SAARLAND UNIVERSITY

Faculty of Natural Sciences and Technology I
Department of Computer Science
Master's Thesis

DEPlaTa - A Digitally Enhanced Planning Table for Rough Factory Layouts

Nico Herbig
Master's Program in Computer Science
September 2015

Advisers
Frederic Kerber, German Research Center for Artificial Intelligence,
Saarbrücken, Germany

Pascal Lessel, German Research Center for Artificial Intelligence,
Saarbrücken, Germany

Supervisor
Prof. Dr. Antonio Krüger, German Research Center for Artificial Intelligence,
Saarbrücken, Germany

Reviewers
Prof. Dr. Antonio Krüger, German Research Center for Artificial Intelligence,
Saarbrücken, Germany

Prof. Dr. Peter Loos, German Research Center for Artificial Intelligence,
Saarbrücken, Germany

Submitted
30th of September, 2015

Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Statement in Lieu of an Oath:

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Saarbrücken, 30th of September, 2015

Declaration of Consent:

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, 30th of September, 2015

Acknowledgments

I sincerely thank Prof. Dr. Antonio Krüger for giving me the opportunity to work on this interesting topic and offering advice when needed. Furthermore, I would like to thank Prof. Dr. Peter Loos for reviewing my thesis. Also, I would like to express my eternal gratitude to my two thesis advisers Frederic Kerber and Pascal Lessel for their continuous support and invaluable feedback, especially during the writing phase.

Moreover, I thank my dear friends Kevin Salvesen and Matthias Hennemann for inspiring discussions and their indispensable help in proofreading. To my sister I say thank you for correcting my linguistic mistakes.

Lastly, I want to thank my friends and family who have supported me throughout the entire process.

Abstract

The layout of a factory is essential to cost- and time-efficient production in today's competitive manufacturing environment. This thesis provides assistance for the rough factory layout planning process, which is the initial design phase of factory work floors. While currently used pure analog approaches are easy to handle and support collaborative work, they miss the advantages of digital planning, namely being able to archive different design alternatives or to run simulations on created layouts. Since pure digital solutions are often complex to use and non-collaborative, planning experts in an initial requirements analysis stated that a system combining the advantages of analog and digital planning tools would be beneficial for rough factory layout design. We analyzed related planning approaches that try to combine the two worlds, however, none supported agile planning where new objects need to be created on the fly. Furthermore, they did not assist users in rebuilding real world physical models, thereby making it difficult to refine existing layouts. Motivated by these shortcomings, we present a tangible system allowing users to plan analogously with fast producible, arbitrarily shaped objects and colored adhesive tape in order to define rough factory layouts. The advantages of digital planning are taken by automatically creating a synchronized digital model of the physical 3D representation on a large planning table. The digital model can be exported in a standard format for archiving, running simulations or further planning. When importing an old state, the digital model is reloaded immediately and projections on the table help users rebuild the physical state. An evaluation at a large German manufacturing company showed that the automatic digitization was appreciated by the planning experts. The system saves the time needed for manual digitization and allows to easily test multiple design alternatives which facilitates creativity according to the participants.

Contents

List of Figures

List of Tables

List of Abbreviations

AR	Augmented Reality
CAD	Computer-Aided Design
CPU	Central Processing Unit
CSCW	Computer Supported Cooperative Work
DEPlaTa	Digitally Enhanced Planning Table for Rough Factory Layouts
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HSV	Hue Saturation Value
IR	Infrared
OMBB	Oriented Minimal Bounding Box
QR	Quick Response
RFID	Radio-Frequency Identification
RGB	Red Green Blue
SDK	Software Development Kit
TUI	Tangible User Interface
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WPF	Windows Presentation Foundation

Chapter 1
Introduction

This thesis presents **DEPlaTa** - a **D**igitally **E**nhanced **Pla**nning **Ta**ble for rough factory layouts. It facilitates rapid prototyping of a factory's interior layout by automatically creating a synchronized digital 3D model of objects and tape placed on a large planning table. Users can easily jump back to specific versions, add meaning to the digital representatives and export the created model to any 3D software (e.g., for running simulations). Thus, we argue that by using our system, collaboration is as easy as in purely analog planning without losing the advantages of digital planning.

