna interagera med dem. Pålitlighet innebär att en autonom maskin kan förmedla sina intentioner och att den har en konsistens i sitt beteende [12]. Förmedlingen kan ske på många sätt, det viktigaste är att alla människor som är i närheten uppfattar dess intentioner.

Kommunikation med ljud är en möjlighet [12], dock kan det uppfattas som störande med regelbundna, högljudda varningar. Dessutom begränsas användningsområdet för systemet eftersom att lokaler så som sjukhus, bibliotek och museum vill ha det lugnt med så lite höga ljud som möjligt. Hörsel-, och även taktila känselanordningar, är ofta inte särskilt effektiva när det kommer till att förmedla mer komplexa budskap [12]. Istället, för att kunna informera människor om t.ex. nästkommande sekvens av rörelser, är visuell kommunikation att föredra. Visualisering gör det möjligt för människor att tolka vart maskinen är på väg och vad den tänker göra, dock måste visualiseringen göras på så sätt att den kan tolkas snabbt och enkelt. Visualisering har tidigare visat sig vara ett mer effektivt kommunikationssätt än hörsel- och känselsystem [13].

Visualisering kan göras på olika sätt. Några exempel är att använda sig av en eller flera skärmar samt att projicera ut på marken eller på föremål. I projektet kommer projicering att användas, eftersom att en stor bild på i stort sett valfri yta återfås, och på så sätt kan delge samma information till fler personer samtidigt. Studien av Chadalavada et al. [11], använde sig av projektion och fick positiv respons.

1.3 Syfte

Projektet har undersökt hur människan kan söka tillit till autonoma system. Med hjälp av projection mapping tillsammans med kameran sensorer skall robotar kunna förmedla vad de ser och vad de tänker göra, samt vad de vill få gjort. Detta skall göra det enklare för människor att interagera med robotar och samtidigt få dem att känna sig tryggare i robotars närvaro.

Målet var därför att utveckla en prototyp, kallad BelieveMe, och med hjälp av denna undersöka hur tydlig och trygg kommunikation mellan människa och prototyp kan uppnås. Resultaten av undersökningen redovisas i denna rapport.

1.4 Utmaningar

En grundläggande fråga var hur ett autonomt system, på ett intuitivt sätt, kan förmedla vilken avsikt det har. Uppgiften bestod till stor del av att undersöka vilka möjliga lösningar som finns för denna frågeställning, samt hur det är möjligt att tillämpa dessa lösningar praktiskt genom att bygga en prototyp.

BelieveMe har med hjälp av AssembleMe visat att det är möjligt att instruera människor att klara av att genomföra uppgifter som blir presenterade för dem utan föregående instruktion. Förmågan att kunna instruera människor leder till mängder av möjliga tillämpningar, t.ex. självinstruerande maskiner eller självkörande fordon som på ett tydligt sätt kan interagera med sin omvärld. Varje specifik tillämpning kräver sina egna lösningar med hur information ska inhämtas och förmedlas men grundprincipen är detsamma.

1.5 Avgränsningar

Då projektet endast var menat att undersöka kommunikationsmöjligheter mellan robot och människa, är BelieveMe enbart en icke-autonom prototyp, som används för att simulera en mobil, autonom konstruktion. BelieveMe samlar heller inte in rörelseinformation utanför projektionsområdet, dels då det inte finns ett sätt att presentera denna data, och dels för att sensorn som ger informationen är riktad mot projektionsområdet.

Den konstruerade prototypen BelieveMe är inte menad att göra en urskiljning på en högre nivå än att den ser skillnad på en människa och ett objekt. Identifieringen av människor sköts av färdiga program [14] och ärver eventuella begränsningar. Detta projekt ämnas inte att användas kommersiellt utan är endast till för att testa interaktionen mellan människa och robot.

De personer, vars erfarenheter ligger till grund för den statistiska datan, hämtades ifrån kandidatgruppens omgivning. Det innebär att det som främst kom att undersökas är attityder och reaktioner från personer hemmahörande i Sverige och med viss relation till teknologstudenter. Antagandet gjordes att västvärlden har såpass homogen attityd till autonoma maskiner, samt ett liknande formspråk, något som gör rapporten relevant även utanför den undersökta sfären.

1.6 Rapportens upplägg

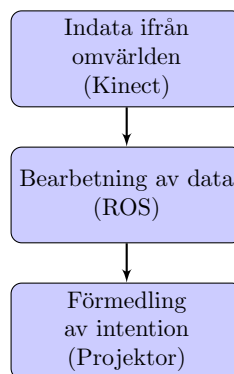
I kapitel 2 ges en översikt över projektet, alltså prototypen BelieveMe, och dess mest väsentliga delar. Kapitel 3 innehåller undersökningar om vad som skall projiceras samt redogör för hur denna projicering sker. I kapitel 4 redovisas all mjukvara bakom projektet och i kapitel 5 diskuteras resultaten som fåtts med hjälp av BelieveMe. Slutligen innehåller kapitel 6 en slutsats om hur väl kommunikationen fungerade och om den bidrog till ökad tillit till autonoma system, och kapitel 7 ger rekommendationer för fortsatt arbete.

2 Prototypen BelieveMe

Detta kapitel är menat att ge en övergripande förståelse över hur systemet bakom prototypen BelieveMe är uppbyggt.

2.1 Konceptöversikt

BelieveMe tar in information om omvärlden, bearbetar denna och presenterar en lämpligt representation till omvärlden. Till förfogande finns en Kinect-kamera¹ [15] och en projektor, vilka styrde valet av hur prototypen skulle byggas. För att hantera och bearbeta datan användes programbiblioteket ROS² [16]. Figur 1 ger en övergriplig bild över hur systemet ser ut.



Figur 1: En visualisering över de delar som ingår i BelieveMe.

2.2 Systemet i sin helhet

För att systemet skall fungera krävs kommunikation mellan de olika ingående delarna, vilket illustreras i Figur 2. Kinect-kameran tar in information från omgivningen, omvandlar den till rådata och skickar vidare den till Freenect³ [17], vilket är ett program som används för att tolka dataflödet från Kinect-kameran. När datan har behandlats i Freenect blir den tillgänglig för ROS via ett flertal topics. Freenect omvandlar rådata från Kinect-kameran till ett punktmoln som skickas till programmen: personidentifiering, navigation och marköridentifiering⁴. När punktmolnet behandlats i de respektive programmen publiceras respektive koordinater på tillhörande topic. För enhetlighet är koordinaterna alltid fyra i antal och representerar hörnen där man vill rita ut en bild. Därefter skickas datan med hjälp av Rosbridge⁵, till projection mapping-mjukvaran⁶, där 3D-koordinater avbildas på ett tvådimensionellt plan. Dessa 2D-koordinater används för att placera motsvarande bilder i programmet, och projektorn projicerar i sin tur ut bilden tillbaka på 3D-miljön.

¹Kinect beskrivs närmare i kapitel 2.3.

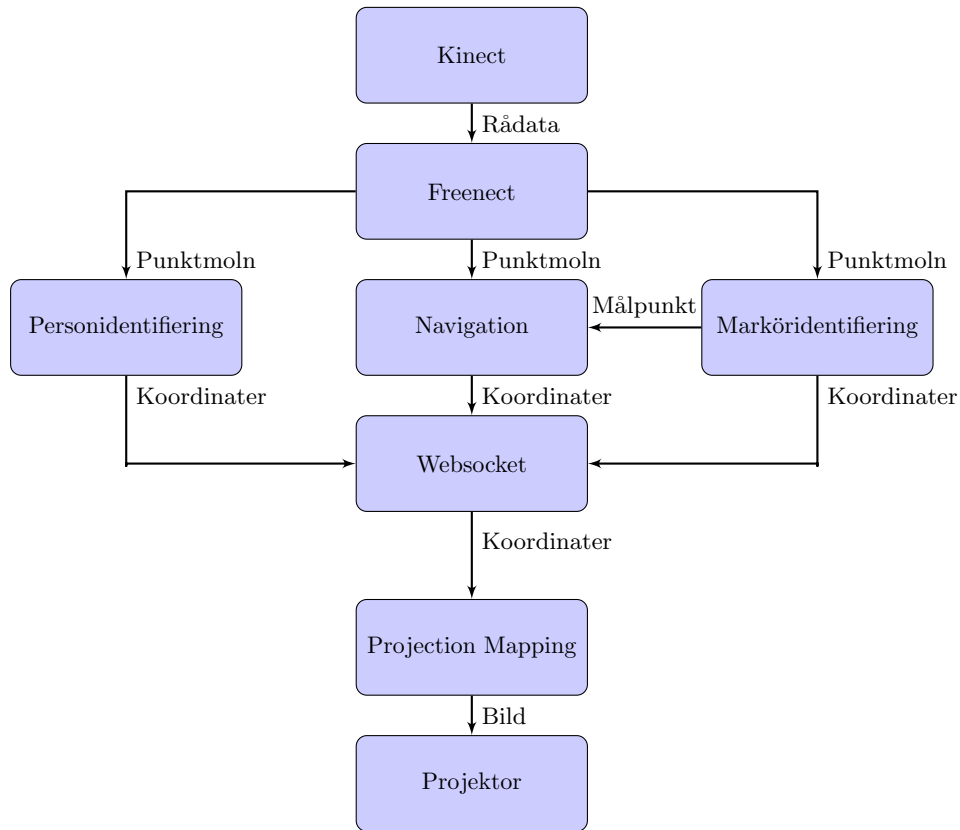
²ROS beskrivs närmare i kapitel 2.5.

³Freenect beskrivs närmare i kapitel 2.4.

⁴Vad dessa tre program gör beskrivs mer ingående i avsnitt 4.3, 4.4 och 4.5.2.

⁵Rosbridge beskrivs närmare i kapitel 4.6.2.

⁶Tillvägagångssättet förklaras noggrannare i kapitel 3.1.



Figur 2: En visualisering över hur delarna i projektet kommunicerar med varandra.

2.3 Kinect-kamera

Kinect är en djupseende kameraenhet utvecklad av Microsoft, ursprungligen utvecklad för deras spelkonsol Xbox 360 [15]. För att läsa av omgivningen framför sig använder den sig av två stycken kameror och en laser.

Den ena kameran har känslighet inom det infraröda spektrumet, som tillsammans med lasern kan mäta djup, och den andra kameran är en vanlig RGB-kamera[15]. Upplösningen är 640x480 pixlar vid 30 FPS[15] på båda kamerorna. Vissa begränsningar finns i systemet då dess sensorer har en maximal horisontell vinkel på 57°, en vertikal på 47° och ett arbetsområde i djupled på ca 0.7-6 m[15]. Beroende på applikation använder sig Kinect-kameran mer eller mindre av RGB-kameran vilket kan medföra problem vid arbete i dåliga ljusförhållanden.

Kinect-kameran användes som systemets datakälla för att identifiera människor och objekt.

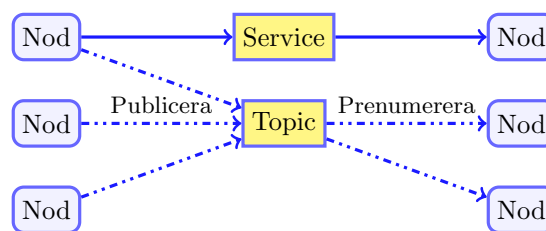
2.4 Freenect

Freenect användes för att kommunicera mellan ROS och Kinect-kameran. Freenect konverterar rådata från kameran till bland annat punktmoln, vilket gör datan tillgänglig för utomstående applikationer som utför bildbehandling och visualisering [17]. Freenects styrka ligger i att det är anpassat för Kinect och har en stabil programkörning, därför var Freenect lämpligt att använda i projektet.

2.5 ROS - Robot Operating System

Robot Operating System (ROS) är ett samlingsbibliotek med öppen källkod för diverse applikationer inom autonoma system [18]. System är uppdelat i så kallade noder, topics, meddelanden, services och en master [19], se Figur 3. Noder utför uppgifter som blivit definierade i deras programkod. Flera noder kan vara sammanlänkade och kommunicerar med hjälp av topics, en form av kanal där noder kan skicka och ta emot datastrukturer i form av meddelanden. Noder behöver inte ha information om att andra noder existerar, utan all kommunikation sköts av mastern som länkar upp noder till det topic som de vill prenumerera på, alternativt publicera till. Med andra ord behöver inte en nod veta vilken nod som publicerar dess indata, och inte heller vilken nod som prenumererar på dess utdata. Vill man koppla noder direkt till varandra, utan ett topic i mellanhand, kan services användas [20].

Denna modulbaserade uppbyggnad gör ROS flexibelt när ett system konstrueras. Utifrån det samlingsbibliotek som finns till ROS går det att bygga upp egna system utifrån redan skapade moduler.



Figur 3: Grafisk representation av hur kommunikation mellan noder sker i ROS.

I projektet användes ROS för att behandla och bearbeta den data som Kinect-kameran samlade in, och utifrån datan styra projektorn för att visa vad systemet såg och ville göra, samt om den ville instruera en människa till en viss handling.

2.6 Projektor

I projektet stod en projektor för kommunikationen från robot till människa. Den projicerade BelieveMes intention över ett stort område vilket gav en stor arbetsyta att förmedla information på.

