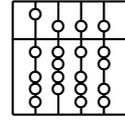


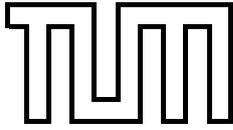
Technische Universität München
Fakultät für Informatik



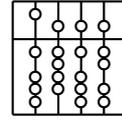
Master Thesis

Display Technologies for Augmented Reality support in Logistics

Troels Frimor



Technische Universität München
Fakultät für Informatik



Master Thesis

Display Technologies for Augmented Reality support in Logistics

Troels Frimor

Aufgabenstellerin: Prof. Gudrun Klinker, Ph.D.

Betreuer: Dipl.-Inf. Daniel Pustka
Dipl.-Inf. Björn Schwerdtfeger

Abgabedatum: 15. Juli 2006

Fixme:
Germen vs
english

Ich versichere, daß ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Juni 2006

Troels Frimor

Fixme: I need
to be
expanded
and in proper
german

Abstract

As the world becomes smaller the complexity of logistic networks increases. With it highly adaptive systems are needed to withstand large-scale rapid changes with influences in the logistics chain all the way down to the internals of warehouses. To this end, several techniques such as pick-to-light and pick-to-voice have been developed to assist workers in the common picking task both in efficiency and to decrease errors. Augmented Reality systems might have the potential to help the workers even more when adapting to new environments, but it has yet to be seen how the different display technologies help the workers perform. In this scope this thesis is based, and by exploring the characteristics of the displays, some of them have been chosen to have their usability in picking scenario evaluated by experiments. Therefore an experimentation framework based on DWARF was developed which allows to rapidly change between different displays while measuring user performance during different task from the picking scenario. And finally two experiments have been conducted to measure the wayfinding times related to navigation between shelves and to picking items from a shelf.

Preface

Purpose of this Document

Target Audience

Acknowledgments

Garching, June 2006

Troels Frimor (troels@frimor.dk)

Overview

1 Introduction	12
Introduction to this Thesis with description of the work flow, Augmented Reality and the <i>ForLog</i> project.	
2 Related Work	17
Current research evaluations of systems within warehouse logistics	
3 Picking Analysis	18
Analysis of the picking scenario with an exploration of how display technologies could play a role	
4 System Design	27
In this chapter the application design is detailed both in relation to Augmented Reality and DWARF but also as an experiment tool for testing display technologies	
5 Usability in Emperical Experiments	45
Description and introduction of concepts for how emperical measurements are used usability experiments	
6 Experiments for Wayfinding and Picking	50
Two experiments were conducted to measure usability of displays types in picking	
7 Conclusion	51
add something usefull here...	
A Experiment Questionnaires and Results	52
The questionnaires used for evaluating satisfaction among the test participants together with result from them and other measurements	
B Application Usage	66

Additional information about the usage of the provided applications such as startup commands

C Poses and Quaternions 70

Descriptions of how the positional information work in DWARF, especially how Quaternions can be used for rotation sequences

D Glossary 73

Abbreviations and Term Definitions

Contents

1	Introduction	12
1.1	Motivation	12
1.2	ForLog and MitLog	13
1.3	Augmented Reality	14
1.4	Structure of the Thesis	15
2	Related Work	17
2.1	Picking types	17
2.1.1	Pick-by-Light	17
2.1.2	Pick-by-Voice	17
2.1.3	Pick-by-Vision	17
2.1.4	RFID and Laser pointers	17
3	Picking Analysis	18
3.1	Base Scenario	18
3.1.1	Actors	18
3.1.2	Flow of events	19
3.2	Picking times	19
3.3	System Constraints	20
3.4	Displays	21
3.4.1	Current Systems	21
3.4.2	Spatial Displays	21
3.4.3	Hand-Held	24
3.4.4	Head-Mounted	25
3.4.5	3D Displays	26
3.5	Visualization	26
3.6	Sum up	26
4	System Design	27
4.1	General classes and structures	27
4.1.1	Orders	28
4.1.2	Items	28
4.1.3	Shelves and boxes	28
4.2	Services	29
4.2.1	WorkerService	29
4.2.2	WorkerInputService	33
4.2.3	Warehouse	33
4.2.4	PoseFactoryService	34

Contents

4.2.5	Display Services	36
4.2.6	LoggingService	39
4.3	Tool applications	40
4.3.1	Spatial relationships by picking points	40
4.3.2	SPAAM calibration	42
5	Usability in Emperical Experiments	45
5.1	Empericism for Human Subject Experiments	45
5.1.1	Factorial Analysis of Variance (ANOVA)	46
5.2	Conducting Usability Experiments	48
6	Experiments for Wayfinding and Picking	50
6.1	Experiment Setup	50
6.1.1	Wayfinding	50
6.1.2	Picking	50
6.2	Experiment Results	50
6.2.1	Wayfinding	50
6.2.2	Picking	50
6.3	Displays	50
6.4	Overview and sum up	50
7	Conclusion	51
7.1	Future Work	51
A	Experiment Questionnaires and Results	52
A.1	Questionnaires	52
A.1.1	Participant Information	53
A.1.2	Visualization Questionnaire	54
A.1.3	Display Questionnaire	55
A.2	Wayfinding Experiment Results	56
A.2.1	Questionnaire Results	56
A.2.2	Errors Measurements	56
A.2.3	Times Measurements	60
A.3	Picking Experiment Results	63
A.3.1	Questionnaire Results	63
A.3.2	Errors Measurements	63
A.3.3	Times Measurements	65
B	Application Usage	66
B.1	Services	66
B.1.1	Startup	66
B.2	Required third party software	66
B.3	XML validation	66
B.3.1	Schema structure for the warehouse XML files	66
B.3.2	Schema structure for the world locations XML files	67
C	Poses and Quaternions	70

Contents

D Glossary	73
Bibliography	75

List of Figures

1.1	Shared Environment Entertainment Pasture	14
1.2	Mixed reality continuum	15
3.1	Shelf system	18
3.2	Order with items	22
4.1	UML diagram of the ShelfSystem	28
4.2	Services used in the framework	29
4.3	UML class diagram of the worker service	30
4.4	Worker monitor GUI	30
4.5	Worker state diagram	32
4.6	Warehouse GUI	34
4.7	HMD visualizations for 3D	36
4.8	REFACTOR IMAGES...	38
4.9	UML diagram of the LogService	40
4.10	Example output from the LogService	41
4.11	Spatial Relationship GUI	42
4.12	SPAAM calibration GUI	43
C.1	Spatial relationships between tracked and non-tracked objects	71

1 Introduction

Introduction to this Thesis with description of the work flow, Augmented Reality and the *ForLog* project.

1.1 Motivation

The industrial sector of logistics influences all parts of the traditional industries in some manner and it is hardly possible to acquire consumer products which have not been through several chains of logistic systems. The topic covers everything from transporting mechanical parts, food or clothes from one end of the world to the other, as well as gathering of small parts for assembly lines. By doing so it creates increasingly complex networks of transportation and has turned into an fast expanding global US\$900 billion industry.[13, 3]

In these supply chains and networks intermediate storage is often not desired, but it is in many situations not completely avoidable and has inspired to much research in becoming as time and cost efficient as possible. Many different kinds of systems have sprung from this research, both automatic and human guided, and while the automatic systems are fast and reliable they can not be used in many situations because of a high degree of flexibility in the warehouse where the entire logistic structure is often changed or for items that are fragile or too small for the robotic arms to pick. Additionally the human picker can aid with quality control on the items making sure that no unusable items reach the final stages of a manufacturing process or the next level in the logistics chain.

Putting human workers in the middle of logistic chains in warehouses, creates the risk of bottlenecks in systems where the worker can not live up to the often hard time constraints. Errors may cause very high costs when for example wrong parts are manufactured into a preordered car or sent to the other side of the earth. For this reason the workers are under high pressure and many systems are designed to ease their work by making them more efficient and to create fewer errors. Augmented Reality ¹ systems might have the potential to ease the burden of the workers by creating systems after the pattern of placing the information the worker needs where and when it is needed.

In this scope this thesis is centered with the development a framework for, and performing preliminary evaluations of the the usability of display technologies and AR assistance for workers in the picking process.

¹Augmented Reality will throughout this Thesis be referenced by the common acronym AR, and is further defined in section 1.3

1.2 ForLog and MitLog

To advance logistic systems to be prepared for future challenges the *ForLog* project has been created² to develop and verify prototype modules within several the areas of logistics. The focus though lies in the logistic systems for the automobile industry. The main mantra within the project is to create supra-adaptive systems:

By Supra Adaptivity what is understood is the systems ability to adapt at minimum expenditure to dynamic enterprise-wide changes. This goal can be reached by a determined combination the ability to transform and networking together with mobility in the physical and especially in the informative and structural sense.³

To achieve this goal *ForLog* has been divided into six sub-projects each with focus on more specialised areas:

FlexLog Addresses the concept flexibility in all aspects from definition to why and when the need for flexibility arises.

SysLog Deals with the architecture of information systems within logistic systems and analyse what requirements are put on these systems to support supra-adaptivity.

PlanLog Has the goal of creating standardisations within the planning of logistic systems and for that identifies and classifies factors herein.

TransLog Creates new concept models for service providers in outsourced production.

NutzLog Identifies and classifies measurable values of different links in logistic supply chains.

MitLog Focuses on the humans in these dynamic changing environments and on how to make them capable to cope with the additional demands put on them. Additionally it deals with expert knowledge and how it has to show the same flexibility as the considered logistics systems.

The *MitLog* project is the frame for the work done with this Thesis and the experiments performed (see chapter 5) focuses on AR techniques used to keep the expert knowledge in the warehouse while the workers using the system might be changed often or otherwise rotated around between different work stations.

²ForLog is project with coorporation between several universities and companies. It is sponsored by the Bayerische Forschungsstiftung (BFS).[27]

³Translated by the autor. Original: "Unter Supra-Adaptivität wird die Fähigkeit eines Systems verstanden, sich mit minimalem Aufwand und zudem unternehmens übergreifend an Veränderungen anzupassen. Erreicht werden kann dies durch eine gezielte Kombination von Wandlungsfähigkeit, Vernetzungsfähigkeit und Mobilität im physischen wie vor allem auch im informatorischen und strukturellen Sinne."

1.3 Augmented Reality

AR is a research area which focuses on bringing information from the computer systems out in the real world where the users can see, hear or otherwise perceive it. Ideally this allows the users to perform their work and taking advantage of additional knowledge in the computer systems without having to use computer terminals or other access points. From a more technical point of view, performing augmentations requires the knowledge of the pose⁴ of both the real objects one wants to augment and most often also the users of the system. To this end, a tracking system is used together with a virtual model of the objects to be augmented so that the spatial relations between objects can be determined.

The traditional way of creating AR systems is by using either a video or optical see-through Head Mounted Display (HMD) with the virtual information superimposed as objects on the real. But many other display methods exist in combination with AR, which, among others, is demonstrated in the *Shared Environment Entertainment Pasture (SHEEP)* (see fig. 1.1).



Figure 1.1: The *Shared Environment Entertainment Pasture* is an experimental system to test interaction techniques with AR. It is based on a virtual landscape with sheeps projected onto a table and where the landscape also can be seen through laptop screens, PDAs and HMDs. Courtesy of the SHEEP team, <http://campar.in.tum.de/Chair/ProjectSheep>

While Virtual Reality (VR) is more commonly known than AR, they are two closely related techniques within computer science. But while they both focus on mixing real with virtual to some extent, VR brings the users into a completely computer created virtual environment while AR brings the virtual objects into the real environment for the users to sense. This mix leads to the *reality virtuality continuum* definition from Milgram (see fig. 1.2) where augmented reality is lying towards the end of real environments end, with more reality than virtuality.

⁴A pose is the combined information of the position and orientation of an object. See the appendix C for how pose information is represented and combined.

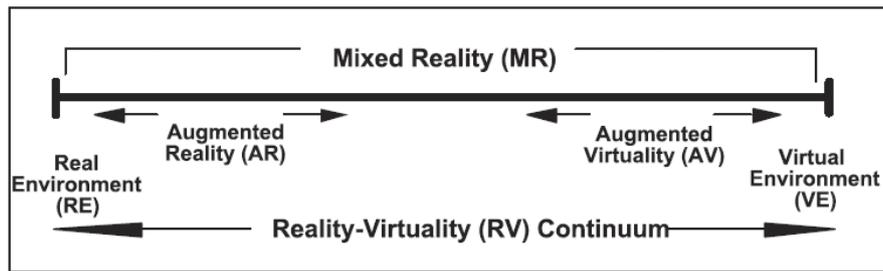


Figure 1.2: Mixed reality continuum. Courtesy of Milgram. [18]

To try and paint a picture of the how different techniques of AR influences the picking process, the performed experiments for this Thesis spans as much as possible on the AR section of the RV continuum of fig. 1.2. Note that this definition of AR is broader than the otherwise common definition from Azuma⁵ since the maps tested are not registered in 3D but only 2D information is used. Also systems which are not registered at all are tested to be compared with, and to create a more complete set of test from the AR spectrum of possibilities.

1.4 Structure of the Thesis

zeigen die Probleme auf, skizzieren die Lösung, die angestrebten Ziele und die erreichten Ergebnisse.

HAS BEEN CHANGED!!! Redo...

The entire work flow leading to this Thesis can be divided into four main parts which is also directly reflected in the central structure of this paper:

Analysis of Scenario(chapter 3) Is the analysis of the picking process together with what requirements it puts on both AR other systems using displays developed for it. It divides into physical constraints of the actual displays and then on the visualizations used. During the time to analyse the picking scenario, several real industrial picking working areas was visited to observe and better understand what difficulties lie therein.

Experimentation framework development (chapter 4) Describes the framework developed to be able to quickly perform experiments with a multitude of different configurations with shelves for picking and guidance subsystems. The development time spanned the entire time frame of the Thesis period since some parts could already be written early while others were dependent on the input from the user interface experts at the chair for AR at the technical university and input from psychology student from university Regensburg.⁶

⁵The very commonly used definition from Azuma contains three requirements for a AR system. It must: 1) combine real and virtual, 2) be interactive in real time, and 1) be registered in 3D.[2]

⁶The *MitLog* project is associated with the chair for Experimental and Applied Psychology at university Regensburg under the direction of Professor Dr. Alf Zimmer.

Experiments (chapter 5) Contains the description of the experiments performed which includes choices of what was tested, why it was tested and how it was set up. The results of these experiments are in the app. [A](#) together with the questionnaires.

Evaluation of Results(chapter 6) Is the last part of the central structure of this Thesis. It contains the results and evaluation of the measured values from the experiments.

Accompanied with the work for evaluating the displays technologies and analysing the picking scenario much related work has been encountered and is described next in the following chapter.

Rephrase!!!

2 Related Work

Current research evaluations of systems within warehouse logistics

Where to find info about this???