1.1 Motivation

In the manufacturing industry, the factory layout has a significant impact on productivity, manufacturing costs and lead times [16]. Multiple optimization goals can be achieved by improving the arrangement of the needed machines, workbenches and supply areas. Not only the obvious objectives of improved productivity and decreased manufacturing costs need to be considered but also juridical and working atmosphere related goals such as noise protection or the locations of break rooms must be carefully planned. Thus, factory layout planning is a vital task to the survival of manufacturers in today's globally competitive environment [48].

During the past years the rapidly changing customer demand has led to shortened development and product life cycles [13, 48]. The manufacturing industry must therefore quickly adapt to current trends by restructuring production sites. At the same time the market's continuous pressure for cost reduction must be met. Consequently, there is a growing need for planning tools supporting manufacturers during the reorganization of old and the design of new production sites [41].

Currently, manufacturers are heavily investing in their production systems: according to a recent study the European industry will invest 140 billion Euro annually in so called *Industrie 4.0* solutions by 2020 [33]. *Industrie 4.0* describes the fourth industrial revolution, a term coined by the German government and industry. Figure 1.1 shows the four industrial revolutions on a time-line: the first industrial revolution started at the end of the 18th century with the mechanization of manufacturing equipment. Around 1870 mass production of goods became possible through the invention of electrical power, leading to the second industrial revolution. A century later, information technology allowed manufacturers to further automate manufacturing processes resulting in a third industrial revolution [30]. The goal of *Industrie 4.0* is to induce a fourth industrial revolution by creating a *Smart Factory*, where products are intelligent by knowing the required steps for their completion, workers are supported in all processes using modern technology and production and logistic systems are managed autonomously by *Cyber-Physical Systems* communicating over the Internet [30]. As the name suggests, *Cyber-Physical Systems* describe a tight coupling of computational and physical system components. They react to information acquired through sensors and can directly influence other modules of the system by communicating through a network [58].

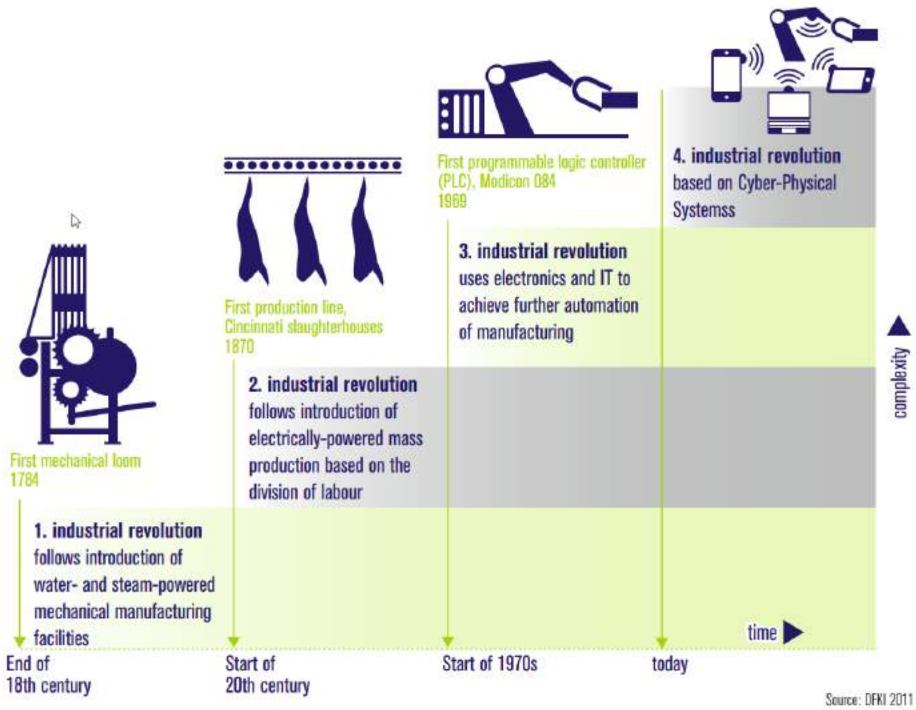


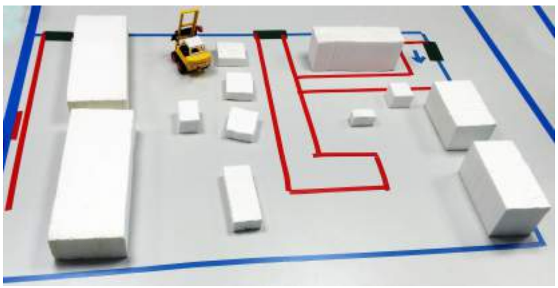
Figure 1.1: The four industrial revolutions [30].