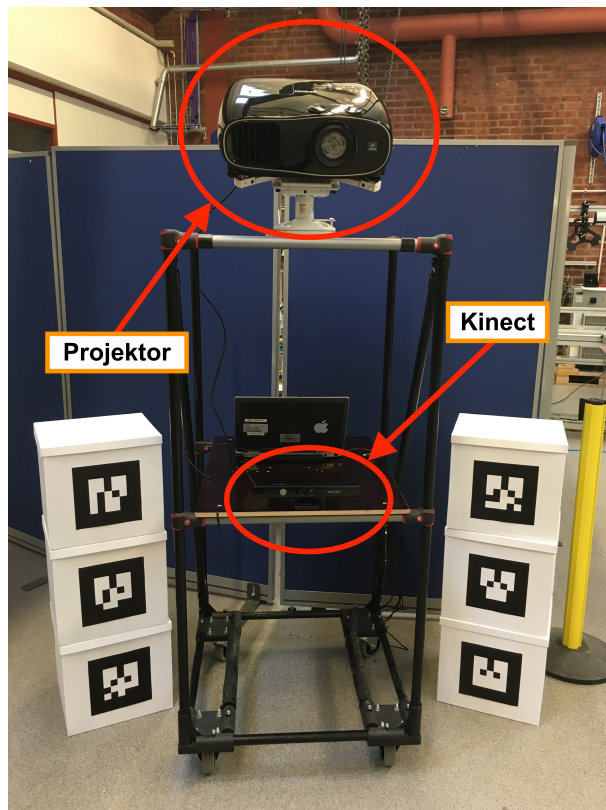
En projektor används vanligtvis för att projicera bilder på väggar, vilka är vinkelräta, vita ytor, i en mörk omgivning. Så var inte fallet under projektet, då projektorn var monterad med vinkel på en vagn och användes i väl upplysta rum med varierande projiceringsytor. För att kunna ge visuell information ställdes kraven att projektorn skulle ha stark ljusstyrka samt projicera färger som syns på alla ytor. I arbetet användes projektorn Epson EH-TW6600 [21].

Projektorn har en bildyta begränsad av dess maximala utlysningvinkel och dess ljusstyrka. Epson EH-TW6600 har en maxvinkel på $\sim 31^\circ$ [21], denna snäva vinkel gör att den möjliga storleken på den projicerade bildytan är starkt beroende av avstånd.

2.7 Förflyttning av BelieveMe

För att systemet skulle kunna flyttas runt under drift behövdes en mobil plattform, lämpligen en vagn. Val av vagn föll på en trilogicvagn på grunden att den är stabil, det går att anpassa utformningen av vagnen efter behov samt att det fanns en tillgänglig.

Två hyllplan fästes på vagnen, det övre hyllplanet för att projektorn skulle befinna sig högt över golvytan, och det nedre för att ställa Kinect-kameran på. Resultatet av konstruktionen demonstreras i Figur 4.



Figur 4: Vagnen efter ihopmontering med Kinect-kamera och projektor, samt lådor uppstaplade bredvid vagnen.

2.8 AssembleMe - Interaktion mellan människa och BelieveMe

För att kontrollera hur väl BelieveMe lyckats förmedla sina intentioner till en människa, skapades testprogrammet AssembleMe. AssembleMe går ut på att BelieveMe skall instruera en människa hur ett enkelt klosstorn skall konstrueras.

Testprogrammet börjar med att visa var på marken en låda skall placeras. När lådan placerats ut upptäcker BelieveMe att det tillkommit en låda och kan därefter bestämma om den tillkomna lådan har en acceptabel position. Om positionen ej accepteras av systemet fortsätter BelieveMe ge direktiv om vart lådan skall placeras. Om positionen accepteras projicerar BelieveMe nya direktiv om vart nästa låda skall placeras. Tekniken projection mapping tillåter projicering på såväl golvytor som lådornas sidor. Testprogrammet kommer att fortsätta enligt detta fram tills

alla lådor placerats och ett klosstorn har byggts. Om en position blockeras av andra lådor och därmed inte kan projiceras ut, skall AssembleMe instruera människan att flytta BelieveMe via en utprojicerad rutt tills positionen är inom räckhåll för projicering.

2.9 Krav och specifikationer för BelieveMe

För att kunna utveckla AssembleMe sattes följande krav på och specifikationer för BelieveMe:

1. Objekt och människor skall kunna identifieras
2. BelieveMe skall använda sig av realtidsdata från sensorer och länka denna med vad som projiceras
3. Kommunikation utåt skall ske med hjälp av projection mapping
4. BelieveMe skall lätt kunna flyttas av en vuxen människa
5. Det skall gå att flytta på BelieveMe utan att det påverkar dess funktion
6. Symboler och färger skall kunna tolkas av utomstående utan svårighet
7. Utomstående person skall kunna förstå BelieveMes intention utan insikt i programmet

Punkt 1-3 testas genom att BelieveMe skall kunna korrekt lokalisera människor som står framför Kinect-kameran, detsamma gäller för objekt. Objekten består, i det här fallet, av lådor med specifika markörer placerade på fyra sidor av respektive låda. BelieveMe skall kunna använda sig av indata ifrån Kinect-kameran för att bestämma rutt samt identifiera utplacerade hinder och lådor. BelieveMe skall kunna lysa upp en rutt i den 3D-miljö där den finner en eller flera människor samt projicera på ett objekt. BelieveMe skall även kunna projicera en bild på en yta oberoende av dess orientering. Projektionen skall följa lådans placering och rotation enligt tekniken projection mapping.

Vidare vagnen anses som transportabel, punkt 4, avgörs av huruvida alla medlemmar i kandidatgruppen kan flytta på plattformen utan problem. Därefter skall BelieveMe testas på om funktionen påverkas av att vagnen flyttas, punkt 5. Detta genomförs genom att test av punkt 1-3 görs för att sedan flytta vagnen i både position och vinkel och sedan köra testet av punkt 1-3 ytterligare en gång. Om testet efter flytten behåller samma egenskaper som testet innan är testet godkänt.

Punkt 6 och 7 testas genom att personer som är utomstående till projektet, det vill säga inte är medlemmar i kandidatgruppen, handledare, examinator eller på annat vis känner till projektets syfte, får pröva AssembleMe och får sedan genomföra enkäten (se appendix B) på AssembleMe.

3 Projektion av intention

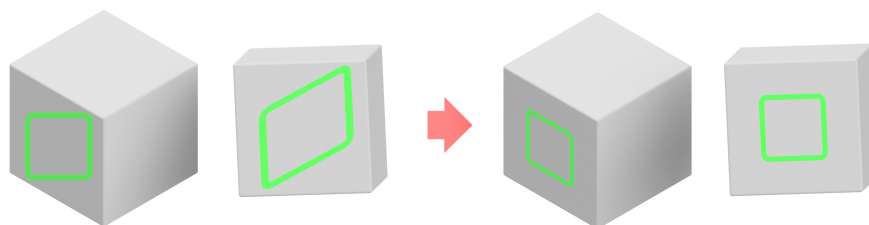
Detta kapitel presenterar hur, samt vad som projiceras för att på bästa sätt få kommunikationen att fungera. Inledningsvis förklaras vad projection mapping är och sedan hur problemet lösts, därefter beskrivs de undersökningar som gjorts för kommunikationen.

3.1 Projection mapping

Projection mapping, att projicera manipulerade bilder anpassade för olika djup, skapar en effekt av en oförvrängd bild på objektet sett med blotta ögat [22]. Tekniken används i dagens läge inom ett fåtal användningsområden. Till de vanligaste hör att skapa beundransvärda illusioner så som ”Vivid Sydney”s projektioner på Sydneys operahus i Sydney, Australien [23] och Disney Worlds projektioner på sagoslottet i Orlando, USA [24]. Det finns stora möjligheter för att användningen av projection mapping kommer öka och flera framtidsplaner på vad det skulle kunna användas till. Förutom trafik användningen (vidare diskussion angående möjlig användning i trafiken sker i avsnitt 3.3.2) finns det t.ex. en undersökning av hur införandet av projection mapping skulle kunna påverka handhygien på offentliga toaletter [25] eller hur projection mapping skulle kunna användas till att öka turism i staden Esfahan [26].

För att kommunicera vad BelieveMe såg, samt vad dess avsikt var, behövde detta presenteras för människor. En vanlig projektor användes för att kunna kommunicera visuellt, genom att visa tvådimensionell grafik på markplan eller på objekt.

Den projicerade bilden uppfattas med hjälp av projection mapping som att den hör hemma i en tredimensionell miljö, trots att bilden som projiceras i själva verket är tvådimensionell. Projektorn behöver inte heller befinna sig ortogonalt mot ytan. För att genomföra detta behövdes programvara för att modifiera utdatan vilken manipulerar bilden för att ge önskat resultat. Denna teknik användes med fördel i projektet för att visa tecken och symboler på marken utan att behöva montera projektorn rakt ovanifrån. Den användes dessutom för att projicera symboler och text på objekt med flera olika djup så som en kub från olika vinklar, se Figur 5.



Figur 5: Bild projicerad på låda före respektive efter projection mapping

I projektet har ett program skapats som hanterar denna projection mapping. Programmet är utformat så att det använder sig av indata i form av hörnen på det man vill rita ut med tredimensionella koordinater. Koordinaterna i rummet är alltid relativt Kinect-kameran vilket gör den oberoende av var BelieveMe befinner sig i rummet.

3.1.1 openFrameworks

För att utföra bildbehandlingen användes verktyget openFramework [27]. OpenFrameworks är öppen källkod och skrivet i C++ och har dessutom ett antal tillägg utöver dess grundfunktioner som underlättar manipulerandet av bilder.

Ett eget vidareutvecklat tillägg, *ofxRO-Mapping* [28], skapades för att anpassa behovet av att kunna manipulera bilder genom att bestämma respektive hörns placering i programkod.

3.1.2 Beskrivning av funktion

Indatan, till programmet *3dTo2dMappingOF* [29] som skapades för denna projection mapping, utgörs av den behandlade datan från Kinect. Som kommunikationsväg används en websocket-server [30] som körs via ROS. Bildprogrammet ansluter sedan till denna och lyssnar efter ändringar från Kinect-programmen. Denna data är fördefinierad i AssembleMe av fyra koordinater, en för varje hörn av en yta, samt ytterligare två variabler för att ange vad som ska projiceras. Alla dessa koordinater beskrivs relativt Kinect-kamerans position. Dessa koordinater transformeras enligt avsnitt 3.2 för att avbildas på ett plan som motsvarar vad projektorn ska projicera, för att sedan manipulera bildens hörn till de motsvarande positionerna.

3.2 Mappning av 3D-koordinater till ett 2D-plan

Syftet med avbildning var att projektorn skulle projicera vad denna såg utifrån sitt perspektiv. Eftersom informationen att tillgå bestod av koordinater i \mathbb{R}^3 kunde detta simuleras på godtycklig plats i rummet. Detta genom att definiera en simulerad skärm som i sin tur definierade ett plan och utifrån ett avstånd bestämde projektorns position. Punkten som skulle avbildas och kamerans position definierade en linje, vars skärningspunkt på planet motsvarade den punkt på planet i \mathbb{R}^3 som skulle ritas ut. För att göra det möjligt att beräkna punkten detta skulle innebära i \mathbb{R}^2 utnyttjades vinkeln mellan överkanten av den simulerade skärmen samt punkten. Dessutom var man tvungen att ta hänsyn till riktningarna på normalerna för att bestämma tecknet på vinkeln.

3.2.1 Teori

Låt \mathbf{B} definiera hörnen på en rektangel enligt Figur 6 som representerar en skärm. För att kunna veta var kameran är placerad, måste mittpunkten M av denna yta som begränsas av hörnen, samt ytans riktning N , tas fram.

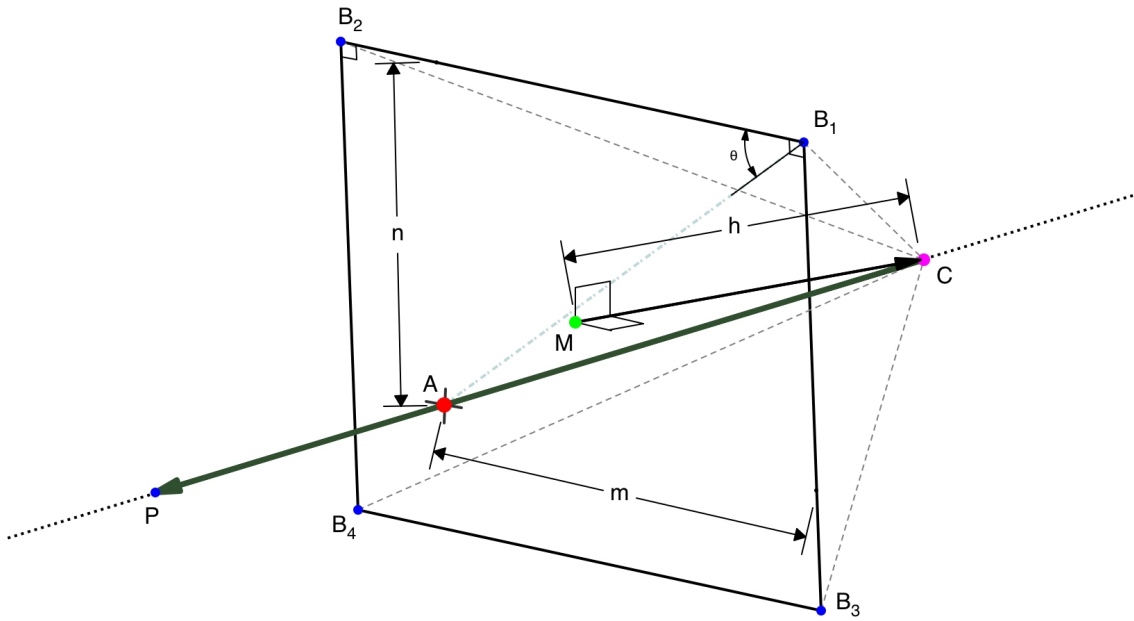
$$M(x, y, z) = \left(\frac{x_{B_2} + x_{B_3}}{2}, \frac{y_{B_2} + y_{B_3}}{2}, \frac{z_{B_2} + z_{B_3}}{2} \right) \quad (1)$$

Därefter räknas kamerans position fram med hjälp av ekvation (3) där h representerar avståndet från kameran till planet.

$$N(a_n, b_n, c_n) = \frac{(\vec{B_1B_2}) \times (\vec{B_1B_3})}{\|(\vec{B_1B_2}) \times (\vec{B_1B_3})\|} \quad (2)$$

$$C = M + hN \quad (3)$$

Linjen som följer vektorn \vec{PC} följer av ekvationen (4) och korsar planet vid någon punkt A (förutom vid fallet att de är parallella), denna punkt $A(x, y, z)$ räknas fram i ekvation (10) med hjälp av ekvation (9) och (2). Denna punkt är alltså skärningspunkten på planet beskrivet i rummet.