2.1 Picking types

2.1.1 Pick-by-Light

A very commonly used system for picking in warehouses is the Pick-by-Light [12]. As a minimum it consists of small lights placed on the shelves, one by each type of item to pick. When the worker receives a new order with items, it gets registered with a system controlling the small lights which then turns on the lights by all the items the worker should gather. Additionally the system often has correction and confirmation buttons by the lights so the worker can tell the system when an item has been picked and correct it if the item is put back.

2.1.2 Pick-by-Voice

By wearing headsets and using a speech recognition system, the workers can receive positional information for the items to pick directly while walking around the warehouse collecting the items and can still give direct feedback to the system of which items are picked or similar information. Systems like this are more commonly used in large warehouses where the span of items cover many large shelves but with small items so both hands are needed to pick them up or in cold warehouses where the worker should wear gloves.

One of the advantages of a system with voice picking is the possibility for non-linearity among the orders. Since the worker is not given any information before it is needed, all orders can be rearranged for how urgent they are before the worker receives the next one.

2.1.3 Pick-by-Vision

2.1.4 RFID and Laser pointers

3 Picking Analysis

Analysis of the picking scenario with an exploration of how display technologies could play a role

In this chapter the picking task will be further analysed by describing the picking scenario, finding requirements for system and display technologies in connection with this scenario, taking an in-depth view of how the different technologies could be used and finally an investigation of visualizations and their effect upon cognitive workload in navigation and picking tasks.

3.1 Base Scenario

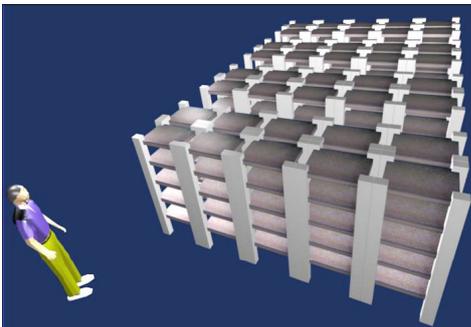


Figure 3.1: Shelf system

so that the worker does not have any problems picking the items out.

The picking scenario which will be used as reference in the following sections and in the later experiment involves one worker, a shelf system as showed on figure 3.1 and a system through which the worker receives orders. The shelves have several layers of boxes with items and consist of several rows where the worker can move freely in-between. There should be so many boxes that the worker cannot easily remember the location of each item, and therefore there is no relation between the item identification and its position. The items themselves are small and light while the boxes are big enough

A more abstract view of a mechanism for displaying information to the worker... The displays/paper/lights all fits into this pheripial.

3.1.1 Actors

The described scenario should be considered a base scenario where the worker can use different display systems to aid navigation and to be notified from the two other actors. Independent of the what displays are used, the base scenario always have the following actors and the structure as described in the following flow of events.

The worker Picks items according to the current order he received from the order system.

The shelf system Registers when an item has been picked.

The order system Receives orders from an external warehouse or logistics system.

3.1.2 Flow of events

The scenario then follows the following flow of events:

1. The worker enters the picking area and turns on the system. As he enters the he is already wearing the nessecary equipment for using the the order and shelf system, including any markers to be tracked and displays he should need.
2. The order system receives an order from the external logistics system. It notifies the worker that an order is ready to be processed.
3. The worker acknowledges that he is ready to process the order. This could be through traditional computer user interfaces or for example speech recognition.
4. The order system sends the order information to the worker, this can be a print out of the order information or something to be represented on a display by the worker.
5. The worker goes the next position displayed on the order
6. He finds the right shelf and picks the needed amount according to the order
7. If there are no more positions then he should go to the base with and put the items at the goal otherwise follow again from step 5 with the next item.
8. If there are more orders waiting for the worker to process then the order system should notify the worker and he should begin from step 2.

another term

3.2 Picking times

The overall picking time can be divided into four different phases[14] which describe the entire time needed for the worker to process an order with items. The times are defined in relation to the number of *positions* in the order, where a position is the location in front of the shelf where the worker has to stand to pick up one or more items of the same type. If more he can reach more types of items it then counts as more positions.

The times described next are the relevant ones to measure for trying to optimize the picking workflow. It is traditionally done via different strategies for which ways the worker walks through the shelf system when picking items.

Base time The time the worker is standing at the *basis* before and after the worker walks around picking items. This time is needed to receive an order, understand the information, finish the order by placing the picked items at the goal and informing the system that a new order can be started.

what is the goal for the items called?

Way time The time span between items the worker should pick, where the worker moves toward the next item. For an order the *way time* is measured as the average time per position.

Picking time Is the time where the worker stands in front of the shelf with the item(s) he needs to collect and moves his arms to pick up the right ones. This is also measured as an average per position in the order.

Dead time The time the worker needs when standing at a position but is not actually collecting items. It includes the time for reorientation and finding what way to take to the next position which is the defined as the *wayfinding* time and together with *traveling* time makes up the time for *navigation*[22]. As above it is also measured per position.

3.3 System Constraints

The constraints describe under what conditions and requirements a system for the picking process should be developed and includes what is necessary for guiding a worker in his job to pick the right items at the right time. Many more constraints exist for more specific picking scenarios but here only the constraints for the base scenario are described. Included are the additional requirements from the high degree of flexibility needed in the *ForLog* project.

Item information The single most important requirement is a mechanism to let the worker know where to go and pick the next item. This information can be given in many different ways and is what can be varied by using different visualizations on different display technologies. It could have a direct influence on efficiency and amount of errors both of which will be measured in the performed experiment described in chapter 5.

Quality Management In most picking processes in warehouses, errors can result in major costs for the company, for example for parts to be used in car manufacturing. For this reason the worker is often used as a form of filter to find and discard damaged items, but making this judgement normally takes much training.

Error Management There should be a system to make reports for unusual situations. This can happen if there are no more of a certain item left, if a hard deadline is broken or if all for some other reason a certain type of item can not be delivered.

Notifications The system should be able to notify the worker for different kinds of events. This ranges from events like when a new order has arrived for the worker to process, another co-worker arrives to help meet the deadlines or when the work for the day is done.

Registration The system should recognize when a certain item has been picked. This can happen manually by the worker pressing a button or automatically for example to register a weight change in the box with the given item. There should be a possibility for the worker to notify the system in case the automatic system fails.

Flexibility The *MitLog* project intends to make the picking process much more flexible in the sense that workers have to move around between different working spaces when needed. For this reason the following requirements can be drawn.

Flexibility ???

Scaleability It should be possible to dynamically extend the amount of workers using the system so any changing in deadlines and work load may be foreseen and handled.

Ease of Use New workers should be able to use the system without training and still meet the deadlines. Thus the system has to be very intuitive to use.

I wanted to write something but forgot what...

3.4 Displays

Traditionally the only information the worker needs is a list of items with their positional data. Using context aware systems and different display technologies does not change the need for knowing which items to find and where, but might help the worker in new ways to improve his efficiency and making fewer errors. Also for having highly flexible workers, as is one of the sub-goals in the *MitLog* project, if giving the worker access to additional information when he needs it could make it possible for both untrained workers and workers who are new to the current working area, to still perform the picking task in a satisfying manor. AR is one of the technologies which could make this new way of performing picking possible and in this section different kinds of displays from AR will be discussed. The term display will be understood in a broad sense to any means to display information to the worker.

The goal of this section is to analyse what benefits and drawbacks the different display technologies could bring when used in the scenario described in the previous sections. First each kind of display is analysed and in the end a summary and comparison is done where the displays will be divided into categories.

Have I done this?

COMPLETE REMAKE!!! Most of the following section is just a copy from an old paper

3.4.1 Current Systems

The systems currently used are typically based on printing out the orders to a list of items as shown on fig. 3.2.

Insert image from the spatial display book

3.4.2 Spatial Displays

3.4.2.1 Lights

Here the worker prints out an order like the one displayed on fig. 3.2, scans in the bar code which makes the system turn on small lights by all the items which should be collected. The



Figure 3.2: Order describing what items to pick and where to find them in a typical picking scenario. Courtesy of BMW Group.

worker then walks around and collects the items, attaches the printout to them and puts them onto a small wago.

Advantages:

- Easy to setup since no tracking or registration is needed and all electronics are easy to come by while the software would be quite direct to write (or buy).
- Changes are easy to create since only small software changes (maybe just a register) are necessary. The lights could also be easy to replace or move.
- Repairs would be easy to perform.

Disadvantages:

- The system is fixed on a certain amount of users. Additional users require more lights in other colors.
- To change or move the setup a complete dismantle and re-setup is necessary.
- Light conditions may blur the contrast needed to spot the lights.

3.4.2.2 Remote

With remote screens two solutions are possible. It can simulate the lights approach by displaying a light on the screens where the worker needs to pick up an item. A description like the one on fig. 3.2 could be displayed on either all or a few chosen screens. The 2D Arrows approach puts an arrow on each screen to guide the worker to the next item. This way a pre-calculated path is computed for the worker to follow to pick up the items. When one item is picked up, all the arrows will turn toward the next item. Maybe if more items are to

be collected all the arrows closest to an item always points toward that one so all items are pointed to at the same time.

Advantages:

- The description and lights on the screens case has just an easy setup as the lights case.
- Using the 2D arrows requires a simplified registration since the screens used for an item would be on the same shelf and thus in the same plane.
- The worker has both hands completely free.

Disadvantages:

- Many small screens are needed if one is to be placed at each possible item.
- Using that many screens and the very direct guidance might not be very comfortable for the worker.
- It is not possible to show with the arrows whether two of the same item is needed, thus the worker might just think that the system haven't understood that a certain item already has been collected.
- The screens have to be small to be on or by the smaller items. But is it possible to see an arrow on such a small screen from a distance?

The requirement of the many screens will properly make this approach too expensive both build and maintain. Furthermore to have a precise enough selection of items, more than one arrow might be needed close to the item to uniquely identify it. Having both hands free is a big advantage though in places where the items are either big or unhandy, and in this case fewer screens with arrows might be needed since the boxes containing the items will be bigger.

3.4.2.3 Projected

Attaching projectors to the ceiling or walls will allow augmentations to be displayed on the walls and shelves. With this the necessary information for guiding and informing the worker could be given with both the above mentioned methods but also with a new system of lines and labels. Here lines could guide the worker to the right items, whereas the labels could provide additional information about the item or the quantity of the given item.

Advantages:

- Only a few projectors are needed per shelf. (Compared to several screens)
- Quality control information can be displayed as a label by items when they are collected.
- No additional equipment is needed to be worn and both hands are free.
- No tracking of the worker is needed.

Disadvantages:

- Whether the contrast is good enough to display the quality information as a label on the shelf.
- Limited number of workers may work with the system since too many lines might be very distracting.
- Requires extensive registration if no geometric data (like projecting perpendicular onto the shelf.)
- May give inconvenient shadows if worker goes in front of the projector beam. Worker path could have to be considered when setting up projectors.

Although relative simple in equipment, using projectors gives big registration and shadow issues which must be resolved to assure the correct items are collected. An alternative to lines and labels could be the Spotlight method where the projectors just highlight the area around the items which should be collected. This method can even be calibrated by automation together with a small camera by the projector and visual markers by the items.

3.4.3 Hand-Held

3.4.3.1 pda... some other name

With a personal screen the worker gets a small screen (for example a PDA) to his wrist. This small screen could then either just be used as a small screen with necessary data or as an instrument to do AR in which case it has to be registered. As an AR instrument it can be used as a see-through display or as a device that can show relevant information from the knowledge of the position of the display and worker.

Advantages:

- All workers have their own input output system with them. No need for sharing a computer or problems with mixing augmentations.
- Errors and special cases are easy to communicate between user and system.
- Quality control is also done directly.
- Only as many devices as workers are needed.

Disadvantages:

- The screen might too heavy for longer usage especially in the see-through case where the arm should be lifted up to see through it.
- The device might interfere with the agility of the worker or prevent him completely from using one arm.
- When doing input both arms are needed.

- Tracking of not only the worker is required, but also the local screen needs to be tracked. This gives a lot of tracking issues like whether device is covered and so on, and a fusion of different tracking mechanisms must be necessary.

Although using a device attached to one arm and the tracking issues, a local screen is still interesting in several ways. Unlike the earlier systems it has a direct input/output system by the user and it is better scaleable to several workers. An alternative to using a screen attached to the workers arm is attaching it to a wagon, if such a wagon is used to collect the different items. With this solution the worker would not be tired of constantly lifting his arm to get information, but if using it to augment the shelves and boxes then the workers head (eyes) would also have to be tracked.

3.4.4 Head-Mounted

3.4.4.1 hmd ... some other name

A head mounted display can be used to simulate all of the above since any visual augmentation can be performed for the worker. There are many different kinds of HMDs but in this context only one with 3D view is considered.

Advantages:

- The advantages include that high precision augmentations can be placed wherever needed. Later experience for the system designers might optimize the user interface shown in the HMD to improve worker performance and system ergonomics.
- Both hands are free.
- Easy change between different kinds of augmentations might be possible, depending on given task or worker preferences.
- No additional movements are needed to get information when needed as long as they are shown in the HMD.
- Smaller HMD might be used if no registration and only information is needed.

Disadvantages:

- It is very difficult to know which augmentations are better with an HMD and many possibilities are possible.
- Additional input/output system is needed.
- Physical size and weight might be hard working conditions.
- HMDs are very expensive.
- The worker has to be registered and tracked with high precision.
- HMD has to be configured to individual worker.

HMDs are interesting to test different augmentation possibilities and have the potential to both increase worker efficiency and ease of understanding of new working environments. The disadvantages are big though, but depend much on the kind of HMD used.

3.4.4.2 Projected displays

3.4.5 3D Displays

Should this section be removed?

3.5 Visualization

May in fact increase the cognitive load on the user [22] (page 6 section 5)

for this reason visualizations play a large role in the evaluation of disp techs. . .

3.6 Sum up

COMPLETE REMAKE!!! This is just a copy from an old paper

Several of the different proposals are worth testing since there are many unanswered questions both in acceptability and efficiency. Using remote screens put many mechanical constraints (one needed per item, takes space next to the items) on the system and would also be quite expensive to build. Projectors on the other hand, are cheaper since not as many are needed, but they create many quality issues like with the contrast and shadows. Personal screens are interesting because they allow for the input to the system to be directly by the workers hands. The screens by the worker also allows for easy scaling with more workers, also while the system is running. HMDs have many advantages above the others as a device for showing augmentations, but as a device used when working 7-8 hours per day it is properly very encumbering.

For testing it would be interesting to make an efficiency test with at least the lights, projectors, personal screens and HMDs since they provide very different ways of showing augmentations for the worker, and have very different disadvantages which could be compared by a user survey. The augmentations used in the different cases would be Lines and Labels with the projector, either Compass or Mini Environment with the personal screen and 3D Arrow with the HMD.