In contrast to the first three industrial revolutions, increased productivity and resource efficiency should not only be achieved on the shop-floor level (the productive part of the factory, as opposed to the administrative area) in this new industrial era. Instead, a lot of improvement can be accomplished through better collaboration of brainwork and decision making processes such as factory layout planning [46]. A related result was proven in a recent survey, where 52% of the companies stated that investments in planning solutions have high priority. This is a similar percentage as assigned to other parts of the value chain such as product development, production and service; thus, making planning solutions one of the most important aspects of future manufacturing.

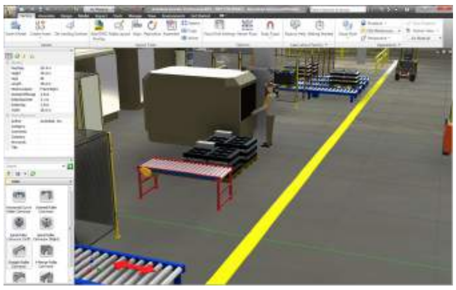
Usually many people with possibly different backgrounds are involved in planning factory layouts. Therefore, a system supporting the planning process must pay special attention to facilitate collaboration. A general term for technology supporting collaborative processes is *groupware* in the research field of *Computer Supported Cooperative Work* (CSCW) (see [40] for an overview). Using groupware for planning can lead to a closer connection of the planning experts in a project team [2]. Furthermore, such systems should allow simultaneous modeling, i.e. multiple users working on a layout at the same time without synchronization conflicts; a feature not supported by most conventional digital modeling tools [2]. Creating a groupware for factory layout planning therefore increases productivity from two perspectives: planning itself becomes more productive and through the support of the system the resulting shop-floor might improve as well.

1.2 Factory Layout Planning Today

There are multiple approaches for factory layout planning: e.g., companies can plan their facilities analogously using true-to-scale 2D or 3D representative objects which can be placed on a large board or table (see Figure 1.2a), or they can use Computer-Aided Design (CAD) software to precisely model the production site digitally and run automated simulations (see Figure 1.2b).



(a) Analog 3D planning.



(b) Digital planning in CAD software[1].

Figure 1.2: Two currently used approaches for factory layout planning.

[1]Software used: Autodesk Inventor.
Image taken from `http://www.solidsmack.com/wp-content/uploads/2010/07/autodesk-factory-design-03.jpg`. [last accessed 10/08/15]

The main advantages of analog planning are its simplicity of use and facilitation of collaboration. For rough prototyping planners can simply cut the representatives of real-world objects (e.g., machines) out of paper, Styrofoam or any other easily modifiable material and use cord, tape or markers to define pathways or walls. This straight-forward approach does not require long training phases and thus allows everyone to participate in the planning process. This is especially important as numerous stakeholders with different backgrounds (e.g., factory workers, production managers and architects) need to be consulted for satisfying results [48]. It is also very easy to communicate ideas since users can simply point to or touch the objects they are referring to and show alternative ideas by reordering the tangibles in place. When planning analogously with 3D models (e.g. using Styrofoam blocks), the physicality of the objects allows planners to intuitively grasp distance and height estimates. If available, precise 3D-printed objects can be used to increase the level of detail in 3D models.

However, analog planning also has many disadvantages compared to CAD planning: accurate cost and time analysis is extremely time consuming because no automatic simulations can be performed [23]. As pointed out by the participants of our evaluation (cf. Chapter 5), creating a digital model from an analog one by hand takes multiple hours even if no detailed objects are used. Therefore, users cannot quickly evaluate intermediate states with digital tools without interrupting the whole planning process and rescheduling with all involved stakeholders. In practice, often only the final state is digitized for further processing and archiving. Also recreating a state from photos can be cumbersome and imprecise depending on the complexity of the model. This hinders creativity because the users might be afraid to test alternative models if this implies erasing a current satisfying state. Table 1.1 summarizes the main advantages and drawbacks of purely analog planning and compares them to planning using CAD software which we discuss in the following.

When planning in the digital domain using CAD software, users can precisely model every part of a production site and run automated simulations on the digital representation [23]. Naturally, all intermediate states can be stored and reloaded if they turn out to be superior to the current one. Furthermore, digital planning also offers the possibility to send the created models to colleagues, allowing them to review and improve the proposed layouts.

	Analog Planning	**CAD Planning**
Advantages	easy to use for non-experts facilitates collaboration direct manipulation	precise automated simulations version control
Disadvantages	no simulations expensive manual digitization loss of old states	complex for non-experts requires training hinders communication

Table 1.1: Analog vs. digital planning using CAD software.