Figur 6: Illustration över hur punkten P avbildas till ett plan för att kunna reproducera hur det skulle se ut på en skärm med projektorn i punkten C

$$\frac{x - x_p}{x_c - x_p} = \frac{y - y_p}{y_c - y_p} = \frac{z - z_p}{z_c - z_p} = k \quad (4)$$

Den kan även skrivas i parameterform enligt (5), (6) samt (7).

$$x = k(x_c - x_p) + x_p \quad (5)$$

$$y = k(y_c - y_p) + y_p \quad (6)$$

$$z = k(z_c - z_p) + z_p \quad (7)$$

Dessa variabler stoppas sedan in i planet ekvation och resulterar i (9). Där D definieras utifrån en godtycklig punkt på planet $N(x, y, z)$, i detta fall väljs B_1 , som beskrivs av planet ekvation i (8).

$$D = a_n x_{B_1} + b_n y_{B_1} + c_n z_{B_1} \quad (8)$$

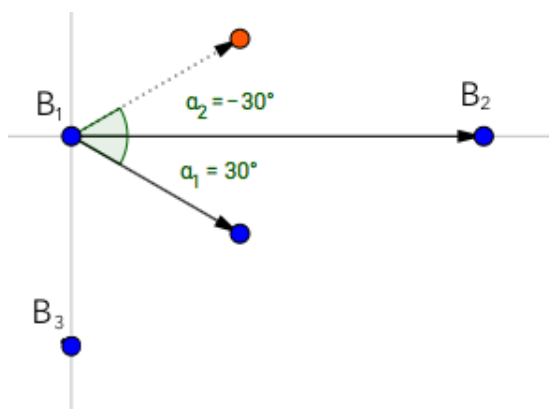
$$\Rightarrow k = \frac{D - (a_n x_p + b_n y_p + c_n z_p)}{a_n(x_c - x_p) + b_n(y_c - y_p) + c_n(z_c - z_p)} \quad (9)$$

Korsningspunkten resulterar i punkten $A(x, y, z)$ enligt (10).

$$A = (k(x_c - x_p) + x_p, k(y_c - y_p) + y_p, k(z_c - z_p) + z_p) \quad (10)$$

För att sedan räkna ut var punkten skall ritas ut på skärmen används vinkeln θ i planet mellan vektorerna $B_1\vec{B}_2$ och $B_1\vec{A}$ som ges av ekvation (11). Det är däremot omöjligt utifrån denna information att redogöra det ursprungliga tecknet på θ vilket kan ses i Figur 7.

$$\theta = \arccos(\cos(\theta)) = \arccos\left(\frac{B_1\vec{B}_2 \cdot B_1\vec{A}}{\|B_1\vec{B}_2\|\|B_1\vec{A}\|}\right) \quad (11)$$



Figur 7: Problem som uppstår när man vill komma åt $\text{sgn}(\theta)$ i $\theta = \arccos(\cos(\theta))$

För att ta reda på detta utnyttjas högerhands-regeln [31] samt att punkten A och punkterna B ligger på samma plan. Detta innebär att normalerna endast kan ha antingen samma eller motsatt riktning på grund av de är parallella. I ekvation (12) tar man enkelt fram tecknet för vinkeln θ_s som har stor betydelse för att få korrekt värde i höjled i ekvation (15).

$$s = \frac{B_1\vec{B}_2 \times B_1\vec{B}_2}{\|B_1\vec{B}_2 \times B_1\vec{B}_2\|} + \frac{B_1\vec{B}_2 \times B_1\vec{A}}{\|B_1\vec{B}_2 \times B_1\vec{A}\|} - 1 = \begin{cases} -1, & \text{motsatt riktning} \\ 1, & \text{samma riktning} \end{cases} \quad (12)$$

$$\theta_s = s\theta \quad (13)$$

Slutligen kan man enkelt att plocka ut koordinaterna på planet från punkten B_1 vilket representerar vänstra övre hörnet $(0,0)$ på skärmen, där .

$$m = \|B_1\vec{A}\|\cos(\theta_s) \quad (14)$$

$$n = \|B_1\vec{A}\|\sin(\theta_s) \quad (15)$$

3.2.2 Förskjutning av projektionsplan

Då projektorn projicerade sin bild mot ett tänkt plan i \mathbb{R}^3 , behövde detta plan placeras i det virtuella rummet så att det motsvarade projektorns verkliga placering i rummet i förhållande till kamerans koordinatsystem. Detta gjordes genom att ta hänsyn till projektorns utlysningsvinkel, vinkel ner mot golvet samt avståndet till kinectkameran.

Planets avstånd d till kameran bestämdes av projektorns utlysningsvinkel α som gav storleken på bildyta på ett givet avstånd. Utlysningsvinkeln definierades som vinkeln mellan den

projicerade bildytans mittpunkt och sida i horisontalld. Bildplanet gavs storleken bredd×höjd = $w \times h = 1 \times \frac{9}{16}$ för att överensstämna med projektorns bildförhållande 16:9, vilket ger:

$$d = \frac{0.5w}{\tan \alpha} \quad (16)$$

Då projektorn var placerad högre upp än kinecten var projektionsplanet tvunget att också förflyttas uppåt i höjddled. Projektorn var dessutom riktad nedåt vilket innebar att förskjutningen var beroende av avståndet mellan projektionsplan och projektor samt vinkeln på projektorn. Avståndet mellan kinect och projektor betecknas d_{kp} och avståndet d ges av ekvationen ovan. Den önskade förskjutningen i höjddled är y . Vilket ger:

$$y = d_{kp} + ad \quad (17)$$

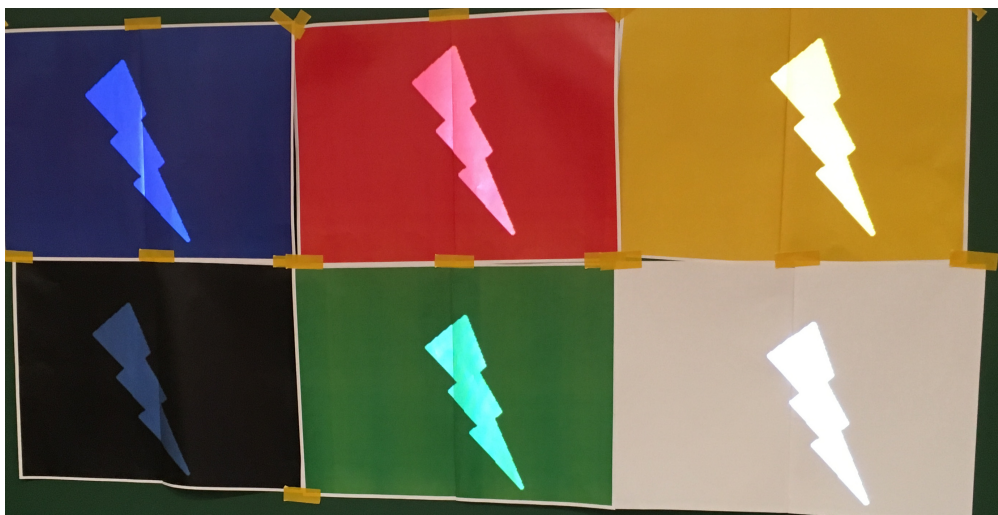
där a motsvarar punkterna som definerar planet flyttat längs normalen till planets riktning.

3.3 Visualisering av intention

För att fastställa vad som skulle projiceras togs ett antal faktorer, så som synlighet och uppfattning, i beaktning. I detta delkapitel presenteras undersökningar och tester som gjorts för att uppnå maximal kommunikationsförmåga.

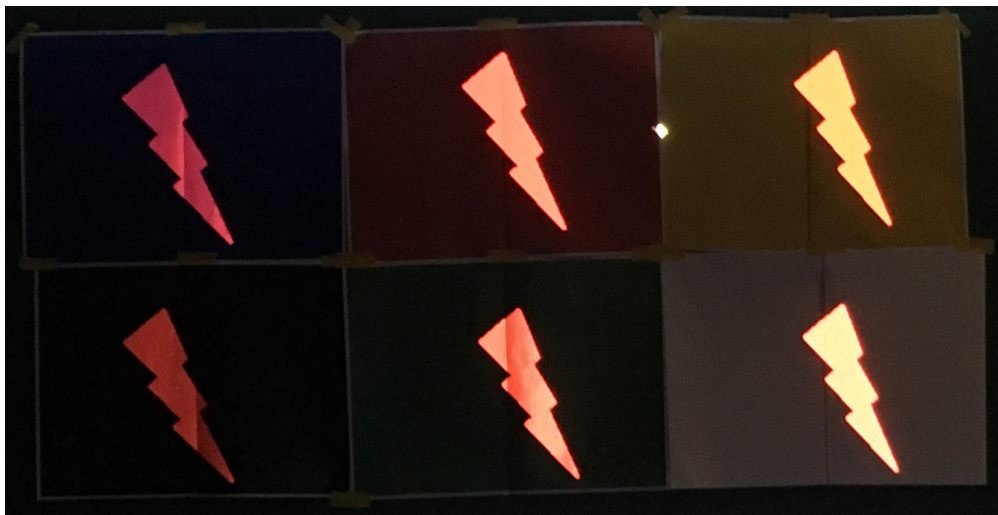
3.3.1 Test av färger med hjälp av projektor

För att avgöra vilka färger som syns bäst mot olika typer av ytor genomfördes ett test. Testet gick ut på att undersöka hur bra olika färger syns vid olika ljusstyrkor och bakgrunder, genom att testa färgerna mot olika ytor i olika ljus. Detta för att sedan kunna poängsätta färgerna. Ljusstyrkan i rummet varierades tre gånger med värden från 14 till 394 lux⁷ för att testa olika tänkbara ljussituationer. För att testa olika ytor att lysa på användes stora färgade pappersark, se Figur 8 och Figur 9.



Figur 8: *Projicering av vitt ljus vid 112lux*

⁷Mätningarna gjordes med en ljusmätare av typen Sekonic L-308S Flashmate. Ljussmätarens felmarginal är liten nog att bortse ifrån.



Figur 9: *Projicering av rött ljus vid 14lux*

Hur bra projektionen syntes på de olika färgade bakgrunderna bedömdes av medlemmar i kandidatgruppen med hjälp av en skala från ett till fem, där ett var mycket otydligt och fem var mycket tydligt. Poängen för varje färg räknades sedan ihop. Färgen som fick högst poäng är den som är mest synlig mot alla bakgrunder.

Den totala poängen från projicerad färg blev för; grön 61, röd 58, blå 41, vit 70 och gul 75, se Tabell 1. Resultatet från denna undersökning är att gul är den optimala färgen att projicera med gällande synlighet på varierande bakgrunder.

Fler faktorer än vilken färg som är tydligast var dock tvunget att tas med i beräkningarna, t.ex. hur människor reagerar och uppfattar de olika färgerna.

Tabell 1: *Sammanställning av poäng för de olika färgerna från testerna i appendix C. Kolumnerna representerar de projicerade färgerna och raderna representerar bakgrundsfärgerna där poängen från de olika ljusstyrkorna adderats.*

	Röd	Grön	Blå	Vit	Gul
Röd	13	7	5	11	13
Grön	10	14	6	13	14
Blå	5	8	9	11	11
Vit	12	14	11	15	15
Svart	7	5	4	7	7
Gul	11	13	6	13	15
Totalt	58	61	41	70	75

3.3.2 Val av visualisering för att skapa tilltro och trygghet

I rapporten *Communicating robotic navigational intentions* [32] redogörs en studie på hur en självkörande rullstol byggts och projicerar den bana rullstolen planerar att köra längs en korridor. Detta för att människor skall kunna anpassa sig efter rullstolen, och jämför sedan detta med om samma rullstol istället får köra utan att projicera någon ruttplanering. Rapporten visar på hur människor känner sig tryggare i det autonoma fordonets närvaro om fordonet projicerar ut

sina intentioner, något som är jämförbart med om personerna sett en annan gående persons kroppsspråk.