4 System Design

In this chapter the application design is detailed both in relation to **Augmented Reality and DWARF** but also as an experiment tool for testing display technologies

To be able to quickly and easily do experiments on display technologies and doing measurements on their usage, an AR framework was created where the classes for controlling the display devices easily can be replaced. It is based on the DWARF system with services each having their own needs and abilities. In this framework one DWARF service for each display type should be created and where it then can draw on the information the framework services make available simple by providing a need for it.

dwarf and Asars dortorarbeit reference. notion in footnote.

In this chapter the framework will be described by first describing the basic structures like an order is represented and sent in the network of services. Afterwards the services will be described in detail with information on how they work and what they are responsible for. Finally some tools for setting up applications with this framework are described. The tools are more general to use than for this framework, but I developed them in conjunction with the services as they proved very handy to have when setting up new systems.

The entire system was developed in the *Java* programming language with *Java3D* for the visualizations but with the exception of the program for the PDA which was programmed via *Visual Studio* in *C++* with *OpenGL ES*¹ for visualizations.

For configuration of applications using the framework, the *MasterConfig.xml* file is used. It contains information like projections matrices, what visualizations are enabled and on which monitors they should be displayed on.

Further information about the framework such as startup information, can be seen in the appendix [B](#)

4.1 General classes and structures

A few classes are used by most of the services in the system and are for this reason described here as reference. More information about their usage is available in the sections of the services using them.

reformulate

¹*OpenGL ES* [28] is a fixed point specification based on a subset of the *OpenGL* graphics API specification. It is specifically targeted for embedded systems such as PDAs.

4.1.1 Orders

The `Order` class is a simple class holding information about a specific order for the worker. It simply consist of a list of `Items` associated with a number of how may to pick of the given item and how many already are picked. The orders are created in `Warehouse` service from where they are sent to the `WorkerService` when the worker should process them.

The `Order` hold the items it contains in a stricht order from when it is created in the `Warehouse`. This order is mentained when serializing and deserializing the `Order` via the static functions `String order2String(Order o)` and `Order string2Order(Stings)` to be sent over the network.

4.1.2 Items

The items simply consist of an ID and a description to reflect what real object it represents. They are only intended to be a part of the `Order` classes and and where their information can be retrieved for different visualizations.

The `Order` directs the resposibility of (de)serializing the item information to the item class.

4.1.3 Shelves and boxes

The `ShelveSystem` class contains several inner classes to represent a shelf system (see fig. 4.1), where most the central class is the `Box` class which is associated with one specific `Item` and contains the `pose` of this specific box in the tracking coordinate frame. These

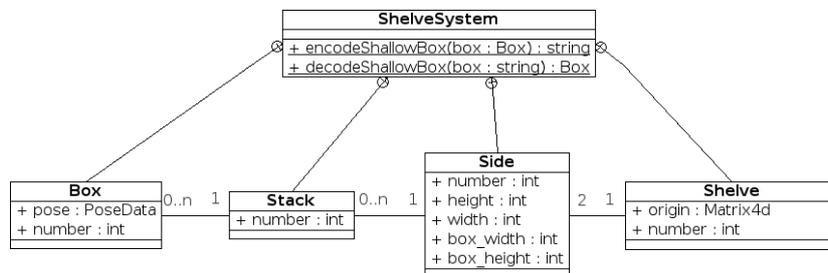


Figure 4.1: UML diagram of the `ShelveSystem`

classes makes it possible to represent two-sided shelves where the boxes on one side all have the same dimensions as specifies in the class representing the sides. A way to send the `Box` over the network is provided by the containing `ShelveSystem` class by serializing the information. The serialization is performed in a shallow form, meaning that only the actual box plus the containing stack, side and shelf objects are serialized but no siblings boxes are added in the serialization.

The shelf system is mainly used in the `PoseFactoryService` where the positional information is computed and used and used as described in section 4.2.4.

4.2 Services

As already mentioned the created framework was created via DWARF and as such is based on different services each with needs and abilities. The structure of these services can be seen on fig. 4.2 and each of the individual services will be described in the following sections. The division into services has all of the benefits from DWARF where the ability to quickly change and add new services properly is the most significant for testing new displays. By

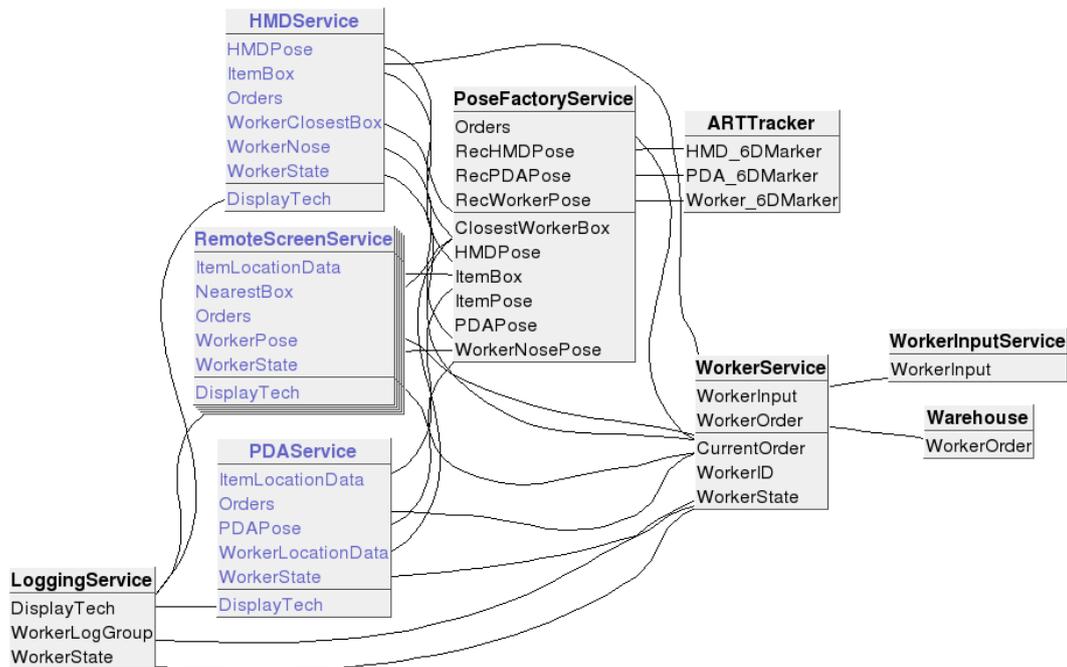


Figure 4.2: Services used in the framework. The blue color indicate that the service is not started and the black services are. In this state any of the display services can be started and the system will start sending the information it needs. Note that the ARTTracker service is a standard part of DWARF.

creating a new service for each different tested display technology the distributed abilities of DWARF can be taken into advantage. Furthermore as described next, the representation of the worker and computations of pose data in the tracking system are separated into reusable services for new tests.

4.2.1 WorkerService

The worker service is the central service representing the state of the worker in the system, ie. what the system thinks the worker currently is doing and what he should be doing next. It includes a information about which `Order` the worker currently is picking items for and which orders he should start processing afterwards. The main structure of this service is displayed as an UML diagram on fig. 4.3 where the *state machine* classes are packaged into `state`.

4 System Design

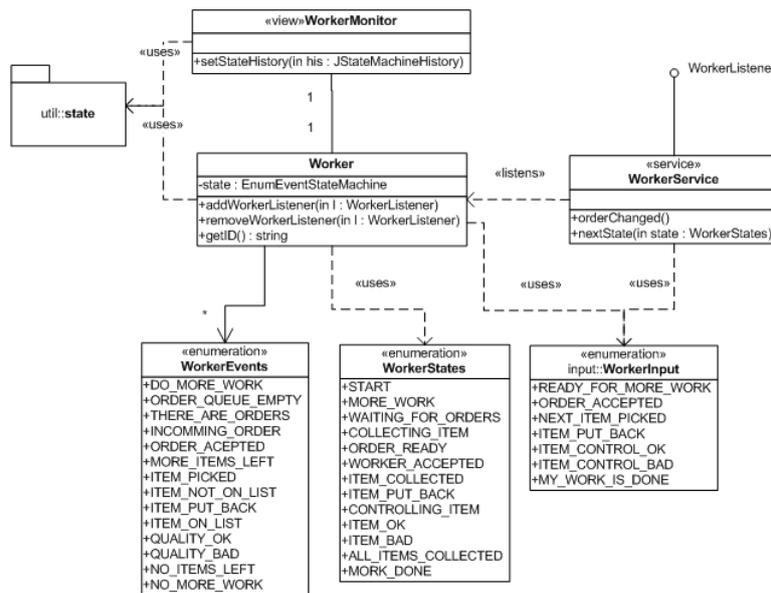


Figure 4.3: Simplified UML class diagram of the package `de.tum.in.FAR.picking.worker` showing the main associations in this package while leaving out how orders are represented and which classes are used for the GUI.

Instances of the input enumeration are received through the `WorkerService` as events from the worker input service (4.2.2).

These enumeration instances are translated in the `Worker` class to events in the state machine which then may then trigger a transition to a new state. If this happens the `Worker` is notified and checks whether the a new event should be sent to the state machine, for example in the case where the worker sends the event that the person is ready for work (`DO_MORE_WORK`), and there are already orders to process, then the `Worker` class automatically sends this event (`THERE_ARE_ORDERS`) as well into the state machine.

The worker monitor gives an overview of the state of the worker including ways of restarting the state machine and changing the ID of the worker. As described in the `LoggingService` the log files are automatically changed with the changes in the worker ID, thus one log file per worker experiment can be created. The bottom part of the GUI (fig.4.4) resembles the current state of the state machine plus a short history of transitions and possible transitions from the current state. By double clicking on a outgoing

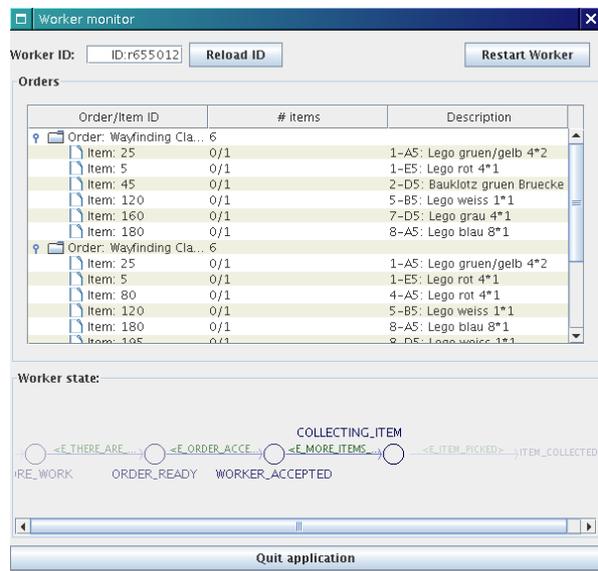


Figure 4.4: Worker monitor GUI

transition the state change is forced in the state machine whether the event for it has happened or not.

Make a formal description of the needs and abilities?

4.2.1.1 State Machine

Using a state machine to represent the worker makes it easier to analyse what is going on and to change or add new states to what the worker should be doing. For this reason the state machine in the package `de.tum.in.FAR.util.state` was created. The package provides very flexible state machines all created via the factory methods in the `StateMachineFactory` class. The one used for representing the worker is an enumeration based event driven state machine, where the states are automatically generated from the enumeration class `WorkerStates` and the events to trigger transitions are created from the enumeration `WorkerEvents`. In the `Worker` class these states are then connected with transitions to and finally represents the worker as the diagram shown on fig.4.5. In the next two sections the semantics of the states and events are then described.

The advantage of creating the state machine on enumerations are that the code becomes much more clear and extend if further states and or transitions and events are required for extended experiments.

4.2.1.2 States

The states are created to reflect the different states of the worker when performing the picking, but also with additional states controlled and checked by the system. These additional states makes it possible for the display controlling services to react and for the logging service to register the time. For example the state `ITEM_OK` which only signals that the worker considers the quality of the last picked item to be ok and is therefore ready to pick the next item. The system then checks whether there are any more items to pick in this order or whether all the items are collected. This allows the service to send the choice of whether the quality is ok as a single state change and it will thus be registered in the `LoggingService` with a timestamp for later analysis.

The meaning of the the different states can be characterized as follows:²

START Is the entry state for the state machine.

MORE_WORK Is the state that is entered when the worker has signaled that he is ready to start receiving new orders.

WAITING_FOR_ORDERS Is a waiting state entered when the worker is ready but there are no orders received from the warehouse yet. Two transitions exit the state going to the `ORDER_READY` state where `THERE_ARE_ORDERS` is added to avoid race conditions.

²Since there are still no the official naming convention for enumerations in the *Java* language and since the states and events are more used as final static constants, they are named as such with full uppercase characters and with underscore between words.

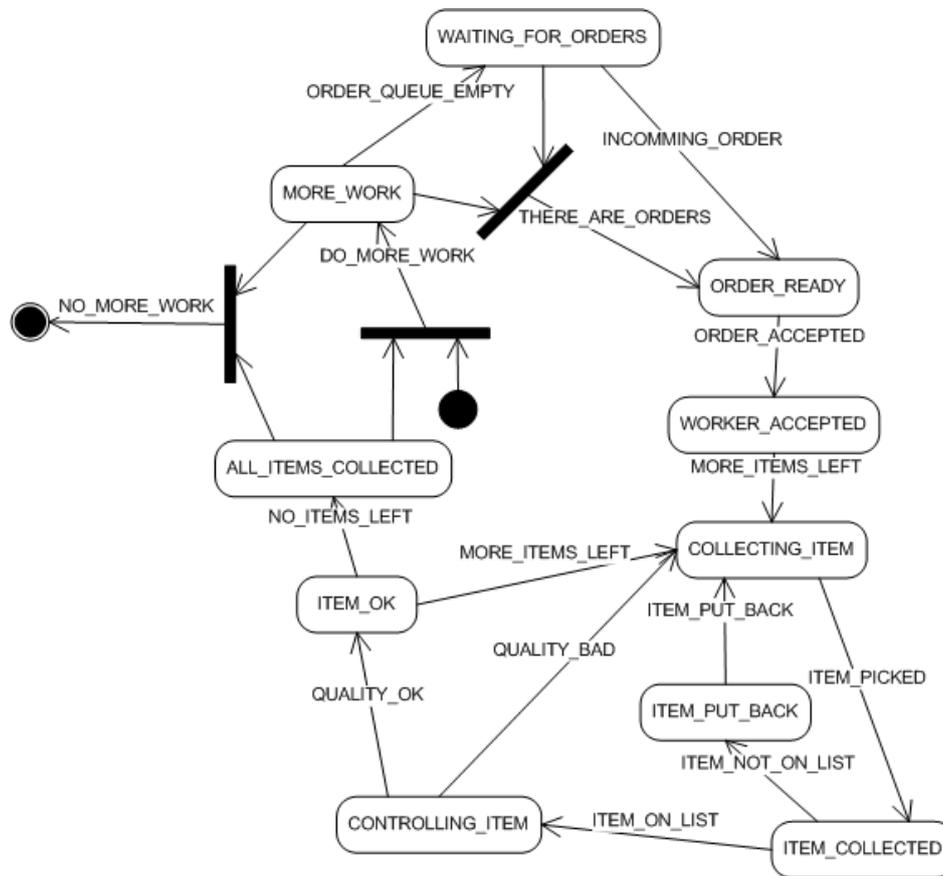


Figure 4.5: Worker state diagram

ORDER_READY Is representing the state where the worker has received atleast one order, but have not yet accepted to start collecting the items from the next one.