But again, these advantages come at a certain cost: due to the amount of features CAD software offers, it is very complex to use for non-experts. Therefore, less people can be involved in the planning process unless the different stakeholders undergo specific training before participating. Furthermore, CAD software is not well suited for collaboration: either participants have to sit in front of different computers, which hinders communication, or they all work together on a single machine, such that they effectively have to take turns to present their ideas or the idea must be described in enough detail that a single person handling the computer can model it. Additionally, the missing depth perspective on 2D screens makes it difficult to understand height relations between different machines.

Designing or restructuring shop-floors can be divided in different planning stages: during rough factory layout planning only the overall arrangements of the different machines, supply areas and pathways are determined. The resulting layout is then refined in more detail during fine planning, where the illumination within the factory, emergency exits and other fine granular aspects are considered [2]. The choice of the appropriate planning approach depends on the current planning stage. For example, CAD software focuses on detailed planning but is poor at representing the information critical at conceptual design [35].

In the remainder of this thesis, we focus on rough factory layout planning. Layout problems detected during this initial planning phase can be resolved considerably cheaper than during later planning stages [51]. Therefore, tools supporting users during this planning stage are especially important.

1.3 Requirements Analysis

As we have seen, both pure analog planning as well as digital planning using CAD software are far from perfect for creating rough factory layouts. To target the described problems, a requirements analysis was conducted within the *SmartF-IT*[2] research project [49], to determine which features a system supporting the planning process should provide. Several German manufacturers and research facilities are part of this large *Industrie 4.0* project, thereby connecting manufacturing know-how with IT-expertise to advance the fourth industrial revolution. Five male employees from one of the manufacturing companies (~800 employees) participated in an unstructured group interview lead by two SmartF-IT researchers. All participants have several years of experience in planning factory work floors for the production of domestic cooking appliances, with both analog and digital tools. To refine the initial findings a semi-structured interview with one of the planning experts was conducted afterwards. As the requirements gathered at a single company might not be generalizable to rough factory layout planning in all manufacturing areas, another interview with an employee of a different German manufacturer (~37,500 employees) was conducted.

[2]SmartF-IT project, sponsored by the German Federal Ministry of Education and Reseach (BMBF) under project number 01IS13015.

This interviewee focuses on planning production sites for large valves of agricultural machines, a fairly different domain than domestic cooking appliances. Based on these three interviews the following requirements were deduced:

R1 Simple Usage
Tools for rough factory layout planning should be very easy to use, even for non-experts. The time needed to learn how to handle the system should be as short as possible.

R2 Support Collaboration
The system should support collaborative work in groups, as many different stakeholders participate in the planning process.

R3 Physical True-to-scale Objects
To increase spatial awareness, the system should allow the modeling of shop-floors using true-to-scale physical objects. According to all interviewees such physical models also help in discussions with decision-makers, who were not involved in the whole planning process, as they are simple to understand and easy to use without prior knowledge.

R4 Support Agile Planning
The system should support agile planning processes, thus, it should not require long set-up or preparation phases. This especially means that the tangible representatives should be easily and quickly producible.

R5 Digital Model
The physical model should be backed by a digital model which can be used as a basis for simulations such as material flow or throughput time. This digital model should also be customizable with additional information. Further, it should adapt to changes of the arrangements of objects in the physical model; no manual effort should be required to synchronize the two models to ensure a continuous work-flow without interruptions.

R6 Version Control
The system should provide a version control mechanism allowing users to easily store and compare different drafts.

R7 Speech Input
To minimize the interaction with the system it should be possible to annotate physical objects and planning states via speech input.

1.4 Research Goals

This thesis investigates how rough factory layout planning can be supported by a system targeting the described problems of current planning approaches. It aims to fulfill all the requirements discussed above. Especially, the following main research goals will be covered:

Conceptual design of a rough factory layout planning system

A concept for a digitally enhanced planning table will be developed. As proposed by Dong and Kamat [14] the system will bridge the gap between the analog and digital worlds by combining the advantages of both. Chapter 2 will investigate which approaches for supporting factory layout planning already exist and to which extent they fulfill the above requirements. Chapter 3 will then present the concept of our system tackling the weaknesses of the related approaches. To the best of our knowledge, DEPlaTa will be the first system using easily producible arbitrarily shaped tangibles being backed by an automatically synchronized digital 3D model without time-intensive manual digitalization.