Vid implementering av projicering behövdes det hållas i åtanke att människan inte kommer att kunna veta om BelieveMe uppfattat den eller inte, utan bara vart BelieveMe vill bli förflyttad till. Därför var ett alternativ att projicera en cirkelsektor eller liknade i fotgängarens riktning, för att påvisa vart BelieveMe inte skulle köra. I detta fall kommer dock inte människan ha någon aning om vart BelieveMe tänkt köra och det skulle därför bli svårt för människan att planera sin rutt.

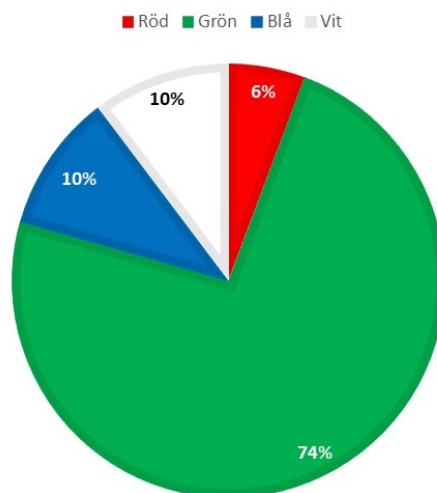
Det som tidigare nämnts är färger som människor reagerar bättre och sämre på. Reaktionen är ofta en kulturell fråga, vad en person i fråga fått lära sig att de olika färgerna representerar. På grund av avgränsningen att endast titta på Sverige diskuterades inte kulturfrågan vidare, utan färgvalet i Sveriges trafik sågs som en bra utgångspunkt. Trafikljus har färgkoordinationen enligt; rött - stopp, gult - stanna, grönt - tillåtet att köra. Likaså gäller för trafikskyltar; de röda och gula skyltarna kräver ofta uppmärksamhet såsom varningsmärken (röd, gul och svart) [33], stoppskylt (röd och vit) och väjningsplikt (röd och gul) [34].

3.3.3 Undersökning om hur BelieveMe kan skapa tilltro och trygghet

För att få en förståelse om vad omgivningen uppfattar genomfördes en undersökning. Undersökningen bestod av 9 frågor vilka utredde hur de svarande kände inför självkörande fordon i samhället och hur de skulle uppfatta olika färger. Av de 156 personer som svarade på enkäten svarade 153 stycken på alla frågor.

Självkörande fordon valdes som ett exempel många kan relatera till inom människa till robot kommunikation. De tre första frågorna, samt delvis fråga fyra, fokuserade på inställningen till att ha ett självkörande fordon i sin närvaro. För att se frågorna och svaren se appendix A. Det visade sig att av de svarande litade ungefär hälften på att de inte skulle kunna bli påkörda av ett självkörande fordon, samt att mer än hälften skulle känna sig trygga och kunde även tänka sig att äga ett självkörande fordon. Nästan hälften av de svarande ansåg att lysa på marken framför människan var det, av de givna, bästa alternativet för att de skulle känna sig säkra. Endast 13 % svarade att de inte skulle kunna lita på tekniken och 15,5 % skulle känna sig trygga och lita på systemet utan någon kommunikation. Eftersom det finns en stor chans att självkörande fordon finns på vägarna inom en relativt snar framtid är det viktigt att alla människor kan känna sig trygga i dess närvaro. Svaren visar på hur viktigt det är att jobba på kommunikationen mellan människa och robot för att få människan att inte anse den som ett hot, och slutligen att den känns vardaglig.

Fråga fyra till nio i undersökningen fokuserade på hur olika typer av kommunikation uppfattas. Sammanfattat upplevdes grönt till största del som en betryggande färg i anseendet att man känner sig sedd och att det inte har skett något fel, se Figur 10. Det motsatta gäller för röd. Där uppfattade de tillfrågade färgen som en varningsfärg, att något gått fel. Angående färgerna blå och vit skapade de mest förvirring bland de svarande, mer än hälften svarade "Vet ej". Blå och vit är då inte så bra att använda vid projicering av cirkelsektor mot fotgängare. Undersökningen resulterade i att grön är den bästa färgen för att projicera att BelieveMe ser människan. Det mest föredragna alternativet var att BelieveMe skulle projicera ut den planerade rutten.



Figur 10: Vilken färg de svarande ansåg betyda att fordonet sett personen och att fordonet tänker lämna företräde. Resultat tagna från fråga 5-8 i undersökningen som presenteras i Appendix A, de svarande hade i och med detta möjlighet att svara flera alternativ på den uppsädda frågan.

3.3.4 Sammanställning av färgval

De färger som syntes bäst oavsett bakgrund och omgivningens ljusstyrka var gult, vitt och grönt. Eftersom enkätundersökningen visade på att människor inte förstår vad den autonoma enheten försöker förmedla om den projicerar vitt ljus, är gult eller grönt ljus att föredra. I enkäten ingick inte någon fråga om hur de efterfrågade skulle reagera på färgen gult, men utifrån vad de svarade på rött och grönt, och med färgkodningen i Sveriges trafik i åtanke, antas gult ge en varnande effekt. Det alternativ som förefaller lämpligast för att påvisa att BelieveMe sett en människa och inte kommer att köra på den är därför grönt ljus.

Vad gäller färgval för projicering på klossar används rött för att påvisa att en kloss är på fel plats och grönt när den är på rätt plats. Detta färgval baseras återigen på enkäten där resultatet blev att rött ljus uppfattades som varnande och att något är fel medan grönt ger en betryggande effekt av att något är rätt.

3.3.5 Visualisering vid AssembleMe

Vid AssembleMe ligger huvudsyftet i att skapa förståelse bland människor för vad BelieveMe försöker förmedla. BelieveMe skall först få en människas uppmärksamhet, sedan ge denne information om vad som skall utföras och slutligen, tala om för människan att den har genomfört uppgiften korrekt. För att få en människas uppmärksamhet kan projicering på eller vid denne vara effektivt. När BelieveMe väl har fått människans intresse måste det vara tydligt vad som förväntas av människan. Att projicera en animation bör göra instruktionerna tydligare än om bara en bild visas, eftersom animationer är rörliga och på så sätt kan förmedla mer information.

Eftersom BelieveMe i det här fallet skall få en människa att flytta på klossar kan det vara fördelaktigt att ha en animation som visar en kloss som flyttas. Alternativt kan pilar, eller liknande, vara att föredra. Huvudsaken är att budskapet om vart och hur klossen skall placeras förmedlas. I allmänhet är det lättare att förstå symboler som man känner igen och kopplar till något [35]. Dessutom är enkla figurer att föredra eftersom detaljer kan göra symbo-

len/animationen rörig och otydlig. Pilar och klossar är därför ypperliga kandidater till projiceringen.

På grund av begränsningar i programvaran, användes endast stillbilder. En solid grön kvadrat för att visualisera var en kloss skall placeras medan resterande budskap, vilken box det är som skall flyttas och om denna box är på rätt eller fel ställe, förmedlas med en grön respektive röd kvadratisk ram. Det första som händer i AssembleMe är alltså att en grön kvadrat projiceras där BelieveMe vill att första klossen skall placeras. Samtidigt blir samtliga klossar som befinner sig inom projektionsytan, men på fel plats, upplysta med en röd kvadratisk ram. När människan, som för tillfället interagerar med BelieveMe, har placerat den röda klossen på rätt plats blir denna låda grön. För att visa att BelieveMe har sett en människa projiceras en grön ring runt människans fötter. Ringen kommer att följa efter människor vart de än går, så länge de inte går utanför BelieveMes räckvidd eller ställer sig bakom något som skymmer deras profil.

Om BelieveMe vill projicera på en kloss som för tillfället inte är inom dess projiceringsområde kommer pilar att lysa upp vägen som BelieveMe vill att människan skall förflytta den. När klossen väl är inom synhåll kommer BelieveMe att visualisera vilken kloss och vart den skall flyttas enligt ovanstående.

4 Programvaran bakom BelieveMe

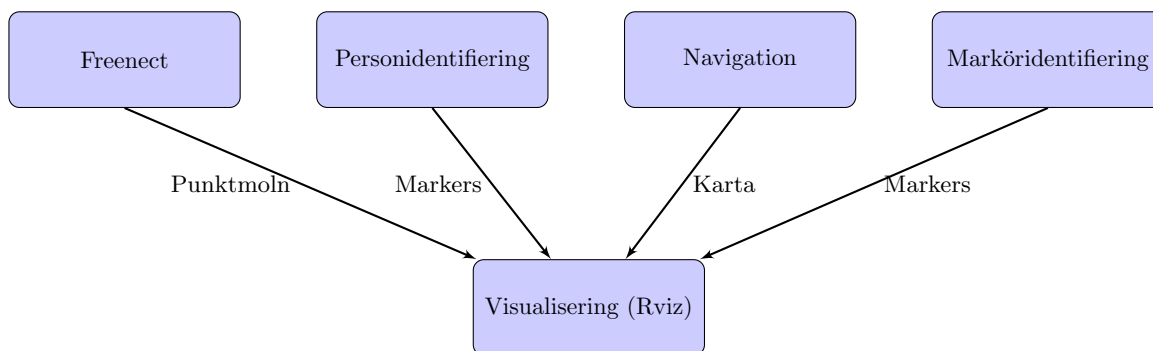
Detta kapitel klargör för hur BelieveMe bearbetar data. Programmen som identifierar personer och lådor samt planerar rutter beskrivs och förklaras.

4.1 Input med hjälp av Kinect

Mycket av den information som fås ut av Kinect-kameran finns redan inbyggt i dess mjukvara, såsom djup- och positionsbestämelse utifrån triangulering, skelettspårning samt att se skillnad på olika objekt [36][37]. Kommunikation mellan kameran och ROS behövs för att få tillgång till datan, och samtidigt konvertera strömmen av rådata till andra dataformat (djupbilder, punktmoln) som sedan används av efterkommande program. Till detta användes Freenect, en existerande programvara från andra ROS-användare, som ger önskad kommunikation och konvertering.

4.2 Rviz

Rviz [38] är ett program inom ROS-miljön för att visualisera data så som punktmoln och 3D-modeller. Genom att lyssna på topics från de noder man är intresserad av går det att ta till sig information från många olika noder samtidigt. Detta gör det enkelt att få en överblick över både de fysiska objekt i rummet som fås genom data från Kinecten, och de objekt som bara existerar virtuellt. Den värld som blir resultatet av denna syntes mellan verklighet och modell är just den värld som ett system som styrs av ROS uppfattar, och visualiseringen har använts i projektet som ett verktyg för felsökning och kalibrering. Figur 11 visar vilka noder i BelieveMes mjukvara som kommunicerar med Rviz och vilken data de önskar visualisera.



Figur 11: Noder kommunicerar med Rviz för visualisering av data.

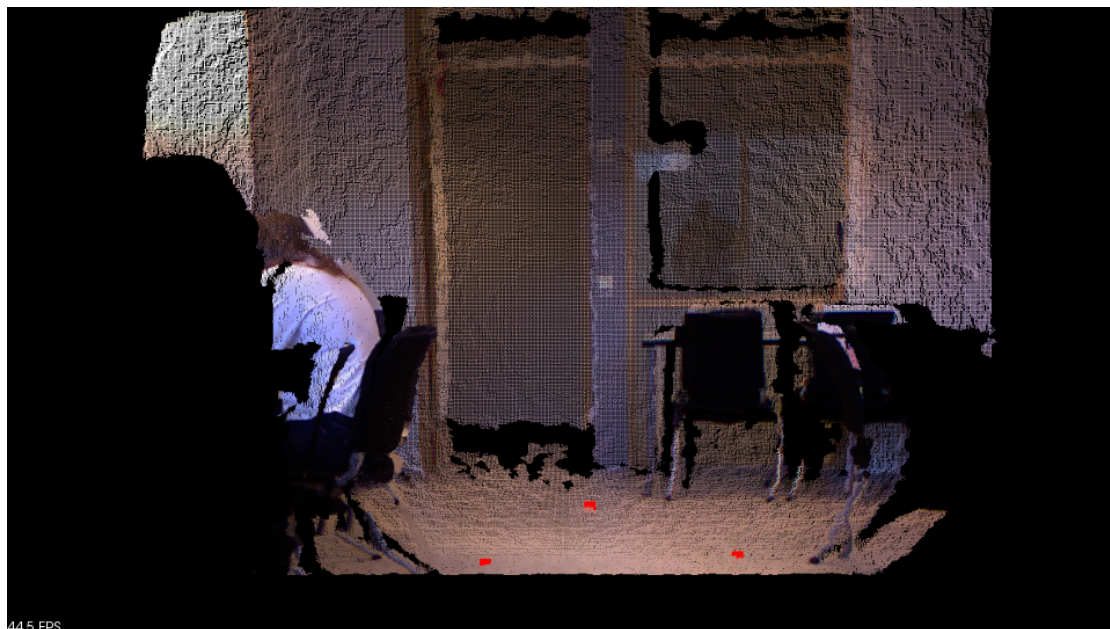
4.3 Personidentifiering

För att interaktioner mellan människor och robotar skall fungera behöver de veta vart motparten befinner sig. Människor upplever detta som en självklarhet, men att identifiera en människa är inte lika enkelt för en robot. För att ge BelieveMe möjlighet att kunna identifiera människor användes ett personidentifieringsprogram *ground-based-rgb-d-people-detector* skrivet i C++ [39], som återfinns på Github [40].