WORKER_ACCEPTED Here the state represent when the worker has accepted the next order and the Worker has has registered the order to be sent out to all listening services.

COLLECTING_ITEM This is the state where the worker is travelling toward the next item in the order.

ITEM_COLLECTED When the worker picks up an item the system enters this state. Here it is still possible that the systems registers that it is the wrong item and notifies the worker to put the item back.

ITEM_PUT_BACK Is the state the system enters if it registered that the item is not on the current order.

CONTROLLING_ITEM The worker has the possibility to check the quality of the picked item and decide whether it is good enough or whether there are errors like an broken item or one lying in the wrong box. The state diagram may be extended on this point with additional states to react on errors, if quality control is to be tested.

ITEM_OK If the quality is good enough for the picked item this state is entered.

ALL_ITEMS_COLLECTED Is entered by an event sent by the `Worker` class when all the items for the current order have been collected.

WORK_DONE Is the final state which can only be entered if the worker input service sends an event that no more work is to be done.

4.2.1.3 Events

The events which can trigger a state change in the state machine consists of a mixture of events from conditions the system reacts on and translated events from the worker input service. All of the possible triggering events are shown on the state diagram on fig. 4.5.

say something more here?

4.2.2 WorkerInputService

The service for the worker input is very simple. Only two classes and an input enumeration are involved, where one class is for the DWARF communications and the other the GUI for giving the input. The events sent are the enumerations `WorkerInput` serialized.

The GUI consists of buttons with one of the `Enums` for the input associated and sent to the service when pressed. With this mechanism one of the experiment conductors have to press the right button when the worker would otherwise have given the system input.

The simple output from services giving the input from the worker, makes it easy to extend or replace these services in case another kind of input should be tested such as speech recognition or touch pads attached to the workers body.

Several services could also be used in conjunction to provide the different kinds of input. For example if a system is used where the shelves registers when an item has been picked, but the worker has an additional button in case the shelves registrations did not work.³

4.2.3 Warehouse

The Warehouse service has is representing the incoming information to the worker from outside the picking area, and as such is responsible for sending orders to the worker service for the worker to process. The structure is simple and there are only three main classes in a standard model, view and controller design.

The `Warehouse` class is responsible to read information about available items and orders from a XML file⁴. This is done statically when the warehouse class is loaded and it creates one `Item` object for each item described in the file and `Order` objects with the lists containing the right items.⁵ It then provides means of getting this information either as lists or via the id of the orders/items as described in the XML file.

³This is the case in the warehouse systems visited during the system development.

⁴The `warehouse.xml` is used and should contain the orders and items as described in the XML Schema definition (XSD) in appendix B.3.1.

⁵See section 4.1 on page 27 for information of the structure of orders and items in the framework.

The class providing the connection to DWARF as a service is the `OrderSource` class. The only service ability it has, is sending serialized orders as described in section 4.1.1.

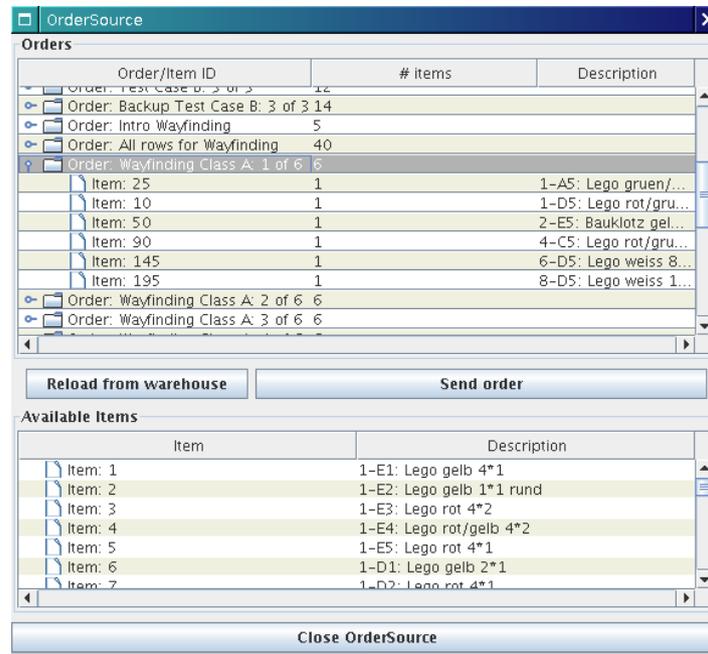


Figure 4.6: Warehouse GUI

The `OrderSourceView` class creates a GUI as seen on fig. 4.6. It gives an overview of what orders are available together with which items they are associated with. Additionally it provides means of reloading the information from the warehouse file and specifying which order to send out in the DWARF network.

4.2.4 PoseFactoryService

The `PosefactoryService` is a central service in the DWARF network. It has the role of computing the relevant positions in the form of `PoseData`⁶ from the current pose information from the tracker and what `Item` is the next to pick in the current `Order`.

These computations are centralized in this service since many of these would be replicated for every display service running, and the model of the shelf system which is build from the `WorldLocations.xml` file may take a longer time to be loaded since each box position out of several hundred needs to be computed. The `WorldLocations` class is responsible for loading the XML file and create the map from items to boxes. This class is also used in the complete occlusion model for the shelves as described on page 37 about visualization for the HMD.

⁶See the appendix C on information on how these computations are done.

4.2.4.1 Needs and Abilities

The abilities of the service consist of two types. `PoseData` and serialized `Box` information. The boxes are described in section 4.1.3 on page 28 but shortly said, they contain the pose information for this box in the tracking frame plus indexes in the shelf system.

The abilities of the service consists of the following:

ClosestWorkerBox With this ability the service provides information on which box is currently the closest to the worker. Of course this requires the worker to be tracked, but it can be done with any of the needs for `PoseData` since they all update the information for this ability.

ItemBox Is the box associated with the next item to pick according to the `Order` received from the need `Orders`.

ItemPose This is a convenience light weight ability which just sends the pose of the next item to pick without the additional information about where it is in the shelf system.⁷

HMDPose The service also provides the pose data for the tracked HMD. This is mainly for legacy reasons since many of the computations was done by the `Ubitrack` subsystem earlier, and the `PoseFactoryService` provided a central point to get this information. The pose data can just as well be taken directly from the `ARTTracker` service.

PDAPose Is provided for the same reasons as the `HMDPose` above, but with the addition that the pose data is changed so the local y-axis points in the direction of the top of the PDA. This transformation is defined as a matrix (in the `toCenter` of the PDA element) in the configuration file.

WorkerNosePose Since the tracked markers are defined in some coordinate frame from to the marker setup, they can not directly be used to determine the viewing direction of the worker. This ability provides the viewing direction by computing it from the incoming tracked markers as described in appendix C. It can use all the tracked objects in this computation⁸ and always takes the newest from the tracking system. The transformations it uses are all read from the configuration file.

The needs for the service to provide the above abilities are:

Orders The current order is needed from the `WorkerService`. When a change in an order is received, the box containing the next item is recomputed and sent out as an event.

RecHMDPose The pose data for the HMD. It is needed to compute the closest box to the worker and in which direction the worker is viewing (`WorkerNosePose`).

RecPDAPose The pose of the PDA is also used to compute the closest box to the worker and the viewing direction, although when this pose is provided it becomes the direction where the PDA is pointing.

⁷The additional information about index in the self system is for example needed for the one of the occlusion models in the visualizations for the HMD

⁸For this reason only the used markers should be visible, since any non used marker would cause the service to switch between two viewing directions

RecWorkerPose Is used when the worker is wearing a tracked marker other than the HMD and PDA. As the poses above it is also used to provide the viewing direction and closest box.

4.2.5 Display Services

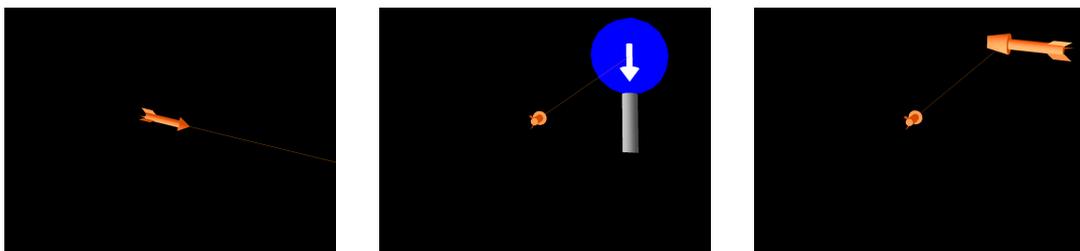
As the last part of this Thesis is based on performing user experiments with display technologies using this framework. Three different display services have been implemented.

These services all provide several different visualizations for the displays and the configuration file (*MasterConfig.xml*) can be used to change the settings of which visualizations to use. These visualizations have been designed to pass into three groups of information for the worker. 3D, 2D and list information for guidance of the worker to the right positions to pick items. This division into three groups allows a better understanding of advantages and disadvantages of the different display technologies in the later experiment. Each of the visualizations are created by classes for it in the `de.tum.in.FAR.piking.shapes` package.

4.2.5.1 HMDSERVICE

The HMD service has the possibility to display many different kinds of visualizations and any combination of these. As mentioned they can be enabled via the configuration file where the service also reads the projection matrix to use, together with information on which physical screen to display the view on and whether antialiasing should be used.⁹

For only visualizations which are not used by other of the displays services are the ones with information registered in 3D. They are:



(a) An arrow and rubberband pointing in the direction for the worker to look.

(b) Sign giving the general direction since it is standing in front of the next item to pick. The bottom part of the sign is occluded.

(c) A bigger arrow pointing into the box to pick the next item from. The front end of the arrow is occluded by the shelf.

Figure 4.7: Visualizations registered in 3D. The black areas are transparent so the real world is seen thus creating the effect of having virtual objects blended into the real world.

Road Sign Shows a sign at the position where the worker needs to go to pick up the next item.

⁹Antialiasing slows down the frame rate to a few frames per second on system that does not have direct support for it, such as the tablet PC used in the later experiments.

Item Arrow Shows an arrow in front of the box to pick an item from.

Local Arrow Gives a small arrow a few centimeters in front of the view of the HMD which points in the direction for the worker to view. When walking toward the shelf containing the next item, the arrow points above the shelf toward the road sign. When standing in front of the self the arrow changes direction to point to the box to pick an item from.

Occlusion Gives an occlusion model of the shelf where the next item to pick is. This model does not require the entire *WorldLocations.xml* to be loaded, but takes advantage of the information sent from the `PoseFactoryService`.

Full Occlusion This creates an occlusion model of the entire shelf system by loading the the XML file though the `worldlocations` class.

Rubber Band The rubber band has the same function as the small arrow and draws a line from a few centimeters in front of the view to the same locations as the local arrow described above. reference

Shelf Grid This draws a grid of small balls. One in front of each box in the relevant shelf, this should give a better 3D feeling of the other visualizations in front of the shelf.

The 2D information visualizations are shared with the remote screen service and contains the following:

Minimap A *strip map* which simply gives an overview over the relevant shelves. It does not provide any height information but just marks the stack in a shelf where the worker needs to go. Additionally a small green dot shows the current position of the worker at the moment. It is constructed as a strip map as described Golledge[13] where the travel direction goes from bottom to top on the map. The travel direction is the general direction of the items¹⁰ and not the immediate direction of the worker, i.e. the map does not turn with the viewing direction.

Height Meter Shows a vertical strip map which gives the height of the item to pick in the stack.

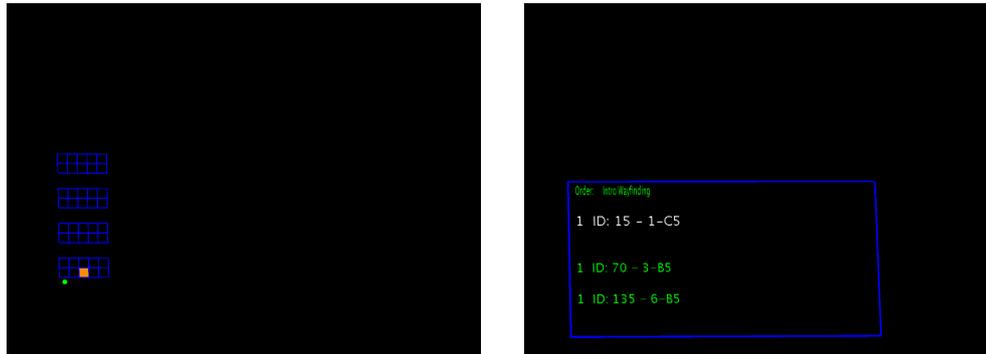
Horizontal Meter Like the height meter but shows which stack the next item is in on the shelf.

Finally the information which can be display but is not registered in space:

Item Amount This gives a number in the right top corner of the view showing the amount to pick of the next item.

List Gives a box with information about the next 3 items to pick for the current order. The list is created to contain the necessary information for finding the right items, but does not contain all the information as a real physical order as seen on image 3.2 on page 22.

¹⁰The orders should have the items ordered in this direction in the shelf system.



(a) Minimap showing the position of the worker (green dot) and the target stack in the shelf system.

(b) List of items to pick. The next item is the white one where it shows the amount, item id and item position.

Figure 4.8: REFRACTOR IMAGES...

4.2.5.2 RemoteScreenService

The remote screen service is intended to display information on several screens and is for this reason unlike the other services, a template service. This means that one instance of this service should be started for each remote screen, and the only difference is the command line parameters given at startup time to identify where in the shelf system the screen is standing (see the appendix B.1 for startup information). The positions of the screens should be one screen next to each shelf plus one at the far end of the shelves. They should be arranged so that from each corridor between shelves a screen can be seen.

Like with the HMD service, this service can enable and disable any combination of the visualizations it supports. The supported visualizations are all the ones from the HMD service except the 3D ones. Additionally it has one visualization for 3D wayfinding

Shelf model This displays a perspective view of the shelves as seen when standing in the middle of the corridor between the shelves. It consist of a combination of the height meter and the mini map and simply displays a mark for the box where an item should be picked.

The shelves model is only displayed on the screen where the next item is. All the other screens displays an arrow pointing in the direction to walk.

images

4.2.5.3 PDAService

images

The PDA service is not controlled by the configuration file like the other display services and the visualizations are chosen via a menu on the PDA. The `PDAService` only consist of

two Java classes with one connecting to the DWARF network and the other creating a TCP connection to the PDA. On the PDA a *OpenGL ES* program is running programmed in C++ and the libraries needed to run this program is detailed in the appendix B.2.