Det första programmet gör är att skapa en prenumerant till topic *camera/depth_registered/points* som publiceras av Freenect, vilket ger tillgång till Kinect-kamerans

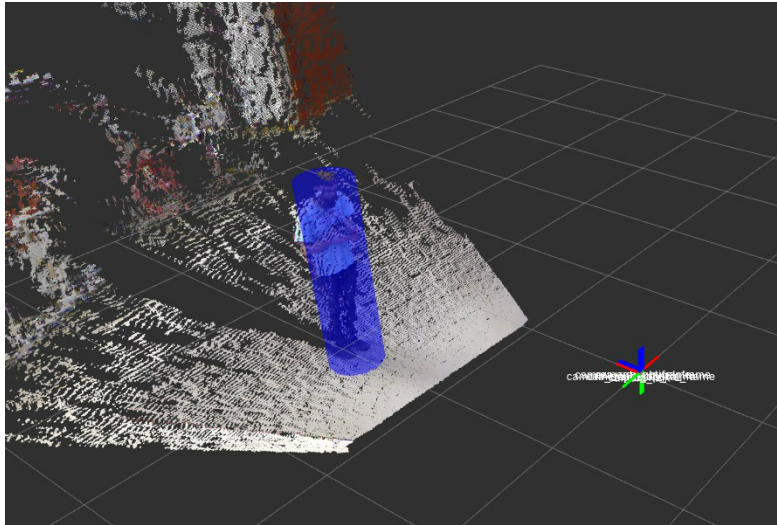
punktmoln. En callback registreras till denna prenumerant vilken berättar för programmet att ett nytt punktmoln finns att hämta.

När första punktmolnet blir tillgängligt för programmet kommer första bilden att frysas och visualiseras. Användaren förväntas därefter markera 3 punkter i golvet för att kunna utföra så kallad grundplan-estimering, se Figur 12, vilket används för att kunna detektera människor i förhållande till grundplanet (golvet).



Figur 12: Initial grundplans-estimering genom att använda 3 punkter på golvet.

Därefter påbörjas detektionen av människor genom att analysera punktmolnet utifrån vissa parametrar (minimumhöjd, maximumhöjd,..) och sparar potentiella människors clusters. Bara clusters ovanför en viss säkerhetsmarginal anses vara människor, och programmet räknar upp hur många människor som detekterats, och utifrån clusterns medlemsfunktioner går det att få fram egenskaper så som en människas position. Cylindriska markers publiceras till topic *visualization_marker_array* vilket visualiseras med hjälp av Rviz, se Figur 13. Samtidigt publiceras fyra hörn runt varje position till topic *ground_based_rgb_people_detector/PeopleCorners* vilket skapar en ruta på 6x6 dm runt varje person för att kunna användas till projection mapping.



Figur 13: Detekterade människor visualiseras som cylinder-markers i Rviz.

En människa bör befinna sig några meter från Kinect-kameran för att utifrån personens längd kunna analysera om det är en människa eller ej. På grund av detta fungerar ej personidentifieringen optimalt i små utrymmen, utan en öppnare yta är att föredra. För att underlätta detta kan Kinect-kameran placeras på en högre höjd och nedåtvinkel mot golvet, då man behöver ett längre avstånd till den om Kinect-kameran befinner sig på lägre höjd.

Då programmet främst använder storleken på en cluster för detektering av människor, händer det ibland att programmet felaktigt identifierade till exempel soptunnor som människor. Detta kan förhindras med att ändra på marginalerna på detekteringsgränserna, med avvägning mot att en mindre andel av människor med viss längd kan identifieras. I AssembleMe var denna fel-detektering inget problem då projiceringsytan var fri från övriga objekt som felaktigt skulle kunna identifieras som människor.

Programmet har en uppdateringsfrekvens som varierar mellan $18-20\text{ Hz}$ under programkörning, vilket är en god prestanda för detta ändamål. Dock upptäcktes det att frekvensen gick ned till under $5-6\text{ Hz}$ när samtliga program kördes samtidigt. Detta beror på att övriga program dessutom skall dela på samma beräkningsresurser. Detta kan förhindras med en kraftfullare dator, alternativt dela upp programkörningen på två ihopkopplade datorer.

4.4 Navigation

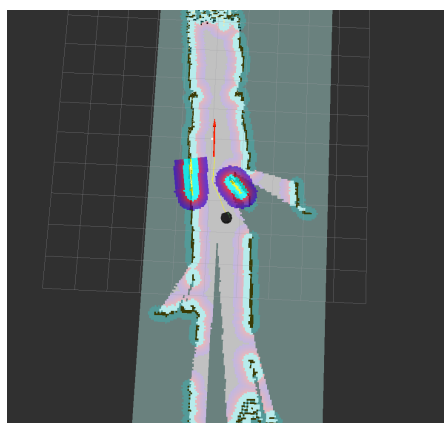
För att kunna navigera i ett rum med hjälp av Kinect-kameran, var *rtabmap_ros* [41] menat att användas, vilket inkluderar metoder för att bestämma BelieveMes position samt ruttplanering. I skrivande stund fungerar dock inte navigationen tillsammans med AssembleMe.

Med hjälp av Kinect-kameran som input skapar programmet en 2D-karta av rummet som den befinner sig i, genom att föra runt kameran i rummet, se Figur 14a. Kinect-kameran fungerar vid planläggning av rummet som en laserscanner, trots att den inte har någon laser, genom att datan som samlas in av Kinect-kameran behandlas och endast tre pixlar i höjddled av punktmolnet extraheras för att simulera en laserscanner, och skapar därefter 2D-kartan med denna. Kameran skapar sin egen odometri genom att jämföra ändringar i det nuvarande punktmolnet mot det föregående, och således kan den räkna ut sin nuvarande position utifrån föregående. Det är också möjligt att ladda in tidigare sparade kartor. Därefter skall Kinect-kameran återigen flyttas

runt tills den kan lokalisera sig i kartan. Då kan man införa målpunkter och programmet skapar en rutt utifrån dess nuvarande position till målet som inte kolliderar med några hinder, se Figur 14b. Målpunkten går att sätta både manuellt och via ett topic `/move_base_simple/goal` med meddelandetyp `PoseStamped` [42]. Punkterna i denna rutt går att extrahera från topic `/move_base/DWAPlannerROS/global_plan` med meddelandetyp `Path` [43]. Ett egetkonstruerat program `path_corner_creator` används till detta. Programmet prenumererar på topic `/move_base/DWAPlannerROS/global_plan`, delar jämnt upp rutten i maximalt 10 punkter beroende på längd och räknar ut 4 hörn runt dessa punkter för att skapa 2x2 dm stora rutor. Dessa hörn publiceras till topic `path_corner_creator/NavigationCorners` för att användas till projection mapping. Visualiseringen sker via Rviz, där en Turtlebot-robot [44] symboliserar BelieveMe för att underlätta visualisering av dess position.



(a) Karta över korridoren skapas under mappning-fasen.



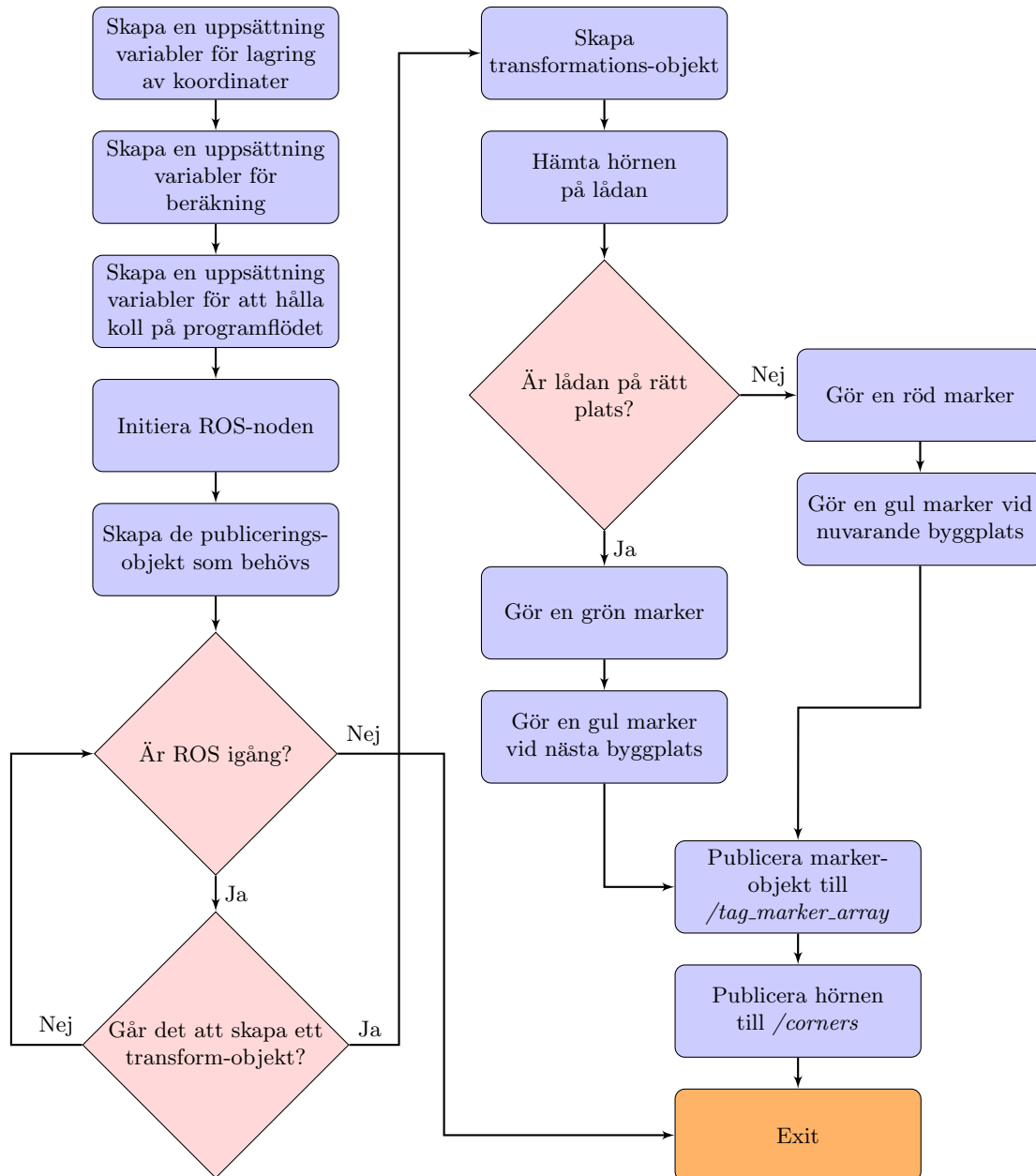
(b) BelieveMe har lyckats lokalisera sig i korridoren, och ritar ut en rutt (gul) för en utsatt målpunkt (röd).

Figur 14: Mappning och navigation

BelieveMe kan ej förflyttas för snabbt, då detta kan medföra att odometrin går förlorad. För att återfå lokaliseringen skall BelieveMe återföras till den position den hade när odometrin gick förlorad. Detta sker främst när punktmolnet från kameran kan motsvara flera möjliga positioner, till exempel om BelieveMe står väldigt nära en vägg. Det är därför inte lämpligt att navigera den i trånga utrymmen, utöver det faktum att Kinect-kameran inte kan uppfatta punktmoln som ligger för nära.

4.5 AssembleMe

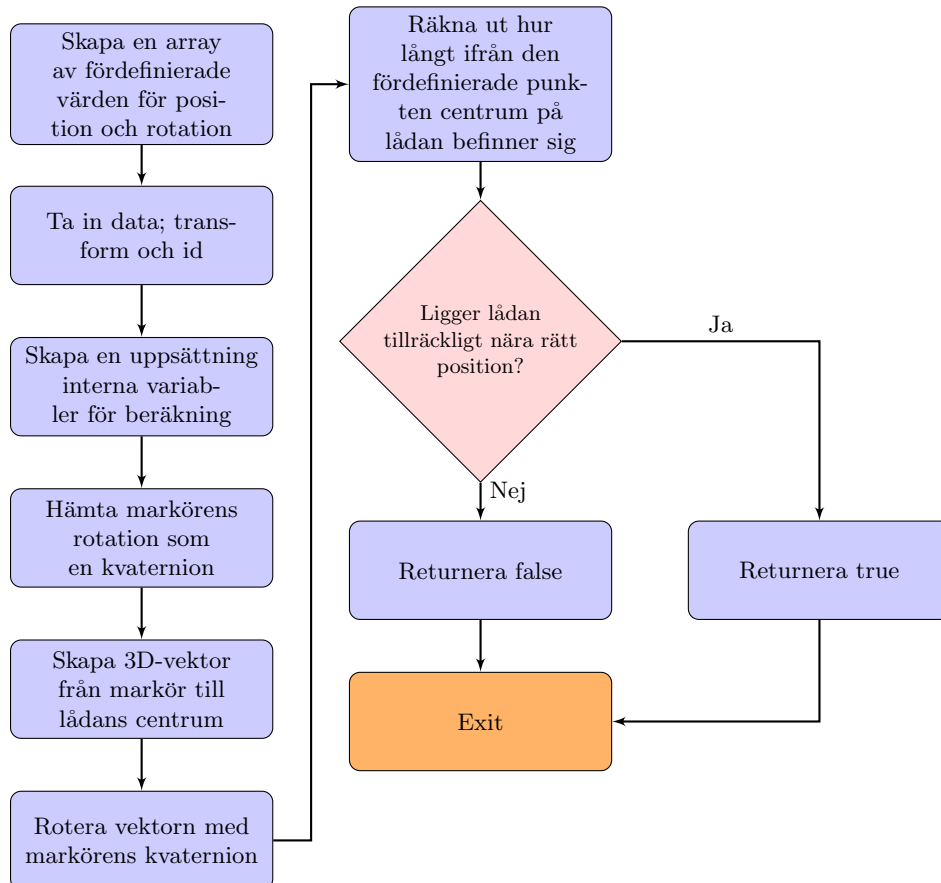
För att knyta samman marköridentifieringen, kontrollen av lådans position och publiceringen av hörn, är ett huvudprogram [45] skrivet där en överblick beskrivs i flödesschemat. Vidare beskrivningar för varje del ges under kapitel 4.5.1-4.5.3.



4.5.1 Kontroll av lådas position

Metoden *boxInCorrectPlace()*, som hanterar verifieringen av lådas position i världen, använder en del metoder från såväl ROS och C++.

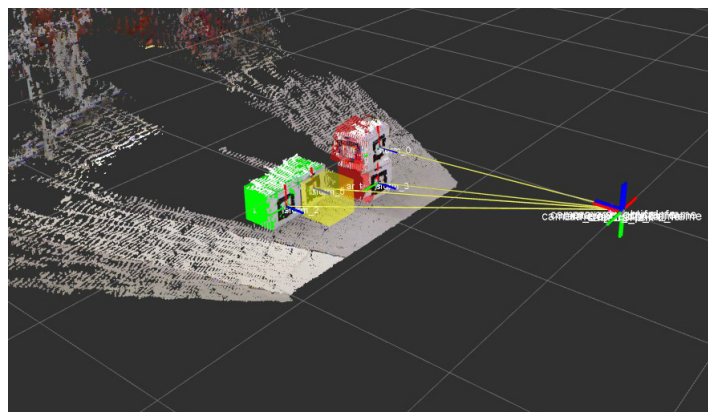
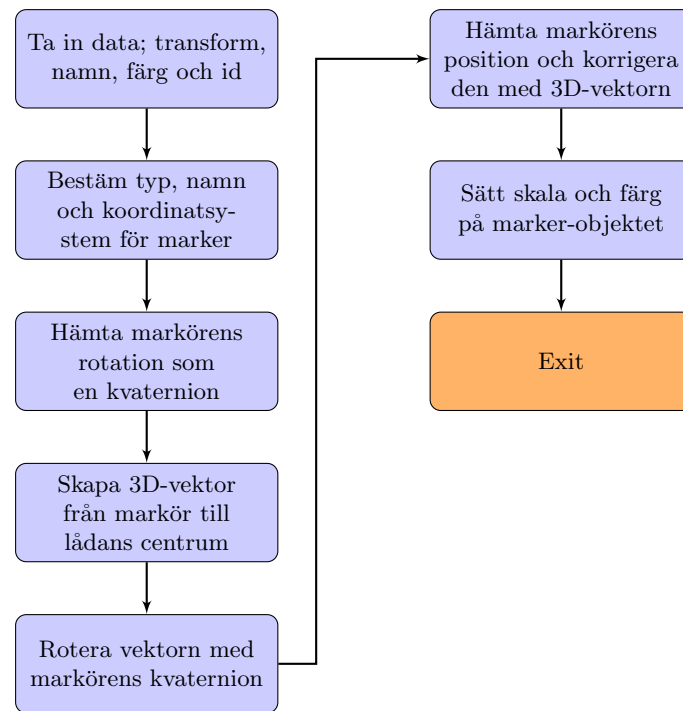
Genom att skapa en vektor från den detekterade lådas centrum till nästa tänkta position för klosstornet och sedan beräkna längden på denna, kan vi avgöra om lådan ligger tillräckligt nära positionen för att det skall kunna räknas som korrekt. För att denna vektor skall följa markörens rotation multipliceras den med dennes kvaternion⁸.



⁸Se Appendix D.

4.5.2 Marköridentifiering

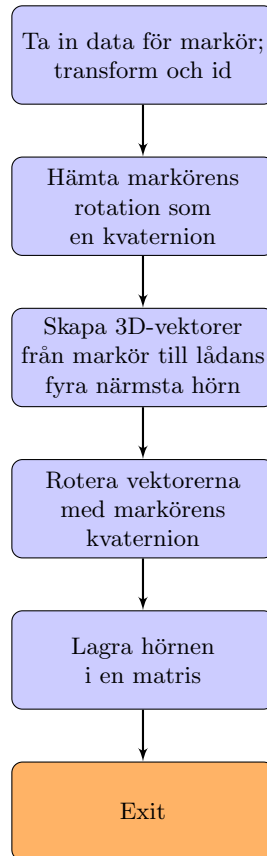
För att virtuellt visualisera AssembleMe kommer markers som visualiserar lådorna att användas. Metoden `makeMarkerArray()` hanterar placeringen av dessa är förhållandevis simpel och använder även den många av de inbyggda metoder som finns i ROS, Eigen [46] och såklart C++. För att veta vart lådorna befinner sig används en modifierad version av `ar_track_alvar` [47] som med hjälp av Kinect-kameran kan upptäcka markörens pose. Önskad funktionalitet återfås med hjälp av ett flertal egna metoder och algoritmer vars användning finns beskriven nedan. Hur programmet ser ut i rviz återfinns i Figur 15.



Figur 15: Lådornas positioner visualiseras i rviz.

4.5.3 Publicering av hörn

För att koden som hanterar projection mapping skall ha data att jobba med, publiceras de fyra framåtriktade hörnen på en detekterad låda. Publiceringen sköts av metoden, *fetchCorners()*, som utifrån markörens position placerar hörnens position i rummet så att huvudprogrammet sedan kan publicera detta till en topic, */corners*.



4.6 Kommunikation mellan ROS och projection mapping

Kommunikationen mellan ROS och openFrameworks är nödvändigt för att BelieveMe skall kunna fungera. Då openFrameworks är ett fristående program utan koppling till ROS, krävs ett gränssnitt för att ha möjlighet att utväxla information mellan dessa. Tekniken som använts för att möjliggöra detta är att använda sig av en websocket, se 4.6.1.

4.6.1 Websocket

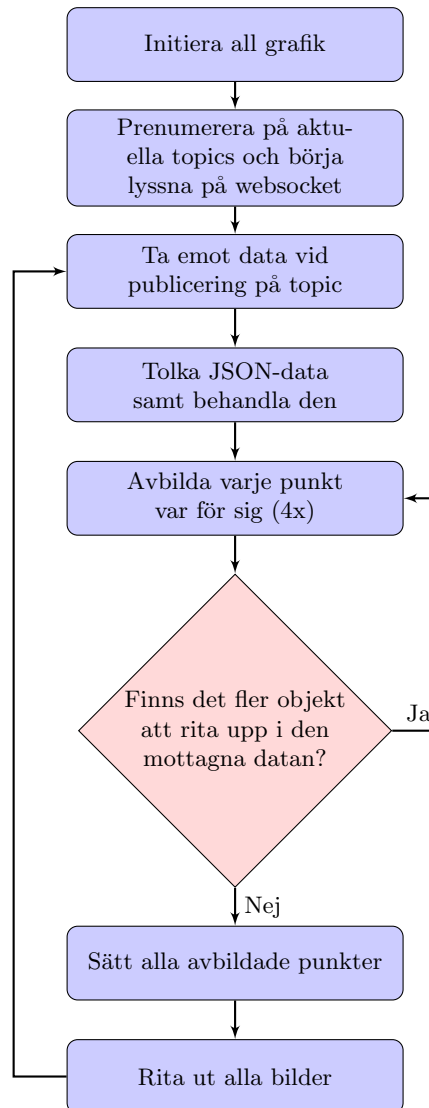
Websocket är en teknik som möjliggör tvåvägskommunikation över TCP-protokollet [48] som är pålitligt, det vill säga att datan kommer fram oförändrad. Enligt websockets protokoll, som definieras av IETF i RFC 6455 [49], består datautbytet av två delar; handskakning samt dataöverföring. Tekniken kringgår de problem som skulle uppstå vid liknande protokoll på grund av sin korta overhead samt tvåvägskommunikation över en och samma anslutning[49] vilket gör den snabb och lämplig för detta ändamål.

4.6.2 Rosbridge suite

Rosbridge suite är ett tillägg i ROS som förser övriga program med just det gränssnitt, som gör det möjligt att kommunicera med andra program, genom att starta en Websocket-server, se 4.6.1. Tillägget fungerar så att den förser klienten med möjligheten att lägga till en prenumeration på ett topic genom att skicka motsvarande kommando samt namnet på önskat topic till servern. Klienten i detta fall är openFrameworks med hjälp av tillägget ofxLibwebsockets [50]. Detta gör att varje meddelande som publiceras i fortsättningen på önskat topic direkt kommer att vidarebefordras till klienten i JSON-format.

4.7 Projection mapping

Här presenteras en kort beskrivning över hur det egenutvecklade programmet *3dTo2dMappingOF*, som hanterar projection mapping, fungerar. Det lyssnar på indata i JSON-format, där information ges om vad som skall ritas ut samt vart det skall ritas ut. Varje bild som skall ritas ut består av fyra koordinater som representerar hörnen på bilden, samt ytterligare information som bestämmer typ av bild. Programmet följer flödesschemat nedan.



5 Utvärdering av BelieveMe

I detta kapitel framförs resultatet av den färdigställda prototypen BelieveMe, samt en utvärdering av hur BelieveMe presterar vid interaktion med människor.

5.1 Verifiering av krav på BelieveMe

Vid bestämning av huruvida BelieveMe nått upp till de uppställda kraven har specificerade tester av kraven definierats för att sedan avgöra vad som klassificeras som avklarat respektive ej avklarat. BelieveMe skall uppfylla de punkter uppställda i 2.9. Krav och Specifikationer, där tester av kraven står utskrivna under listan av krav. Punkt 6 (Symboler och färger skall kunna tolkas av utomstående utan svårighet) och 7 (Utomstående person skall kunna förstå systemets intention utan insikt i programmet) i listan behövde prövas av personer utanför kandidatgruppen.

BelieveMe projicerar ut cirklar runt alla upptäckta människor samt projicerar ut röda ramar på lådorna, vilket visar att den uppfattar respektive människas och lådans position. Prototypen anses därmed ha uppfyllt punkt 1-3 i avsnitt 2.9.

Alla medlemmar kan flytta på vagnen utan problem, bortsett från att projektor och dylikt är uppkopplat med en sladd mot ett eluttag vilket hindrar vagnens rörlighet med avseende på dess räckvidd. Trots den begränsade rörligheten i räckvidd anses även punkt 4 som godkänd.

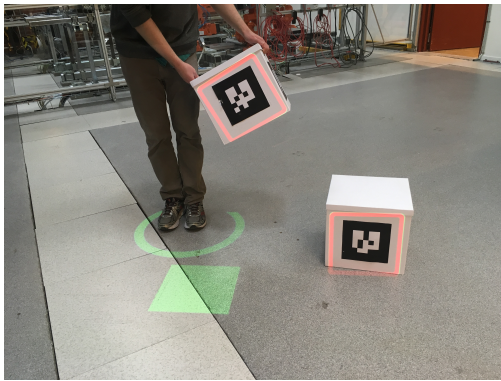
Navigationen har ej lyckats implementeras i AssembleMe, därmed har punkt 5 ej blivit uppfyllt, och AssembleMe går endast att genomföra i de fall BelieveMe ej förflyttas från sin ursprungliga position. Då det antydde i den första undersökningen att personer i närheten av robotar helst ser dess ruttplanering för att känna sig säkra är förverklingen av detta krav viktigt att fortsätta sträva mot.

5.2 Förutsättningar inför AssembleMe

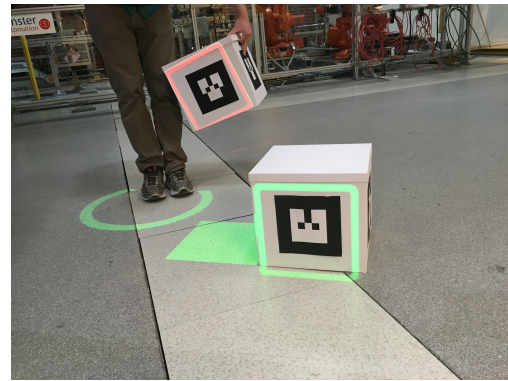
För att kunna göra en så oberoende undersökning som möjligt hade ingen av de som fick pröva på AssembleMe några förkunskaper om hur systemet fungerade eller hur AssembleMe gick till. De fick inte heller se varandra utföra uppgiften AssembleMe gav dem. De flesta av de som testade AssembleMe hade en teknisk bakgrund, vilket kan ha påverkat inställningen mot att bli instruerad av en robot. Det var även få testdeltagare och en liten variation i ålder.