Four different visualizations have been implemented on the PDA:

Arrow Is registered in 3D and points in the direction of the next item to pick. For the later experiments this visualization was left out since it requires the PDA to be tracked, but in practise it was too often covered and the pose could be determined. As an alternative the head of the worker could be tracked, but it was too difficult to use since the PDA is not constantly aligned with the workers head.

Mini map A map similar to the one described for the HMD was developed.

Item list A list similar to the one described above but because of the limited space on the PDA screen, the item description is on a line for itself and the following items only show their id and no description, i.e. it is not possible with this setup to foresee the position of the next item, since this information is in the description.

Horizontal and vertical Shows a horizontal bar and vertical one as a combination of the height meter and vertical meter described in the HMD section above.

4.2.6 LoggingService

When doing measurements one normally needs to save the information or data that is gathered. For this reason the DWARF service `LoggingService` was created, so all interesting events in the DWARF network could be captured and recorded in a log file.¹¹

The goal of this service is to be able to create a service description file with details of what events the service should receive and more specifically what it should do with the received data. The different options include the possibility to group information according to topic, change filenames of the logfiles at runtime (from DWARF events), save the information with a timestamp and ignore duplicates if the same event information is received more than once.

A class diagram can be seen on fig.4.9 where the `LoggingService` is the main service class. At startup it connects to the service manager and receives a `NeedDescription` for each need in the service description. It then determines whether the *need* describes a new log group or a member for a log group and makes one of the two factories create a `LogGroup` or `GroupMember` accordingly. Both the `LogGroup` and `GroupMember` have an `SvcProt` associated with them and either receives or pulls for data directly from the underlying CORBA system through them. While the `LogGroup` creates the log files and has the possibility to change log files to match the events it receives, the `GroupMembers` just prefix incoming strings¹² with the given string in the `service description` attribute.

change
SvcProt to a
better name.

¹¹Although the DWARF service `DataRecorder` which also records events, already a part of DWARF it did not quite meet the needs for recording the data in this framework. For example it does not meet the requirement of changing file name of the log file at runtime, and ignoring duplicates of string events.)

¹²Currently only string events can be accepted by the `GroupMembers` (both by pulling and receiving push events), but the `NeedFactory` can easily be extended to create other types of event receivers like DWARF `PoseData` structures. They would just have to create a string from the event and send it to the associated `groups logMessage(String)` method

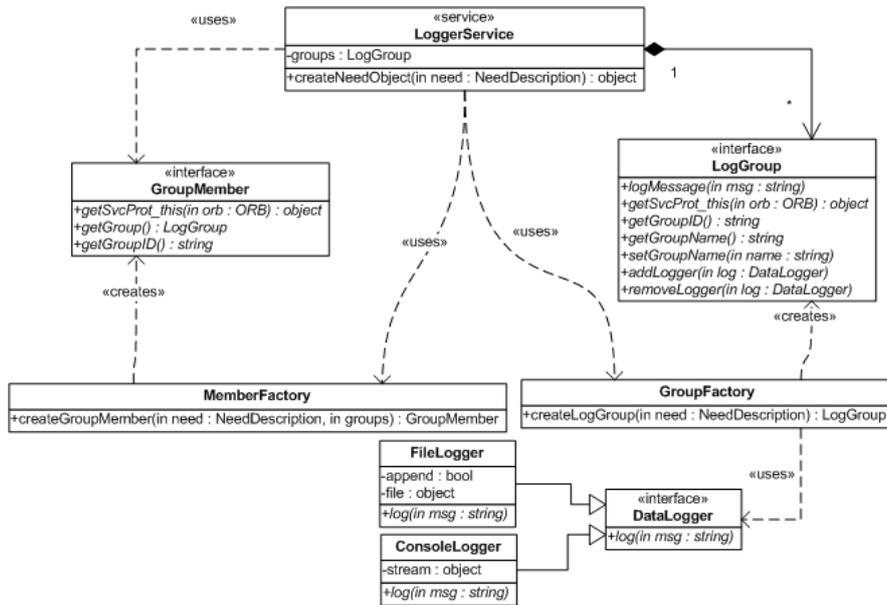


Figure 4.9: UML diagram of the LogService. Both factories have several internal classes which implements the interfaces and are created according to the NeedDescription received from the ServiceManager

The output from each LogGroup can be directed at the standard out, standard error or to a logfile. Depending on the attributes in the NeedDescription, different parts are logged with the message as seen on table 4.1. An example of use can be seen on figure 4.10 where a cut of one of the log files from the experiment is shown.

Group output : [Timestamp -][Time difference -][Group prefix -]+Member message
 Member message : [Member prefix -]+message

input
reference

Table 4.1: Log message output syntax

4.3 Tool applications

The following two tools have been implemented to ease the implementation and testing of displays. Together they provide an easy way to determine coordinates for points in the workspace in the virtual world and a method calibrating displays to align the virtual with the real world on the screen.

4.3.1 Spatial relationships by picking points

Package software properly and name it. Give step by step usage

When setting up AR applications the spatial relation between both real and virtual objects are often needed. In other cases the positions and or rotations of real object are needed in the

4 System Design

```
=====  
Log started at Fri May 19 14:45:11 CEST 2006  
Milliseconds 1148042711467  
Application 'Master' in subsystem 'Logging'  
Group: HMD_3D_2 - id: worker  
=====  
1148042711461-111079-Worker-HMD_3D_2-State:-MORE_WORK  
1148042711512-111127-Worker-HMD_3D_2-State:-ORDER_READY  
1148042712280-111896-Worker-HMD_3D_2-State:-WORKER_ACCEPTED  
1148042712294-111909-Worker-HMD_3D_2-State:-COLLECTING_ITEM  
1148042737087-136702-Worker-HMD_3D_2-State:-ITEM_COLLECTED  
1148042737090-136705-Worker-HMD_3D_2-State:-CONTROLLING_ITEM  
1148042737096-136712-Worker-HMD_3D_2-State:-ITEM_OK  
1148042737100-136715-Worker-HMD_3D_2-State:-COLLECTING_ITEM  
1148042749087-148702-Worker-HMD_3D_2-State:-ITEM_COLLECTED  
...  
...
```

Figure 4.10: Example output from measuring a test participant (number 2) while using the HMD and visualizations with 3D navigation. Here all the possible options are enabled so both timestamp and time difference is logged.

coordinate frame of the virtual world. In both of these cases the tool presented here comes in handy. It is a simple application which can capture positions of different tracked objects and computes the spatial relationship between them. These results can then be viewed immediately via and HMD, projector or similar and the computed transformation matrices are shown in the GUI for easy copying into other applications, see fig. 4.11.

The top section is for the pointing device. It is used specifying transformation matrix from the end-tip of the pointing device to the origin of the tracked marker, and the *Tracker ID* of the device. The tracking system used is ART like in the other systems, and the application is setup to start receiving the tracking data on port 5000. The label by the tracking id's for all tracked objects will turn green when the items are registered (within the last 250ms). When changing either the id of the pointing device or transformation matrix, the reload button should be pressed to reset the tracking adapter and view.

The second part is for visualizing the *pose* that are computed. It enables a *Java3D* view to turned on fullscreen on the selected monitor and uses the projection matrix set in the textbox.

Finally is the part with the target to which the spatial relation should be made. Here an object can be tracked and the captured *pose* will be in relative to that objects *pose* or the global position, ie. relative to the coordinate frame used by the tracking system. The *pose* is computed when the *Grab new point* button is pressed and the resulting transformation matrix is displayed in the textbox, in case the view is used a small coordinate system will be shown at the captured *pose*.

For the display and visualization framework used for the experiments, this spatial relationship tool was used to find the viewing direction of the HMD. Simply by capturing the *pose* while the pointing device was in front of the HMD and the HMD was set as the target, a spatial relation from the HMD marker to the viewing direction was computed. Furthermore the tool was used to find and verify the global position of the shelves compared to the computed in the program.

get image for
the j3d view

insert image of the transformations and relations

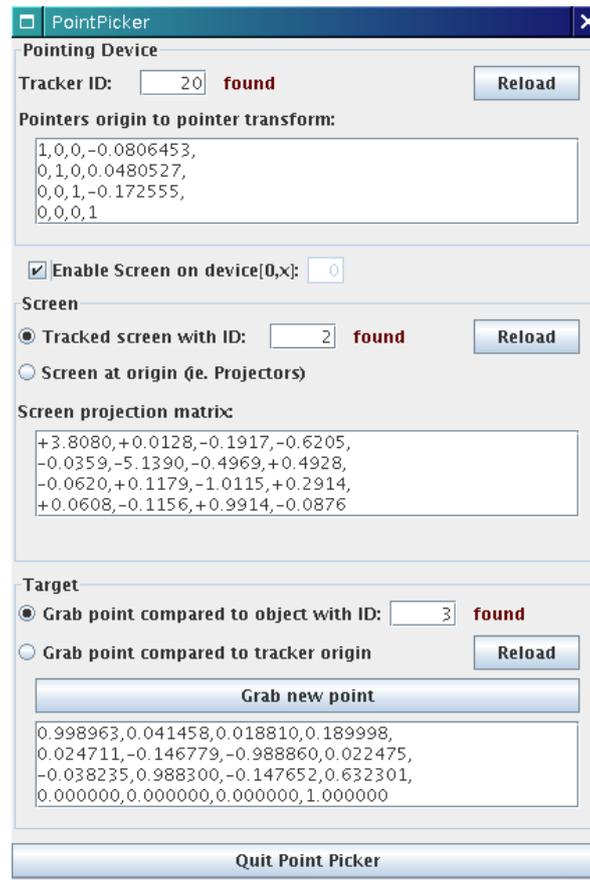


Figure 4.11: Spatial Relationship GUI

4.3.2 SPAAM calibration

When using optical see-through HMDs a human computer interaction is needed in the calibration to align the virtual with the real world on the screen. One of the methods for doing this calibration is the Single-Point Active Alignment Method (SPAAM) [26] which is based on aligning single points on the screen to tracked real world objects or otherwise known points in the virtual world. The SPAAM calibration tool implements¹³ this method and provides an easy to use GUI to determine the calibration matrix for viewing devices, see fig. 4.12.

Reference
daniel- how?

Most of the configuration of the application can be done either through the accompanying XML file (*config.xml*), or directly in the GUI. There are several different features like changing the ID of the tracked target or screen, setting an offset compared to the target, setting screen resolution and changing the number of points relations which should be captured for the calibration.

When first starting to capture points, a view like on fig. appears on the selected monitor. The point should then be aligned with the tracked target (plus offset if one is given) and

get image

¹³The algorithm is implemented by Daniel Pustka

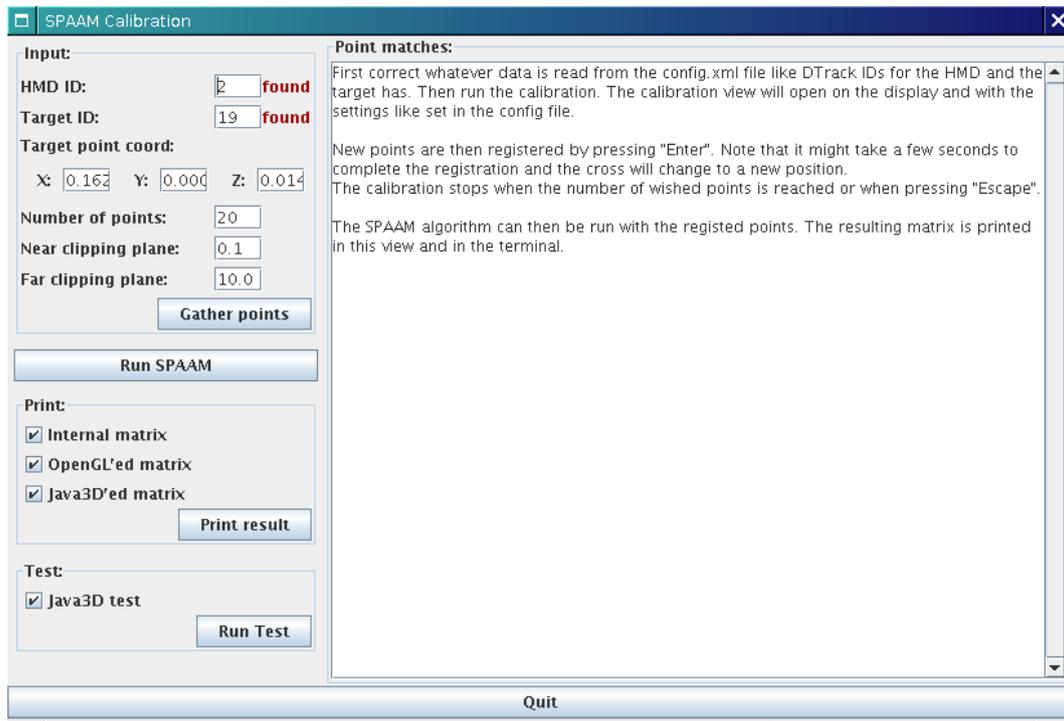


Figure 4.12: SPAAM calibration GUI

the relation is captured by pressing `enter`. When all the needed relations are captured the SPAAM algorithm can be run and finally the result can be printed out.

Note that the third row of the projection matrices differs for *Java3D* compared to *OpenGL*. This is because *OpenGL* uses a left-handed coordinate system in the clipping space, while *Java3D* is a pure right-handed framework so the third row is flipped compared to the one for *OpenGL*. (Actually *Java3D* uses *OpenGL* as an underlying framework so it flips the row back again before sending the matrix down the *OpenGL* pipeline).

Finally the program can run a small test for the newly gained projection matrix. This can be either with a direct implementation in *SWING* which is very slow, or with *Java3D*. In both cases a coordinate system is shown in the center of the tracked target and with *Java3D* a rotating cube next to it and a small red ball in the offset point.

image?

The application has been tested and used to find the calibration matrix for both the optical see-through monoscopic HMD used in the experiments and a ceiling mounted projector projecting onto a table beneath it.

For the HMD calibration a tracked wooden board was tracked with a crosshair in the middle¹⁴ and the point was aligned using the left eye. During the calibration the calibration board was moved around in different distances to the HMD to avoid co-planer points and 30 points relations were captured. The procedure was done with several different resolutions for the HMD all producing adequate results for use in the experiments.

Using the tool for calibrating a projector was also a success and the *Java3D* test augmented

¹⁴The point was found using the tool described in section 4.3.1

the tracked object (a pointing device) both while lying on the table as when in the air with the right `pose`. While calibrating a non-tracked ID was given as the ID of the screen, so the application used the identity matrix for the position of the projector.

5 Usability in Emperical Experiments

Description and introduction of concepts for how emperical measurements are used usability experiments

Asking and answering questions to try and understand the nature of objects around us, has probably been done since the epoch of civilization. It is also an activity that occupied the ancient greeks, although they often had a more philosophical approach than research has today.

The kinds of question we ask are as many as the kinds of things which we know. They are in fact four.

Aristotle in Posterior Analytics, book 2[1]

The first kind of question identified by Aristotle is whether there is a connection from certain attribute to a *thing*, but it is in fact not always an easy question to answer. For answering these questions, the science of empericism has been developed as a contrast to logical deduction, and bases the answers on conducting observations of the connection in question.

This chapter takes a more indepth view of why empericism is used, the ANOVA method for validating the results and how it relates to conducting usability experiments. Several concepts are introduced which are necessary to understand the following chapter where the two experiments for display technologies in picking are described.

5.1 Empericism for Human Subject Experiments

Most of the natural sciences are analytically based which also counts for computer science, but as soon as humans subjects becomes involved in the studies an emperical approach becomes nescesary with a solid foundation on statistics. For computer science it is especially the field of HCI where emperical experiments are commen, but also for topics like VR these experiments are becomming more frequent and for example 45% of the papers for the IEEE conference on VR (2005) contains experiments with humen subjects[24].

To conduct an humen subject experiment with an emperic validation of the results, the used technique can be devided into the following phases:

- A testable hypothesis should be developed on the dependant and independant variables in the experiment. It is the one factor which should be accepted or refused to validate the results.

The independant variables are varied in different parts of the experiment, and the dependant ones are measured for each of these variations.

- Setup and perform the experiment collecting the measureable data.
- Analyse the data and apply tests on the hypothesis with significance tests. This step can also include meta statistical analysis by describing the data in summaries and via graph plots and interpretation of the tendency toward the center (mean, median and mode) together with the variance and deviation. While the significance tests are nescesary to validate the experiment, the descriptive part helps to understand the data distributions.
- Finally the experiment hypothesis is either accepted or refuted and the results are related back to the experiment itself and to theories on the topic.

There are different ways to test the hypothesis depending on what the experiment measures and how many dependant and independant variables there are. In the comen case with two dependant variables in the experiment, a *t-test* is the most comen method. For experiments with more independant variables *one-factor* or *multi-factor* ANOVA can be used.[21, 10]

Commen for these two kinds of tests is that a testable hypothesis (H_1) is needed and from that the logical opposite *null hypothesis* (H_0) is created. From H_0 the *signifiance* (α) of the test is determined and if it is less than 5%¹, then H_0 is rejected and H_1 is thereby accepted. If H_0 is refuted, but is in fact true, it is called a *Type I* error. Oppositely if H_0 is accepted, but is fact false, it is called a *Type II* error.

5.1.1 Factorial Analysis of Variance (ANOVA)

When there are more than two independant variables in the experiments, it is not possible to compare and validate the results via a single *t-test* and if one combined several test then the risk of error would increase[21]. In these cases a *Analysis of Variance (ANOVA)*² must be used to determine the validity for the means of the data, this is done by splitting the observed variance as described in the following. ANOVA may be considered a generalization of the *t-test* and can be used to both compare multiple dependent or independent variables[24]. In the scope of this Thesis only the one-factor ANOVA is treated since the later experiments only have one factor to vary: the visualizations used.

Before carrying on with the method description, following there are a few definitions and basic concepts to make the discussion about the method more clear:

Factor Is the term used for independant variables when dealing with ANOVA and when only having one dependant variable to test, it gives the name to the ANOVA method (one,two,...,multi factor ANOVA). This is the value which is to be changed by the experimenter to different levels.

Level A certain level of the factor (numerical) and it is often either present or absent.

¹The value of $\alpha < 0.05$ is arbitrary choosen and comen values include 0.1, 0.5 and 0.01, but it depends on what the field of study is.

²ANOVA is also sometimes referred to as the *f-test*, because it is based on the F-distribution.

Treatment Is a combination of levels from different factors which should be compared with other treatments in the test.

Mean Squares (MS) The commenly used value in statistics dependent on the also comenly used standard deviation s :

$$MS = s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1} = \frac{\text{Sum of Squares (SS)}}{\text{Degrees of Freedom (DF)}}$$

where X_i is a measured value, \bar{X} the mean and n the number of samples measured.

As mentioned two hypothesises are needed for ANOVA and with the one-factor ANOVA the first one is the testable one which typically have the form $H_1 : f(\mu_1, \mu_2, \dots) \neq 0$ and the null hypothesis which have the form $H_0 : f(\mu_1, \mu_2, \dots) = 0$. The μ 's are the means in the different treatments.

Like with the *t-test*, ANOVA aims at finding one value to reject or accept the null hypothesis. In this case this value is called F and the acceptable values can be seen on F-distribution tables.³ Note however that like with the standard t-test, the only knowledge gained from rejecting the null hypothesis is that there *is* (with $1 - \alpha$ probability) a measureable difference in the means of tested treatments, and not to what extend the measured means are correct.

The method of ANOVA is based on the *total MS* of all the measurements performed, or more precisely it partitions the total MS into different components resposible for the variation. First the numerator of the total MS (SS_{total}) is refactored into a part depending on the treatments and a part dependent on the SS of the different levels ($N = \sum n_i$ is the total number of measurements, \bar{X} is the total mean, \bar{X}_i is the mean of treatment i and k is the number of treatments):

$$\begin{aligned} SS_{\text{total}} &= \sum_{i=1}^N (X_i - \bar{X})^2 \\ &= \sum_{j=1}^k \sum_{i=1}^n (X_{ji} - \bar{X})^2 \\ &= \sum_{j=1}^k n_j (\bar{X}_j - \bar{X})^2 + \sum_{j=1}^k \sum_{i=1}^n (X_{ji} - \bar{X}_i)^2 \\ &= \text{Sum of Squares of Treatment (SST)} + \text{Sum of Squares Error (SSE)} \end{aligned}$$

Now the mean squares can be found of these two components giving the variance in the treatment and in the error. This is simply done by dividing them with their corresponding

³See for example the *Engineering Statistics Handbook*[21] for this tables over the F-distribution.

degrees of freedom:

$$\begin{aligned}
 MST &= \frac{\text{Sum of Squares of Treatment (SST)}}{\text{Degrees of Freedom Treatment (DFT)}} \\
 &= \frac{\sum_{j=1}^k n_j (\bar{X}_j - \bar{X})^2}{k - 1} \\
 MSE &= \frac{\text{Sum of Squares of Error (SSE)}}{\text{Degrees of Freedom Error (DFE)}} \\
 &= \frac{\sum_{j=1}^k \sum_{i=1}^n (X_{ji} - \bar{X}_j)^2}{N - k}
 \end{aligned}$$

Finally coming back to the F value for the hypothesis test. It is defined as the relation between the meas squared treatment and the mean squard error:

$$F = \frac{MST}{MSE} \quad (5.1)$$

It is not directly obvious whether the value of F rejects or accepts H_0 , but it should be compared against (higher than) the corresponding values in the F-distribution tables. The value in the tables are based on both DFT, DF and the required significance level α for the experiment.

The values found in the ANOVA test is usualy filled into an ANOVA table as:

Source (of variance)	SS	DF	MS	F
Treatment	SST	$k - 1$	$\frac{SST}{k-1}$	$\frac{MST}{MSE}$
Error	SSE	$N - k$	$\frac{SSE}{N-k}$	
Total	SS_{total}	$N - 1$		

5.2 Conducting Usability Experiments

Determining how usable a product is, is one of the kinds of questions requiring an emperical evaluation to answer. To this means the ISO standard *ISO 0241-11*[15] can be used to make it clear on what is meant with usability:

Effectiveness Measures of effectiveness relate the goals or subgoals of the user to the accuracy and completeness with which these goals can be achieved.

Efficiency Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources. Relevant resources can include mental or physical effort, time, materials or financial costs.

Satisfaction Satisfaction measures the extend to which users are free from discomfort, and their attitudes towards the product.

The goal is for example to have a measurement of the acceptability of the workload during different tasks using the given system.

These charecteristics should be all be included in some form when evaluating usability. For example, values for effectiveness could be the amount of type errors in a typing test, efficiency values could then be the amount of pages produced divided by the amount of time or keyboard costs. Satisfaction is a more subjective measurement and could be found via questionnaires or in long-term experiments by for example observing the amount of time the user is absent from work.

For usability studies in relation to the picking process, the measures could then include⁴:

- Effectiveness:

Accuracy By the number of falsely picked items or positions.

Completeness By the number of succesfully completed orders. This could be extended if intrducing a hard deadline in the setup, and if broken it counts as an error for the completeness.

- Efficiency:

Temporal By noting the time for each pick and other change in state for the worker. These times can them be used to find the average times of the different phases of the picking process (see page 19). The measurable states are the ones represented by the `WorkerService` and can be seen on page 31.

More???

- Satisfaction:

Acceptability Through questionnaires it could be tested to what extend the users would accept working with the system. But one would have to conduct long term experiments to get a proper feedback of this.

Discomfort Also through questionnaires it tests whether the use of the system causes problems of discomfort. This is proberbly especially an issue for the HMD, but also using a PDA could give long term issues like having to lift the same arm constantly to see the information.

Learnability An evaluation of how easy the system is to use and or learn to use. An estimate could be done via questionnaires, but it could also be done as an efficiency measurement, for example by seeing the how the accuracy changes over time.

⁴Note however that this is not an exhaustive list of measureable charecteristics of the picking process. Other kinds of charecteristics could also play a big factor, like costs.

6 Experiments for Wayfinding and Picking

Two experiments were conducted to measure usability of displays types in picking

6.1 Experiment Setup

6.1.1 Wayfinding

6.1.1.1 Setup

6.1.1.2 Plan

6.1.2 Picking

6.1.2.1 Setup

6.1.2.2 Plan

Wayfinding describes out map type as strip maps page 14-15
--

6.2 Experiment Results

6.2.1 Wayfinding

6.2.1.1 HMD

6.2.1.2 Remote Screens

6.2.1.3 PDA

6.2.2 Picking

6.3 Displays

6.4 Overview and sum up

7 Conclusion

add something usefull here...

7.1 Future Work

A Experiment Questionnaires and Results

The questionnaires used for evaluating satisfaction among the test participants together with result from them and other measurements

A.1 Questionnaires

The following pages shows the questionnaires given to the test subjects in the experiments. The first gives information about the participant and was filled out before the experiment began. The second is a questionnaire about what the user thought about the different visualizations used for a certain display. Each participant was given one of these for each different kind of display they tried and had to fill it out directly after the test with the specific display. The last questionnaire is an overall questionnaire about the display technologies and what the test participant thought about it.

Several topics have been cut of the questionnaire to avoid getting the experiment to last too long (preferably not longer than one hour), but still as they are several people took around 15 minutes per questionnaire, and since the wayfinding experiment has five questionnaires¹ per person the time was reaching its limits, and made it very difficult to plan a schedule.

Finally a small interview was done with each test participant after they had completed the experiment. The interview was based on what they had filled out in the questionnaires but always included the following topics:

Advantages and disadvantages In general a small talk about the advantages and disadvantages. It is intended to give a deeper view of why the test participant has filled out the questionnaires as they did.

Working Whether they could imagine working with some or all of the different display technologies. Why and why not?

Long working days Did they get any physical problems like head aches or dizziness, or could they imagine using the system for longer periods of a time.

Cognitive workload Did they need to concentrate on using the different techniques or was it easy?

¹There was a questionnaire for each display type, one final about the displays and of course the one with information about the participant.

A.1.1 Participant Information

Test participant ID: ____

Participant Information:

Age: ____

Gender: Male: __ Female: __

Occupation: _____

Education (highest completed level): _____

Computer games experience (1: Every day – 5: None)

- I actively play computer games 1 5
- I used to play computer games 1 5

Of computer games I have experience with (1: Only – 5: None)

- 3D Computer Games 1 5
- First person Games 1 5

I have used a (1: I use it every day – 10: I never tried)

- Computers 1 10
- Head mounted display 1 10
- Personal digital assistant 1 10
- Mobile phone 1 10

I am using: (1: everyday – 5: once a week – 10: never)

- Virtual Reality systems 1 10
- Augmented Reality systems 1 10

I work with with: (1: everyday – 5: once a week – 10: never)

- Developing User Interfaces 1 10
- Picking 1 10

A.1.2 Visualization Questionnaire

Test participant ID: ____

Visualization Questionnaire:

I found the following technique helpful: (1: Very helpful – 5: Not usable)

- Mini map 1 5
- 3D navigation 1 5
- List 1 5

Which technique did you like the most and why: (1: Very much – 5: Not at all)

- Mini map 1 5
- 3D navigation 1 5
- List 1 5

• Order of preference: _____

Impressions of the user interface:

- At first impression I found following confusing (1: Intuitive – 5: Confusing)
 - Mini map 1 5
 - 3D navigation 1 5
 - List 1 5
- At second impression I found following confusing (1: Intuitive – 5: Confusing)
 - Mini map 1 5
 - 3D navigation 1 5
 - List 1 5

I found the following confusing or in the way:

Comments:

A.1.3 Display Questionnaire

Test participant ID: ____

Display Questionnaire:

I found the following technique helpful: (1: Very helpful – 5: Not usable)

- HMD 1 5
- Remote screens 1 5
- PDA 1 5

Which technique did you like the most and why: (1: Very much – 5: Not at all)

- HMD 1 5
- Remote screens 1 5
- PDA 1 5
- Order of preference: _____

I had problems relating the displayed information with the shelves:

(1: No problems – 5: Many problems)

- HMD 1 5
- Remote screens 1 5
- PDA 1 5

A.2 Wayfinding Experiment Results

For the wayfinding experiment 18 people participated, although 6 people did not complete the PDA part of the tests.

A.2.1 Questionnaire Results

In the wayfinding experiment there was 12 male and 6 female participants who filled out the participant information as seen in table A.1, the HMD visualization questionnaire in table A.2, remote screen visualization questionnaire in table A.3.

The questionnaire about visualizations on the PDA in table A.4. Of the 18 people in the tests, 12 of them performed the PDA test.

All the percentage results are rounded to one significant digit after the comma and averages to two.