5.3 Test av AssembleMe

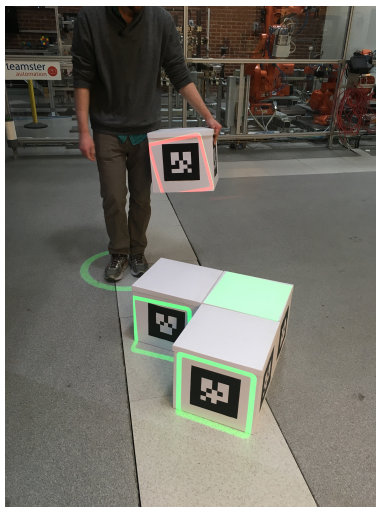
Vid provkörning av AssembleMe upptäcktes att systemet var känsligt för exakt placering och att projection mappingen inte alltid projicerade ut precis där systemet ville ha lådan då det fanns behov av noggrannare kalibrering. I övrigt fungerade systemet som förväntat. Bilder från det första testet av AssembleMe demonstreras i Figur 16.



(a) AssembleMe har startat och ingen låda är ännu placerad på den korrekta positionen



(b) En låda utplacerad, BelieveMe instruerar till nästa lådas placering



(c) Tre lådor utplacerade, BelieveMe instruerar till nästa lådas position ovanpå en tidigare låda

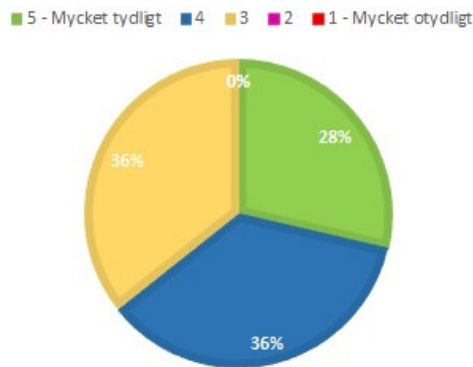


(d) Klosstornet färdigbyggt, AssembleMe är därmed slutförd

Figur 16: Genomförandet av AssembleMe.

Information samlades in från personer som prövade AssembleMe med hjälp av en undersökning. I undersökningen ställdes frågor om hur bra de uppfattade vad BelieveMe försökte förmedla. Frågorna i undersökningen samt de svar som samlades in finns presenterade i appendix B.

Undersökningen av AssembleMe, där totalt sett 14 personer deltog, gav en uppfattning om hur väl krav 6 och 7 uppfylls. Från undersökningen framgick att deltagarna överlag förstod BelieveMes intentioner väl, se Figur 17, samt att absoluta majoriteten, 92,9 %, tyckte färgvalet var passande. Därav kan också dessa mål anses uppfyllda.



Figur 17: Cirkeldiagram av svaren på frågan "På en skala 1-5, hur tydlig fann du kommunikationen från prototypen?".

Från resultaten från AssembleMe kan man se att majoriteten av deltagarna ansåg att kommunikationen fungerade väl och att färgvalet passade bra in i sammanhanget. Dock var det en viss andel som hade problem med att helt förstå instruktionerna. För ett praktiskt tillämpat system är inte endast majoritet tillräckligt för att godkännas, samtliga personer bör snabbt kunna förstå vad till exempel ett autonomt fordon vill förmedla för att säkert kunna förhålla sig till fordonet, och fokus bör därför läggas för extra tydlighet.

Trots att cirkelar projicerades ut runt fötterna på upptäckta människor, var cirka hälften av de tillfrågade osäkra eller ovetandes om huruvida BelieveMe hade sett personen i fråga. Detta kan eventuellt förklaras med att deltagarna var fokuserade på lådorna och därmed inte kollade nedåt mot cirkeln runt deras fötter. Ytterligare en anledning kan vara att datorn inte klarade av att leverera mer än 3-5 FPS när samtliga program kördes samtidigt, samt att personerna inte alltid identifierades när de stod på huk för att lyfta lådorna. Detta kunde stundvis göra markören eftersläpande. Att veta att roboten ser människor i dess närhet är en väsentlig del av att känna trygghet och tillit till robotar, och utvecklingen av bättre visualiseringsätt är välbehövligt för verkliga system.

Utöver resultaten från undersökningen fick projektgruppen dessutom kommentarer från deltagarna i undersökningen som är värda att nämna. Bland annat förstod inte vissa initialt vad de skulle göra. Först efter att de hade placerat ut första lådan på korrekt plats förstod de vad AssembleMe gick ut på och hur de skulle fortsätta. Därav kan det vara bra tydliggöra detta ännu mer, till exempel genom att skriva ut text i lådorna eller att innan start projicera ut instruktioner. Dessutom påpekades att färgblinda skulle tänkas ha stor nackdel i förmågan att tolka BelieveMes instruktioner. För ett system som skall kunna tillämpas i verkligheten för alla människor är detta ett problem som behöver adresseras.

Flertalet personer flyttade dessutom på första lådan när väl den placerats korrekt och nästa position visades, istället för att ta en ny låda. Detta tyder på att det inte var helt självklart att AssembleMe handlade om att bygga ett klosstorn. Detta kan alltså vara värt att tydliggöra.

Då projektorn fungerar optimalt i mörkare omgivningar, samtidigt som Kinect-kameran kräver ljus för att kunna urskilja RGBD-punktmoln, behövde oftast en avvägning göras vad gäller ljusförhållandena. Detta resulterade antingen i suboptimal visualisering eller bristfällig indata från kameran med risk för utebliven detektering. Då BelieveMe testades inomhus när dessa problem visade sig, kommer aldrig AssembleMe kunna implementeras utomhus där ljusförhållandena är ännu mer okontrollerbara, speciellt med avseende på solens IR-strålning

som markant påverkar Kinect-kamerans funktion.

Trots detta anser projektgruppen att BelieveMe uppfyller projektets samtliga mål, förutom målet om att BelieveMe skall gå att flytta runt, då navigeringen inte lyckades implementeras i AssembleMe. Det tål dock att poängteras att det finns mycket mer man kan göra för att fortsätta utveckla projektet, och fortsatt arbete rekommenderas starkt då utvecklingen av teknologi med kraftigt behov av denna typ av kommunikation är stundande inom en snar framtid.

6 Slutsats

Då samtliga undersökningsdeltagare ansåg att det kändes bra eller naturligt att bli instruerade av BelieveMe, dras slutsatsen att prototypen uppfyllt sitt syfte att undersöka hur tillit till robotar kan skapas. Resultatet påvisar även att konceptet bakom BelieveMe är ett möjligt tillvägagångssätt för att på ett tydligt sätt förmedla intentioner, men att tydligheten i visualiseringen var delvis otillräcklig.

Överlag är gruppen nöjd med projektets resultat och prototypens funktion. Det vore dock önskvärt att ytterligare utföra tester med prototypen för att på en större skala undersöka hur interaktion med och tillit till autonoma fordon kan skapas. Mest tid har lagts på att konstruera den faktiska prototypen och förhoppningar om att genomföra ytterligare undersökningar av hur dess intentioner kan förmedlas har ej blivit realiserade.

Användandet av en projektor tillsammans med en Kinect-kamera som informationsförmedlare har gett blandade resultat. Projektorn fungerar, på grund av sina begränsningar i ljusstyrka, bäst i ett mörkt rum. Kinect-kameran fungerar däremot optimalt i ett väl upplyst sådant, vilket gör att en avvägning har behövt göras. Valet av projektor i kombination med Kinect är därför inte att föredra för praktiska tillämpningar.

7 Förslag för fortsatt arbete

I nuläget anses en låda som blivit korrekt placerad förbli korrekt, och nästa byggposition kvarstår även om föregående låda flyttas. Som fortsatt arbete kan ytterligare kontroll av positionerna implementeras, för att säkerställa att lådorna ligger kvar i rätt position, annars projicera ut att lådan skall läggas tillbaka på positionen.

Då BelieveMes funktionalitet nedsätts kraftigt vid förflyttning under AssembleMes gång rekommenderas komplett implementering av navigationen i AssembleMe, för att kunna förflytta BelieveMe under körning utan att påverka dess funktion.

Huruvida individer kände sig upptäckta av BelieveMe eller inte var stundtals otydligt; att utveckla nya visualiseringsätt för att på bättre sätt fånga människors uppmärksamhet är därmed ett utmärkt förslag till fortsatt arbete.

Färgvalet, om än bra för en genomsnittlig befolkning, kan bli problematisk vid till exempel färgblindhet. För ett system som skall kunna tillämpas i verkligheten för alla människor är detta ett problem som behöver adresseras.

Eftersom flertalet personer istället för att hämta en nya låda, flyttade på första lådan när väl den placerats korrekt och nästa position visades, kan det vara värt att på något sätt tydliggöra instruktionerna.

Navigationen hann aldrig implementeras i AssembleMe. Detta på grund av att ruttens koordinater angavs utefter kartans globala koordinater och ej utifrån Kinect-kameran som vilket först antogs och var önskvärt. När dessa koordinater sedan försökte transformeras gav koden kompilersfel, på grund av att kompilatorn saknade länkar till direktivet, som förblev olösta på grund av tidsbrist. Som fortsatt arbete föreslås därmed att dessa länkfel löses och att ruttplaneringen implementeras i AssembleMe.

Då Kinect-kameran i kombination med projektor påvisade okompatibla tendenser kan valet av alternativa anordningar för detta syfte därav vara ett lämpligt undersökningsområde vid fortsatt arbete.

Överlag är det för fortsatt arbete viktigt att utföra ytterligare efterforskningar på vad som ska projiceras ut. När prototypen nu finns, ger det en möjlighet att testa flertalet olika visuella lösningar och närmare undersöka styrkor och svagheter med visuell kommunikation.

Referenser

- [1] S. Vasileiadou och D. Kalligeropoulos, "Myth, theory and technology of automatic control in ancient greece", s. 249–256, juni 2007.
- [2] J. Markoff, "Google cars drive themselves, in traffic", *The New York Times*, vol. 10, s. 9, A1 2010.
- [3] I. Asimov, *Runaround*. Street & Smith, 1942.
- [4] L. Takayama, W. Ju och C. Nass, "Beyond dirty, dangerous and dull: What everyday people think robots should do", i *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, ser. HRI '08, Amsterdam, The Netherlands: ACM, 2008, s. 25–32, ISBN: 978-1-60558-017-3. DOI: 10.1145/1349822.1349827. URL: <http://doi.acm.org/10.1145/1349822.1349827>.
- [5] F. Kaplan, "Who is afraid of the humanoid? investigating cultural differences in the acceptance of robots", *International journal of humanoid robotics*, vol. 1, nr 03, s. 465–480, 2004.
- [6] E. Coelingh, A. Eidehall och M. Bengtsson, "Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking", s. 155–160, aug. 2010, ISSN: 2153-0009. DOI: 10.1109/ITSC.2010.5625077.
- [7] D. Bourne, "My boss the robot", *Scientific American*, vol. 308, nr 5, s. 38–41, 2013.
- [8] C. Breazeal och B. Scassellati, "A context-dependent attention system for a social robot", *Rn*, vol. 255, s. 1, 1999.
- [9] C. Breazeal, A. Edsinger, P. Fitzpatrick och B. Scassellati, "Active vision for sociable robots", *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 31, nr 5, s. 443–453, 2001.
- [10] C. Breazeal, "Social interactions in hri: The robot view", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, nr 2, s. 181–186, 2004.
- [11] R. T. Chadalavada, H. Andreasson, R. Krug och A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space", i *European Conference on Mobile Robots (ECMR)*, 2015.
- [12] M. D. Coovert, T. Lee, I. Shinde och Y. Sun, "Spatial augmented reality as a method for a mobile robot to communicate intended movement", *Computers in Human Behavior*, vol. 34, s. 241–248, 2014.
- [13] J. Park och G. J. Kim, "Robots with projectors: An alternative to anthropomorphic hri", i *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, ACM, 2009, s. 221–222.
- [14] *Detecting people on a ground plane with rgb-d data*, http://pointclouds.org/documentation/tutorials/ground_based_rgb_d_people_detection.php#ground-based-rgb-d-people-detection. (hämtad 2016-04-28).
- [15] Z. Zhang, "Microsoft kinect sensor and its effect", *MultiMedia, IEEE*, vol. 19, nr 2, s. 4–10, 2012.
- [16] *Ros structure*, <http://www.generationrobots.com/en/content/55-ros-robot-operating-system>. (hämtad 2016-03-15).
- [17] C. Done, K. Conley och C. Müller, *Freenect 1.2.1*, <https://github.com/chrisdone/freenect>, sept. 2015. (hämtad 2016-05-02).