	Score (in %)									
	1	2	3	4	5					
Computer games user										
Active player	11.1	11.1	11.1	44.4	22.2					
Used to play	33.3	27.8	11.1	16.7	11.1					
3D games	11.1	38.9	11.1	16.7	22.2					
First Person games	22.2	22.2	11.1	11.1	33.3					
	1	2	3	4	5	6	7	8	9	10
Usage of ...										
Computers	94.4	5.6	0	0	0	0	0	0	0	0
HMDs	0	0	0	11.1	0	16.7	5.6	5.6	11.1	50.0
PDA's	11.1	5.6	0	5.6	5.6	0	0	11.1	22.2	44.4
Mobile phones	77.8	11.1	0	0	0	0	0	0	5.6	5.6
Usage of ...										
VR	5.6	0	0	5.6	0	5.6	0	0	27.8	55.6
AR	0	0	0	0	16.7	16.7	0	11.1	11.1	44.4
Work with ...										
User interfaces	5.6	5.6	11.1	0	5.6	0	0	38.9	0	33.3
Picking	0	0	0	0	5.6	0	0	22.2	0	77.8

Table A.1: Participant information for the Wayfinding experiment

A.2.2 Errors Measurements

	Score (in %)					Average score
	1	2	3	4	5	
Technique helpful						
Mini map	16.7	44.4	16.7	22.2	0	2.44
3D navigation	0	11.1	44.4	38.9	5.6	2.39
List	22.2	50.0	11.1	11.1	5.6	2.28
Liked technique						
Mini map	16.7	44.4	16.7	5.6	16.7	2.61
3D navigation	16.7	5.6	33.3	27.8	16.7	3.22
List	27.8	38.9	11.1	16.7	5.6	2.33
Confusing at first impression						
Mini map	27.8	27.8	33.3	5.6	5.6	2.17
3D navigation	0	16.7	11.1	38.9	33.3	3.89
List	61.1	22.2	11.1	0	5.6	1.67
Confusing at second impression						
Mini map	27.8	22.2	22.2	22.2	5.6	2.56
3D navigation	0	38.9	38.9	5.6	16.7	3.00
List	50.0	33.3	11.1	0	5.6	2.78

Table A.2: HMD visualization questionnaire results

A Experiment Questionnaires and Results

	Score (in %)					Average score
	1	2	3	4	5	
Technique helpful						
Mini map	22.2	44.4	22.2	11.1	0	2.22
3D navigation	16.7	22.2	33.3	11.1	11.1	2.78
List	27.8	44.4	22.2	5.6	0	2.05
Liked technique						
Mini map	33.3	22.2	16.7	22.2	5.6	2.44
3D navigation	27.8	16.7	22.2	22.2	11.1	2.72
List	16.7	27.8	44.4	11.1	0	2.50
Confusing at first impression						
Mini map	27.8	44.4	5.6	22.2	0	2.22
3D navigation	22.2	16.7	27.8	22.2	11.1	2.83
List	61.1	22.2	5.6	11.1	0	1.67
Confusing at second impression						
Mini map	38.9	44.4	5.6	5.6	5.6	1.94
3D navigation	27.8	16.7	33.3	11.1	11.1	2.61
List	44.4	38.9	11.1	5.6	0	1.78

Table A.3: Remote screen visualizations questionnaire results

	Score (in %)					Average score
	1	2	3	4	5	
Technique helpful						
Mini map	25.0	41.7	8.3	25.0	0	2.33
List	25.0	50.0	25.0	0	0	2.00
Liked technique						
Mini map	41.7	25.0	16.7	8.3	0	1.75
List	33.3	41.7	16.7	8.3	0	2.00
Confusing at first impression)						
Mini map	16.7	41.7	25.0	0	16.7	2.58
List	50.0	41.7	8.3	0	0	1.58
Confusing at second impression						
Mini map	16.7	41.7	8.3	33.3	0	2.58
List	50.0	50.0	0	0	0	1.50

Table A.4: PDA visualizations questionnaire results with 12 test participants.

	Score (in %)					Average score
	1	2	3	4	5	
Technique helpful						
HMD	25.0	8.3	41.6	16.7	8.3	2.33
Remote Screens	25.0	33.3		0	0	2.50
PDA	33.3	50.0	8.3	8.3	0	1.92
Liked technique						
HMD	25.0	8.3	25.0	16.7	25.0	2.58
Remote Screens	25.0	25.0	25.0	16.7	8.3	3.08
PDA	25.0	41.7	33.3	0	0	2.08
Problems with relating information)						
HMD	0	25.0	50.0	25.0	0	2.58
Remote Screens	41.3	16.7	16.7	25.0	0	3.00
PDA	58.3	16.7	16.7	8.3	0	1.75

Table A.5: Display technology questionnaire with 12 participants. See the questionnaire on page [55](#).

A.2.3 Times Measurements

Since the experiment contains 18 (12 for the PDA) people, 3 display types, 3 (2 for the PDA) visualizations for each display, 3 orders for each visualization and 6 positions for each order, there was measured 2376 times. It would not be feasible to bring all these here, instead a thorough sum up is shown in the following tables.

Note however, that a few of the position times are filtered out, which is done in cases where the time is 0 or a few milliseconds from it. This might have happened if the experimenter has multi clicked the button when an item has been picked, and since the visualizations will not have changed it will not give an influence on the subject, only on the overall time for the order (which is not used in the statistics).

	Mean (<i>ms</i>)	Median	Mode	Mean Abs. Dev.	Std. Dev.
HMD with 3D visualization					
Order 1:	18354.37	11627	10000.0	11703.33	21790.76
Order 2:	8954.09	8010	6000.0	3577.07	5347.51
Order 3:	8525.70	7628	8000.0	2945.81	4291.17
Total:	11944.72	8816	6000.0	6125.38	13911.10
HMD with 2D visualization					
Order 1:	7138.96	6190	5500.0	2684.12	3460.10
Order 2:	6296.64	5385	6000.0	2760.06	3951.88
Order 3:	6836.83	5927	5500.0	2748.53	4975.79
Total:	6757.48	5898	5500.0	2829.27	4187.44
HMD with a list					
Order 1:	8457.25	6190	5500.0	2684.12	5359.56
Order 2:	5588.02	5307	3500.0	1838.93	2169.24
Order 3:	5411.75	5349	5000.0	1835.81	1665.34
Total:	6485.67	5976	5500.0	2184.43	3760.12

Table A.6: Wayfinding experiment with HMD. For the mode, the results are rounded to nearest 500ms. Each order has 18 people with each 6 positions to find giving 108 positions and 324 positions for each group.

Source (of variance)	SS	DF	MS	F
Treatment	6083719428.54	2	3041859714.27	40.45
Error	72271410735.8	961	75204381.62	
Total	78376068104.6	962		

Table A.7: ANOVA table for the HMD in the wayfinding experiment.

	Mean	Median	Mode	Mean Abs. Dev.	Std. Dev.
Remote screens with 3D visualization					
Order 1:	6862.81	6105	3500.0	3051.96	4110.56
Order 2:	5149.71	4595	2500.0	2226.57	3284.22
Order 3:	5391.06	4975	2000.0	2157.16	3205.26
Total:	5801.19	5266	2500.0	2494.84	3640.41
Remote screens with 2D visualization					
Order 1:	4776.81	4653	3000.0	1561.96	2065.64
Order 2:	4496.16	4088	2500.0	1603.97	2096.87
Order 3:	4632.85	4439	2500.0	1613.58	2186.95
Total:	4635.27	4439	2500.0	1594.94	2138.37
Remote screens with a list					
Order 1:	4986.57	4855	4500.0	1308.65	1682.09
Order 2:	4488.07	4282	3000.0	1531.32	1993.50
Order 3:	4638.15	4273	4500.0	1504.99	2230.80
Total:	4704.26	4591	4500.0	1457.81	2006.84

Table A.8: Wayfinding experiment with Remote Screens. For the mode, the results are rounded to nearest 500ms. Each order has 18 people with each 6 positions to find giving 108 positions and 324 positions for each group.

Source (of variance)	SS	DF	MS	F
Treatment	275752763.96	2	137876381.98	18.94
Error	7024625942.63	965	7279405.12	
Total	7294192592.96	966		

Table A.9: ANOVA table for the Remote Screens in the wayfinding experiment.

	Mean	Median	Mode	Mean Abs. Dev.	Std. Dev.
Remote screens with 2D visualization					
Order 1:	5625.03	5600	3000.0	1875.42	2620.20
Order 2:	4870.94	4586	3500.0	1639.17	2035.45
Order 3:	4950.51	4661	3500.0	1639.22	2117.49
Total:	5148.83	4944	3500.0	1745.50	2288.68
Remote screens with a list					
Order 1:	5606.46	4940	5500.0	1887.24	3692.29
Order 2:	4596.89	4518	4500.0	1281.30	1607.70
Order 3:	4596.03	4770	4500.0	1087.79	1279.73
Total:	4933.13	4779	4500.0	1434.15	2475.56

Table A.10: Wayfinding experiment with PDA. For the mode, the results are rounded to nearest 500ms. Each order has 12 people with each 6 positions to find, giving 72 positions and 216 positions for each group.

Source (of variance)	SS	DF	MS	F
Treatment	5025033.48	1	5025033.48	0.89
Error	2446061894.29	431	5675317.62	
Total	2451086927.77	431		

Table A.11: ANOVA table for the PDA in the wayfinding experiment.

A.3 Picking Experiment Results

A.3.1 Questionnaire Results

In the wayfinding experiment there was 12 male and 6 female participants for the experiment. Their information can be seen on table A.1. All the percentage results are rounded to one significant digit after the comma.

CHANGE										
	Score (in %)									
	1	2	3	4	5					
Computer games user										
Active player	0	22.2	11.1	44.4	22.2					
Used to play	0	33.3	27.8	33.3	5.6					
3D games	0	27.8	27.8	22.2	22.2					
First Person games	0	27.8	22.2	16.8	33.3					
	1	2	3	4	5	6	7	8	9	10
Usage of ...										
Computers	94.4	5.6	0	0	0	0	0	0	0	0
HMDs	0	0	0	11.1	0	16.7	5.6	5.6	11.1	50.0
PDA's	11.1	5.6	0	5.6	5.6	0	0	11.1	22.2	44.4
Mobile phones	77.8	11.1	0	0	0	0	0	0	5.6	5.6
Usage of ...										
VR	5.6	0	0	5.6	0	5.6	0	0	27.8	55.6
AR	0	0	0	0	16.7	16.7	0	11.1	11.1	44.4
Work with ...										
User interfaces	5.6	5.6	11.1	0	5.6	0	0	38.9	0	33.3
Picking	0	0	0	0	5.6	0	0	22.2	0	77.8

Table A.12: Participant information for the picking experiment

A.3.2 Errors Measurements

	Score (in %)					Average score
	1	2	3	4	5	
Technique helpful						
Mini map	0	22.2	50.0	22.2	5.6	3.11
3D navigation	16.7	44.4	22.2	11.1	5.6	2.44
List	33.3	22.2	16.7	16.7	11.1	2.50
Liked technique						
Mini map	5.6	16.7	22.2	44.4	11.1	3.39
3D navigation	33.3	38.9	5.6	16.7	11.1	2.22
List	22.2	16.7	27.8	16.7	11.1	2.61
Confusing at first impressions)						
Mini map	16.7	11.1	44.4	11.1	16.7	3.00
3D navigation	38.9	38.9	5.6	11.1	11.1	2.33
List	5.6	27.8	38.9	16.7	5.6	2.72
Confusing at second impression						
Mini map	16.7	38.9	27.8	11.1	5.6	2.50
3D navigation	50.0	11.1	11.1	22.2	0	1.94
List	33.3	27.8	16.7	5.6	16.7	2.44

Table A.13: Picking test visualization questionnaire results. Questionnaire from page 54.

A.3.3 Times Measurements

As for the wayfinding experiment there are too many times to shown them all (1620 measurements), but the following table gives a sum up of the important statistical concepts. Also here are few times filtered out (4 times are removed), with the same explanations as in

	Mean	Median	Mode	Mean Abs. Dev.	Std. Dev.
HMD with 3D visualization					
Order 1:	10394.79	8276	6500.0	4169.09	7409.11
Order 2:	7850.21	6313	5000.0	2686.93	4840.34
Order 3:	7544.07	6691	7500.0	2391.58	3840.75
Total:	8596.36	7043	5000.0	3158.29	5739.59
HMD with 2D visualization					
Order 1:	7867.11	7317	6500.0	2386.29	3023.50
Order 2:	7639.26	6842	5500.0	2232.75	5059.13
Order 3:	6047.03	5643	5500.0	1656.03	2251.19
Total:	7184.47	6565	5500.0	2078.16	3722.17
HMD with a list					
Order 1:	8673.30	7251	6000.0	3271.11	4920.00
Order 2:	6152.80	5643	5000.0	1521.14	2251.51
Order 3:	6448.44	6009	5500.0	1697.39	2539.20
Total:	7091.51	6264	4500.0	2269.67	3612.57

Table A.14: Picking experiment with HMD. For the mode, the results are rounded to nearest 500ms. The number of positions are 18 people with each 10 positions per order and 3 orders giving 540 positions for each group.

Source (of variance)	SS	DF	MS	F
Treatment	754075259.23	2	377037629.62	18.98
Error	32095733540.9	1616	19861221.25	
Total	32875908400.6	1617		

Table A.15: ANOVA table picking experiment.

B Application Usage

Additional information about the usage of the provided applications such as startup commands

B.1 Services

something

B.1.1 Startup

startup information

B.2 Required third party software

name the required software.. i.e. Java3D DWARF and so on..

B.3 XML validation

To guarantee that the read information about orders and items from the *warehouse.xml* and positional information from the *WorldLocations.xml* files are correct, they are validated against two XSD schema files during startup.

B.3.1 Schema structure for the warehouse XML files

The warehouse XML file has to conform to the XSD *warehouse.xsd* file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- An order -->
<xs:element name="order">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
```

B Application Usage

```
        <xs:attribute name="id" type="xs:string" use="required"/>
        <xs:attribute name="amount" type="xs:positiveInteger" use="required"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>

<!-- An item -->
<xs:element name="item">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="description" type="xs:string"/>
            <xs:element name="number" type="xs:positiveInteger"/>
            <xs:element name="image" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>

<!-- The root element: -->
<xs:element name="warehouse">
    <xs:complexType>
        <xs:sequence>

            <xs:element name="orders"> <!-- A list of orders-->
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="order" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>

            <xs:element name="items"> <!-- A list of items -->
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="item" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>

        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>
```

B.3.2 Schema structure for the world locations XML files

The *WorldLocations.xml* is checked up against the *worldlocations.xsd* schema:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <!-- A tuple type in 3D -->
    <xs:complexType name="Tuple3DType">
        <xs:attribute name="x" type="xs:decimal" use="required"/>
        <xs:attribute name="y" type="xs:decimal" use="required"/>
        <xs:attribute name="z" type="xs:decimal" use="required"/>
    </xs:complexType>
```

B Application Usage

```
<!-- A vector -->
<xs:complexType name="VectorType">
  <xs:sequence>
    <xs:element name="vector" type="Tuple3DType"/>
  </xs:sequence>
</xs:complexType>

<!-- An offset i a direction -->
<xs:complexType name="OffsetType">
  <xs:sequence>
    <xs:element name="vector" type="Tuple3DType"/>
  </xs:sequence>
</xs:complexType>

<!-- A repeated offset -->
<xs:complexType name="OffsetChainType">
  <xs:sequence>
    <xs:element name="vector" type="Tuple3DType"/>
  </xs:sequence>
  <xs:attribute name="number" type="xs:positiveInteger" use="required"/>
</xs:complexType>

<!-- Coordinates for the pose of a shelf -->
<xs:element name="coordinates">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="origin">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="point" minOccurs="1" type="Tuple3DType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="axis" minOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="xAxis" type="VectorType"/>
            <xs:element name="yAxis" type="VectorType"/>
            <xs:element name="zAxis" type="VectorType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- A side of a shelf -->
<xs:element name="side">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="offset">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="alongEdge" type="OffsetType"/>
            <xs:element name="intoBox" type="OffsetType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="interDistance" type="OffsetChainType"/>
      <xs:element name="layers" type="OffsetChainType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- A shelf -->
<xs:element name="shelf">
```

B Application Usage

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="coordinates" maxOccurs="1"/>
    <xs:element name="boxes" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="side" minOccurs="1" maxOccurs="2"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="number" type="xs:positiveInteger" use="required"/>
</xs:complexType>
</xs:element>

<!-- The root element -->
<xs:element name="world">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="shelves">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="shelf" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

C Poses and Quaternions

Descriptions of how the positional information work in DWARF, especially how Quaternions can be used for rotation sequences

The main data structure for spatial information in DWARF comes from the CORBA structure *PoseData.idl* in the `DWARF` module of the `idl` source. While containing information about several tracking topics such as source, target, covariance and time, the main structures are the 3D position and a quaternion¹ to specify a point in space.

When building up AR applications the spatial relations between objects can be described as scene graphs in much the same way as it is well known from computer graphics[23] where spatial relations is represented as hierarcical trees. For applications using quaternions this means that quaternion rotation sequences might have to be merged when computing new positions in space, for example when wanting to compute the position just in front of the tracked HMD to add a floating item in the visualization.

As a hyper-complex number of rank 4, a quaternion can be represented as a real number with a vector part in R^3 and orthogonal basis (i, j, k) :

$$q = q_0 + \mathbf{q} = q_0 + q_1i + q_2j + q_3k$$

and with following definitions for the product between the components of the orthogonal basis

$$i^2 = j^2 = k^2 = -1$$

$$ij = k = -ji$$

$$jk = i = -kj$$

$$ki = j = -ik$$

the multiplication of quaternions becomes:

$$p = p_0 + p_1i + p_2j + p_3k \quad \text{and} \quad q = q_0 + q_1i + q_2j + q_3k$$

$$pq = p_0q_0 - \mathbf{p} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}$$

As with complex numbers of rank 2, the complex conjugate is defined as $q^* = p_0 - \mathbf{q}$ and with the norm $N(q) = \sqrt{q^*q}$ the inverse becomes

$$q^{-1}q = qq^{-1} = 1 \Leftrightarrow q^{-1}qq^* = q^* \Leftrightarrow q^{-1} = \frac{q^*}{N^2(q)} = \frac{q^*}{|q|^2}$$

¹Quaternions are hyper-complex number of rank 4. For a more indepth treatment see, for example, [16]

Also analog with complex numbers of rank 2, an angle $\theta \in]\pi, \pi]$ can now be associate with the unit quaternion²:

$$\begin{aligned} 1 &= q_0^2 + |\mathbf{q}|^2 \\ &= \cos^2\theta + \sin^2\theta \\ q_0^2 = \cos^2\theta &\text{ and } |\mathbf{q}|^2 = \sin^2\theta \end{aligned}$$

Finally a rotation operator $L_q(v)$ on vectors in R^3 can be defined through a quaternion q .

$$L_q(\mathbf{v}) = q\mathbf{v}q^*, \text{ with quaternion } \mathbf{v}_q = 0 + \mathbf{v}$$

which rotates the vector \mathbf{v} around the axis defined by \mathbf{q} with an angle of 2θ .

Coming back to the PoseData an example can be used as illustration of how spatial relations are computed. On figure C.1 a spatial relationship is shown where the point p 's positional data shall be computed in relation to the tracking coordinate system and where the rotation relations q_0 and q_1 are known together with the cooresponding translations t_0 and t_1 .

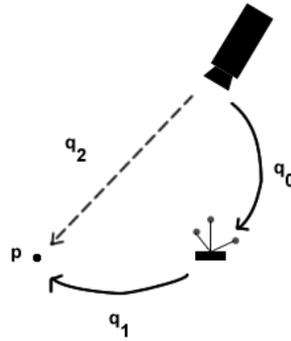


Figure C.1: Spatial relationship between a tracking system, a tracked object and a object where the spatial relation to the tracking frame, q_2 , is wanted.

Since a spatial relationship consist of both the postion t and rotation R they are often written as

$$(R_2|t_2)p = (R_0|t_0)(R_1|t_1)p$$

where $(R_1|t_1)$ brings the point into the coordinate system of the tracked object, and $(R_0|t_0)$ then brings it into the tracking coordinate system. With quaternions representing the rota-

²A unit quaternion has its norm equal to 1

tions, the equation then becomes (seeing the point p as a vector represented as a quaternion):

$$\begin{aligned}
 L_{q_2}(p) + t_2 &= (R_0|t_0)(R_1|t_1)p \\
 &= (R_0|t_0)(L_{q_1}(p) + t_1) \\
 &= L_{q_0}(L_{q_1}(p) + t_1) + t_0 \\
 &= L_{q_0}(q_1 p q_1^* + t_1) + t_0 \\
 &= q_0(q_1 p q_1^* + t_1)q_0^* + t_0 \\
 &= q_0 q_1 p q_1^* q_0^* + q_0 t_1 q_0^* + t_0 \\
 &= (q_0 q_1) p (q_0^* q_1^*) + q_0 t_1 q_0^* + t_0 \\
 &= q_2 p q_2^* + t_2, \text{ where } q_2 = q_0 q_1 \text{ and } t_2 = q_0 t_1 q_0^* + t_0
 \end{aligned}$$

Thus the positional data can easily be recomputed every time it is needed while only relying on the measurements of the tracked object. In AR applications this would typically be needed atleast one time per frame in the visualizations, but higher update rates are more common in an effort to minimize the *swimming-effect*.

D Glossary

Abbreviations and Term Definitions

AR. See *Augmented Reality*.

ART. Advanced Realtime Tracking. A tracking system based on cameras tracking markers with infrared light.

Augmented Reality A technique to mix the virtual and real world by registering the virtual objects *pose* with the real, and finally making it possible for the users to sense these augmented objects. Normally this is performed with a *HMD* where the virtual objects are overlaid the real.

CORBA. Common Object Request Broker Architecture. A communications layer for systems to share objects accross networks. It is used internally in *DWARF*.

DTD. Document Type Definition. A way to define the structure of *XML* files in a text based non-hierarchic structure.

DWARF. Distributed Wearable Augmented Reality Framework. A framework to create ad hoc networks of services specifically designed for *Augmented Reality*.

ForLog A project from the Bayerischer Forschungsverbund which research on how to make todays logistic systems more adaptive and flexible.

GUI Graphical User Interface.

HMD. See *Head Mounted Display*.

Head Mounted Display A display attached to the head of the user so to display small screens with information just in front of the users eyes.

IDL. Interface Definition Language. Is the standard API for calling *CORBA* services.

MitLog A sub-project of *ForLog* with the goal of making it possible for co-workers to deal with the high demands that working in a supra-adaptive logistic system will put on them.

OpenGL. Open Graphics Language. An API for applications to represent and draw 2D and 3D objects.

- OpenGL ES.** Open Graphics Language for Embedded Systems. A subset of the *OpenGL* API extended with fixed-point data types for systems without Floating Point Unit. Is intended to be used on small embedded systems and is promoted by Khronos Group[28].
- PDF.** Portable Document Format. A widely adopted standard for representing documents.
- Pose** 6 degrees of freedom to represent the position and orientation of an object.
- SPAAM** Single Point Active Alignment Method. A method for finding calibrations matrices for standard camera models.
- Supra-adaptivity** Is a definition from the *ForLog* project and defines a logistic system which has the ability to adapt to dynamic company-wide changes with minimal costs. This includes the abilities of change, integration and mobility in the physical, information oriented and structure sense.
- TCP** Transport Control Protocol. A standard protocol for transporting data over networks and make sure that all packets reach their destination.
- Tracker.** A device or system to determine the *pose* of an object.
- UDP.** User Datagram Protocol. A communications protocol without the guarantees of *TCP* to make sure that all packets arrive at the destination.
- UML.** Unified Modeling Language. A standard modelling language for specifying design of different aspects of computer programs and usage.
- VR.** See *Virtual Reality*.
- Virtual Reality** A technique to allow users to enter a completely artificial environment.
- XML.** Extensible Markup Language. A standard for storing data in a hierarchic way to allow it to be easily read by any system conforming to this standard. It is broadly used from specifying homepages to the grammar of programming languages.
- XSD.** XML Schema Definition. A way to define the structure of *XML* files and are in themselves also defined as *XML* files.

Bibliography

- [1] ARISTOTLE, *Posterior Analytics, book 2*, ~ 350BC. Translated by G. R. G. Mure, eBook at <http://etext.library.adelaide.edu.au/a/aristotle/a8poa/>.
- [2] R. AZUMA, *A Survey of Augmented Reality*, in *Teleoperators and Virtual Environments*, vol. 6, 1997, pp. 335–385.
- [3] J. BARRILLEAUX, *3D User Interfaces with Java3D*, Manning Publications Co., 2001.
- [4] O. BIMBER and R. RASKAR, *Spatial Augmented Reality, merging real and virtual worlds*, A K Peters LTD, 2005.
- [5] D. A. BOWMAN, E. KRUIJFF, J. JOSEPH J. LAVIOLA, and I. POUPYREV, *3D User Interfaces, Theory and Practice*, Addison-Wesley, 2005.
- [6] D. CURTIS, D. MIZELL, P. GRUENBAUM, and A. JANIN, *Several devils in the details: making an AR application work in the airplane factory*, in *IWAR '98: Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes*, R. Behringer, G. Klinker, and D. W. Mizell, eds., Natick, MA, USA, 1999, A. K. Peters, Ltd., pp. 47–60.
- [7] W. DANGELMAIER, B. MUECK, and W. FRANKE, *Mixed Reality in Lagerprozessen*, in *Augmented & Virtual Reality in der Produktentstehung*, J. Gausemeier and M. Grafe, eds., vol. 167 of HNI-Verlagsschriftenreihe, Paderborn, 9 - 10 June 2005, pp. 133–144.
- [8] W. DANGELMAIER, B. MUECK, M. HÖWER, and W. FRANKE, *AR Support for Picking*, in *Simulation und Visualisierung 2005*, T. Schulze, G. Horton, B. Preim, and S. Schlechtweg, eds., SCS European Publishing House, 3 - 4 Mar. 2005, pp. 355–362.
- [9] S. DONNEAU, A. FÜSSNER, K. LUTTER, and L. BIEVELEZ, *Zukunftstechnologien in der Logistik*, tech. rep., European Business School, International University Schloß Reichartshausen, 2006.
- [10] S. R. ELLIS, *Human Performance and Preference Studies*, in *IEEE Virtual Reality 2006*, 2006.
- [11] J. D. FOLEY, A. VAN DAM, S. K. FEINER, and J. F. HUGHES, *Computer Graphics Principle and Practice, second edition in C*, Addison-Wesley Systems Programming Series, Addison-Wesley Publishing Company, 1997.
- [12] W. FRANKE and B. MUECK, *Gibt es heute noch Innovationspotential in der Lagerlogistik*, in *Supply Chain Management in der Automobil- und Zulieferindustrie*, W. Dangelmaier, D. Kaschula, and J. Neumann, eds., vol. 12 of ALB-HNI-Verlagsschriftenreihe, April 2004, pp. 417–428.

BIBLIOGRAPHY

- [13] R. G. GOLLEDGE, ed., *Wayfinding Behavior, Cognitive Mapping and other Spatial Processes*, The Johns Hopkins University press, 1999.
- [14] T. GUDEHUS, *Logistik Grundlagen, Strategien, Anwendungen*, Springer-Verlag GmbH, 2005.
- [15] *ISO 9241-11 Guidance on usability*, in Ergonomic requirements for office work with visual display terminals (VDTs), no. 11 in 9241, ISO International Organization for Standardization, e ed., March 1998.
- [16] J. B. KUIPERS, *Quaternions and Rotation Sequences*, Princeton University Press, 1998.
- [17] A. MACWILLIAMS, *Using Ad Hoc Services for Mobile Augmented Reality Systems*, Master's thesis, Technische Universität München, 2001.
- [18] P. MILGRAM and H. COLQUHOUN, *A Taxonomy of Real and Virtual World Display Integration*, (1999).
- [19] M. MORRISON, *XML Unleashed*, Sams publishing, 2000.
- [20] B. MUECK, M. HÖWER, W. FRANKE, and W. DANGELMAIER, *Augmented Reality applications for Warehouse Logistics*, in Soft Computing as Transdisciplinary Science and Technology - Proceedings of the fourth IEEE International Workshop WSTST'05, A. Abraham, D. Yasuhiko, T. Furuhashi, M. Köppen, O. Azuma, and Y. Ohsawa, eds., Advances in Soft Computing, Springer-Verlag, 25 - 27 May 2005, pp. 1053–1062.
- [21] NIST/SEMATECH, *e-Handbook of Statistical Methods*.
<http://www.itl.nist.gov/div898/handbook/>.
- [22] S. P. SMITH and J. HART, *Evaluating Distributed Cognitive Resources for Wayfinding in a Desktop Virtual Environment*, in IEEE Symposium on 3D User Interfaces 2006, Alexandria, Virginia, USA, March 2006, IEEE, pp. 3–10.
- [23] H. SOWIZRAL, *Scene graphs in the new millennium*, Computer Graphics and Applications, IEEE, 20 (2000).
- [24] J. SWAN, S. R. ELLIS, and B. D. ADELSTEIN, *Conducting Human-Subject Experiments with Virtual and Augmented Reality*, in IEEE Virtual Reality 2006, 2006.
- [25] A. TANG, J. ZHOU, and C. OWEN, *Evaluation of Calibration Procedures for Optical See-Through Head-Mounted Displays*, ismar, (2003).
- [26] M. TUCERYAN, Y. GENÇ, and N. NAVAB, *Single-Point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality*, Presence: Teleoper. Virtual Environ., 11 (2002), pp. 259–276.
- [27] ForLog. <http://forlog/>.
- [28] OpenGL ES. <http://www.khronos.org/opengles/>.
- [29] *Questionnaire Design*. <http://www-static.cc.gatech.edu/classes/cs6751-97-winter/Topics/quest-design/>.

BIBLIOGRAPHY

- [30] *TyphoonLabs Real Time Technologies*. <http://www.typhoonlabs.com/>.
- [31] W3C. <http://www.w3.org/>.