- [18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler och A. Y. Ng, "Ros: An open-source robot operating system", i *ICRA workshop on open source software*, vol. 3, 2009, s. 5.
- [19] F. Rigual Aparici, "Object recognition applied to mobile robotics", 2012.
- [20] J. M. O'Kane, *A Gentle Introduction to ROS*. nov. 2014, ISBN: 978-14-92143-23-9.
- [21] *Eh-tw6600*, <https://www.epson.se/se/sv/viewcon/corporatesite/products/mainunits/overview/13979>. (hämtad 2016-04-14).
- [22] B. Ekim, "A video projection mapping conceptual design and application: Yekpare", 2011.
- [23] *Vivid sydney*, <http://www.vividsydney.com/>. (hämtad 2016-04-13).
- [24] *Disney projection mapping*, <https://disneyworld.disney.go.com/entertainment/magic-kingdom/celebrate-the-magic/>. (hämtad 2016-04-13).
- [25] T. Asai, A. Minazuki, H. Hayashi och A. Kanazawa, "Construction of a mr space using projection mapping to promote hand hygiene", s. 247–252, sept. 2014. DOI: 10.1109/IIAI-AAI.2014.59.
- [26] M. Khosravi och Y. Lim, "Projection mapping for tourism in the city of esfahan", s. 335–339, sept. 2013. DOI: 10.1109/ICICM.2013.72.
- [27] D. Perevalov, *Mastering openFrameworks: Creative Coding Demystified*. Packt Publishing Ltd, 2013.
- [28] proformance - Marius Grimstad, *Ofxro-mapping*, <https://github.com/proformance/ofxR0-mapping>, maj 2016. (hämtad 2016-05-17).
- [29] —, *3dto2dmappingof*, <https://github.com/proformance/3dTo2dMappingOF>, maj 2016. (hämtad 2016-05-17).
- [30] J. Mace, *Rosbridge-suite*, https://github.com/RobotWebTools/rosbridge_suite, april 2016. (hämtad 2016-04-28).
- [31] P. Monk och L. J. Munro, *Maths for chemistry: A chemist's toolkit of calculations*. Oxford University Press, 2010, s. 429.
- [32] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita och N. Hagita, "Communicating robotic navigational intentions", i *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, s. 5763–5769.
- [33] Transportstyrelsen, *Varningsmärken*, <https://www.transportstyrelsen.se/sv/vagtrafik/Vagmarken/Varningsmarken/>. (hämtad 2016-03-11).
- [34] Transportstyrelsen, *Väjningspliktmärken*, <https://www.transportstyrelsen.se/sv/vagtrafik/Vagmarken/Vajningspliktsmarken/>. (hämtad 2016-03-11).
- [35] S. J. Mcdougall, M. B. Curry och O. de Bruijn, "Measuring symbol and icon characteristics: Norms for concreteness, complexity, meaningfulness, familiarity, and semantic distance for 239 symbols", *Behavior Research Methods, Instruments, & Computers*, vol. 31, nr 3, s. 487–519, 1999.
- [36] K. Khoshelham och S. O. Elberink, *Accuracy and resolution of kinect depth data for indoor mapping applications*, febr. 2012.
- [37] Z. Zhang, "Microsoft kinect sensor and its effect", *MultiMedia, IEEE*, vol. 19, nr 2, febr. 2012, ISSN: 1070-986X. DOI: 10.1109/MMUL.2012.24.
- [38] B. Chretien, *Rviz*, <http://wiki.ros.org/rviz>, juni 2014. (hämtad 2016-03-09).

- [39] *C++*, <http://www.cplusplus.com/>. (hämtad 2016-05-05).
- [40] *Github-kod*, <https://github.com/trykks>, mars 2016.
- [41] *Rtabmap_ros*, http://wiki.ros.org/rtabmap_ros, jan. 2016. (hämtad 2016-04-01).
- [42] *Geometry_msgs/posestamped message*, http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html, febr. 2016. (hämtad 2016-04-02).
- [43] *Nav_msgs/path message*, http://docs.ros.org/api/nav_msgs/html/msg/Path.html, febr. 2016. (hämtad 2016-04-20).
- [44] *Turtlebot*, <http://wiki.ros.org/Robots/TurtleBot>, mars 2015. (hämtad 2016-04-14).
- [45] F. Macintosh, *Tag_listener*, https://github.com/trykks/tag_listener, 2016. (hämtad 2016-05-17).
- [46] *Eigen*, http://eigen.tuxfamily.org/index.php?title=Main_Page. (hämtad 2016-04-16).
- [47] S. Niekum, *Ar_track_alvar*, http://wiki.ros.org/ar_track_alvar. (hämtad 2016-03-26).
- [48] J. Postel, "Transmission control protocol", 1981.
- [49] I. Fette och A. Melnikov, "The websocket protocol", 2011.
- [50] robotconscience - Brett Renfer, *Ofxlibwebsockets*, <https://github.com/robotconscience/ofxLibwebsockets>, april 2016. (hämtad 2016-05-17).
- [51] R. Adams och C. Essex, *Calculus: A Complete Course, Seventh Edition (7th Edition)*. Pearson Education Canada, 2009, ISBN: 0321549287.
- [52] M. Koecher och R. Remmert, "Numbers", i. New York, NY: Springer New York, 1991, kap. Hamilton's Quaternions, ISBN: 978-1-4612-1005-4. DOI: 10.1007/978-1-4612-1005-4_10.

A

Undersökning av hur färger, former och autonoma fordon uppfattas

1. Litar du på att ett självkörande fordon inte kommer att köra på dig? (156/156)
 - (a) Ja - 51,28 %
 - (b) Nej - 24,36 %
 - (c) Osäker/ Vet ej - 24,36 %
2. Skulle du känna dig trygg runt ett självkörande fordon? (156/156)
 - (a) Ja - 58,97 %
 - (b) Nej - 20,51 %
 - (c) Osäker/ vet ej - 20,51 %
3. Skulle du kunna tänka dig att äga ett självkörande fordon? (156/156)
 - (a) Ja - 68,59 %
 - (b) Nej - 17,95 %
 - (c) Osäker/ Vet ej - 13,46 %
4. Vilket av följande alternativ anser du vara bästa sättet att få dig att känna dig trygg runt fordonet? (155/156)
 - (a) Inget speciellt, jag litar på tekniken - 15,48 %
 - (b) Att fordonet visar att den har sett dig genom att lysa på marken framför dig - 25,16 %
 - (c) Att fordonet visar vilken väg den tänker åka genom att lysa upp sin planerade rutt - 46,45 %
 - (d) Jag kommer aldrig känna mig trygg, tekniken fungerar inte alltid - 12,90 %
5. Om ett fordon kommer mot dig och lyser upp rött på marken framför dig. Vad skulle du då tro? (155/156)
 - (a) Att du håller på att bli påkörd och bör flytta på dig - 67,74 %
 - (b) Att fordonet har sett dig och kommer att lämna företräde - 6,45 %
 - (c) Att fordonet har sett dig och vill att du står kvar där du är - 18,06 %
 - (d) Vet ej - 7,74 %
6. Om ett fordon kommer mot dig och lyser upp grönt på marken framför dig. Vad skulle du då tro? (155/156)
 - (a) Att du håller på att bli påkörd och bör flytta på dig - 6,45 %
 - (b) Att fordonet har sett dig och kommer att lämna företräde - 83,23 %
 - (c) Att fordonet har sett dig och vill att du står kvar där du är - 4,52 %
 - (d) Vet ej - 5,81 %

7. Om ett fordon kommer mot dig och lyser upp blått på marken framför dig. Vad skulle du då tro? (154/156)
- (a) Att du håller på att bli påkörd och bör flytta på dig - 6,49 %
 - (b) Att fordonet har sett dig och kommer att lämna företräde - 11,69 %
 - (c) Att fordonet har sett dig och vill att du står kvar där du är - 29,22 %
 - (d) Vet ej - 52,60 %
8. Om ett fordon kommer mot dig och lyser upp vitt på marken framför dig. Vad skulle du då tro? (155/156)
- (a) Att du håller på att bli påkörd och bör flytta på dig - 10,32 %
 - (b) Att fordonet har sett dig och kommer att lämna företräde - 11,61 %
 - (c) Att fordonet har sett dig och vill att du står kvar där du är - 12,26 %
 - (d) Vet ej - 65,81 %
9. Vilken av följande färger anser du skulle passa bäst för att visa att ett fordon ser dig? (153/156)
- (a) Vit - 11,11 %
 - (b) Grön - 54,90 %
 - (c) Röd - 7,19 %
 - (d) Blå - 17,65 %
 - (e) Gul - 9,15 %

B

Undersökning av AssembleMe

1. Har du en teknisk bakgrund, dvs studerat på teknisk högskola eller jobbat inom teknikbranschen? (14/14)
 - (a) Ja - 85,7 %
 - (b) Nej - 14,3 %
2. På en skala 1-5, hur tydlig fann du kommunikationen från prototypen? (14/14)
 - (a) 5 - Mycket tydlig - 28,6 %
 - (b) 4 - 35,7 %
 - (c) 3 - 35,7 %
 - (d) 2 - 0 %
 - (e) 1 - Mycket otydlig - 0 %
3. Hur snabbt uppfattade du vad prototypen ville förmedla? (14/14)
 - (a) Direkt - 14,3 %
 - (b) Snabbt, några sekunder - 42,9 %
 - (c) Upp emot 20-60 sekunder - 28,6 %
 - (d) Mer än en minut - 14,3 %
 - (e) Jag förstod aldrig - 0 %
4. Kändes det som att prototypen såg dig och visste var du var? (14/14)
 - (a) Ja - 53,8 %
 - (b) Nej - 0 %
 - (c) Osäker/ Vet ej - 46,2 %
5. Hur kändes det att bli instruerad av prototypen? (14/14)
 - (a) Bra och tryggt - 28,6 %
 - (b) Ganska naturligt, tänkte inte så mycket på det - 71,4 %
 - (c) Onaturligt och fel - 0 %
6. Tyckte du att färgerna prototypen projicerade passade bra in i sammanhanget? (14/14)
 - (a) Ja - 92,9 %
 - (b) Nej - 0 %
 - (c) Osäker/ Vet ej - 7,1 %

C

Poängsättning vid test av projektor

Tabell 2: *Grön projektion*

	14lux	112lux	394lux	Tot.
Röd	4	2	1	7
Grön	5	5	4	14
Blå	4	2	2	8
Vit	5	5	4	14
Svart	3	1	1	5
Gul	5	5	3	13

Tabell 3: *Röd projektion*

	14lux	112lux	394lux	Tot.
Röd	5	5	3	13
Grön	5	3	2	10
Blå	4	1	1	5
Vit	5	4	3	12
Svart	5	1	1	7
Gul	5	4	2	11

Tabell 4: *Blå projektion*

	14lux	112lux	394lux	Tot.
Röd	3	1	1	5
Grön	4	1	1	6
Blå	5	3	1	9
Vit	5	4	2	11
Svart	2	1	1	4
Gul	3	2	1	6

Tabell 5: *Vit projektion*

	14lux	112lux	394lux	Tot.
Röd	5	4	2	11
Grön	5	5	3	13
Blå	5	4	2	11
Vit	5	5	5	15
Svart	5	1	1	7
Gul	5	5	3	13

Tabell 6: *Gul projektion*

	14lux	112lux	394lux	Tot.
Röd	5	5	3	13
Grön	5	5	4	14
Blå	4	3	3	11
Vit	5	5	5	15
Svart	4	2	1	7
Gul	5	5	5	15

D

Rotation av kvaternion

En kvaternion är ett sätt att beskriva rotation, som till skillnad från den euclidiska metoden att använda sig av en eller flera rotationsmatriser istället använder en fyrdimensionell enhetsvektor.

En kvaternion innehåller en realdel och tre imaginärdelar, där realdelen är en skalär på rotationen och imaginärdelen är den enhetsvektor vi vill rotera vår punkt kring. Antag en euclidisk, tredimensionell vektor, v .

$$\vec{v} = (v_x \mathbf{i}, v_y \mathbf{j}, v_z \mathbf{k}) = (1, 0, 0)$$

där \mathbf{i} , \mathbf{j} , \mathbf{k} representerar de tre kartesiska axlarna. Vi kan nu skapa en enhetsvektor u kring vilken vi vill rotera denna vektor v .

$$\vec{u} = (u_x \mathbf{i}, u_y \mathbf{j}, u_z \mathbf{k})$$

Genom att utveckla eulers formel [51] kan vi få fram även en fjärde dimension på \vec{u} och därmed en kvaternion, q .

$$q = \sin\left(\frac{\theta}{2}\right)(u_x \mathbf{i}, u_y \mathbf{j}, u_z \mathbf{k}) + \cos\left(\frac{\theta}{2}\right)$$

där θ är rotationen i radianer kring \vec{u} .

Kvaternionen för rotation av \vec{v} kring u_y med $\frac{\pi}{2}$ radianer återfås nu enligt

$$q = (0, \sin\left(\frac{\pi}{4}\right)\mathbf{j}, 0, \cos\left(\frac{\pi}{4}\right))$$

För att sedan rotera \vec{v} med ovan skapade kvaternion används

$$v_{rot} = qvq^*$$

med \vec{v}, q och q^* nu beskrivna som en kvaternion uppdelad i en realdel och en vektordel där realdelen för $\vec{v} = 0$. Med denna uppdelning kan vi enligt hamiltons produkt [52], härefter benämnd som $hamil(q_1 * q_2)$, utföra multiplikation av kvaternioner på följande vis.

$$(r_1, \vec{v}_1)(r_2, \vec{v}_2) = (r_1 r_2 - \vec{v}_1 \cdot \vec{v}_2, r_1 \vec{v}_2 + r_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (18)$$

Första steget i användandet av denna ekvation enligt $hamil(q * v)$ ger

$$qv_{hamil} =$$

$$\left(\cos\left(\frac{\pi}{4}\right) \cdot 0 - \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \cos\left(\frac{\pi}{4}\right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 0 \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Denna kvaternion delas nu upp som tidigare och ger nya värden på r_1 och v_1 . Med dessa nya värden gör vi nu beräkningen $hamil(qv_{hamil} * q^*)$

$$qvq^* =$$

$$\left(0 \cdot \cos\left(\frac{\pi}{4}\right) - \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix}, 0 \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix} + 0 \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} + \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \times \begin{pmatrix} 0 \\ \sin\left(\frac{\pi}{4}\right) \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

Genom att plocka ut den imaginära biten ur kvaternionen får vi nu \vec{v}_{rot} som är \vec{v} roterad 90° kring u_y

$$\vec{v}_{rot} = (0, 0, -1)$$