Implementation for the usage in a planning environment

A hardware prototype consisting of a depth camera and one or multiple projectors will be built, allowing to track the tangibles as well as rendering digital information back on the table. Special attention will be payed to the development of an algorithm creating clean 3D models with individual meshes per object from the noisy real world data. Furthermore, the possibility to interact with the system using a graphical user interface or speech input will be realized. The implementation details on all the features will be given in Chapter 4.

Investigation of the planning experts' interest in the system

A user study with five planning experts will be conducted to investigate the usefulness of the system in practice and analyze its strengths and weaknesses. It will be investigated which parts of the system need further refinement in following development cycles. The method and results of this evaluation will be presented in Chapter 5. Further research steps, which can be deduced from the study, will be outlined in Chapter 6.

Chapter 2
Related Work

This chapter provides an overview of the conceptually related works. Multiple systems supporting users in factory layout planning or other planning-related tasks are presented. To impose structure, these systems are ordered by their general type: purely tangible systems, tangible augmented reality approaches and finally virtual reality systems. At the end of this chapter, a comparison between the presented approaches and DEPlaTa is made.

2.1 Tangible Approaches

Tangible User Interfaces (TUIs) were introduced by Fitzmaurice et al. in 1995 [19] and were initially called *Graspable User Interfaces*. The idea was to control digital data by manipulating physical representatives. Two years later, this concept was renamed to *Tangible User Interface* by Ishii and Ullmer [26]. The coupling of virtual objects with physical objects increases spatial awareness, thereby making interaction more intuitive [19]. In the last 20 years, researchers addressed planning problems using TUIs: already the very first paper on this topic [19] proposed a very simple floor planner. Since then, many tangible planning tools have been developed; the ones most related to rough factory layout planning are presented in this section.

2.1.1 BUILD-IT

A system called *BUILD-IT* supporting the early design process of assembly line planning and building plants was provided by Rauterberg et al. in 1997 [43]. BUILD-IT consists of two working areas: a table augmented by a top-projection and a second projection to a wall. On the table, users can interact using a single tangible in the form of a brick. Machines can be selected from a machine store by placing the brick on their corresponding projection. After selection, the brick is linked to the machine such that it can be positioned and rotated by moving the brick or changing its orientation. All changes to the layout are directly projected on the table. To fix the layout and remove the coupling of brick and machine, the user must place his hand above the brick such that it is temporarily hidden from the camera tracking its shape. A virtual camera object exists which can also be positioned and oriented using the universal interaction handler. However, the camera object is not part of the designed plant layout. Instead, a rendering of the model from the cameras point of view is projected on the wall, thereby allowing a walk through the virtual factory. Furthermore, layouts can be stored and printed by placing the brick on corresponding menu entries. Figure 2.1 shows the setup of the system and interaction with the brick.

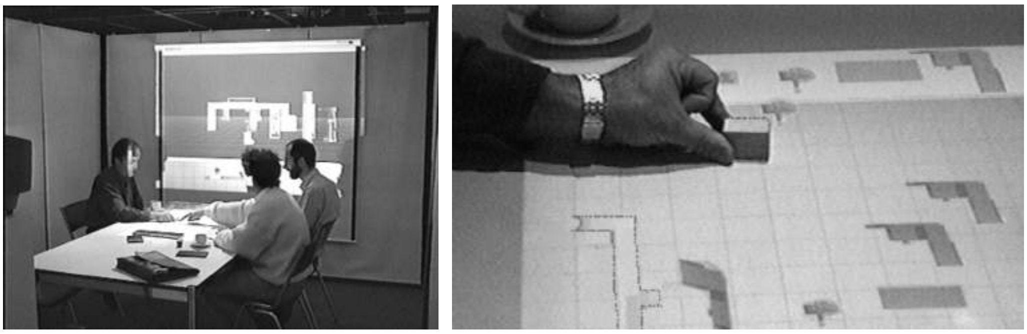


Figure 2.1: System setup and interaction on the table in BUILD-IT [43].

An evaluation with managers and engineers from companies producing assembly lines and plants was conducted. According to the participants, the system was intuitive and enjoyable to use and facilitates customer involvement. Since no knowledge of Computer-Aided Design (CAD) programs is required, it is easy to learn and allows non-experts to participate in the planning process. However, the system also has many drawbacks: there is only one interaction handler, therefore only one person can interact at a time and only one hand can be used. Furthermore, all used machines must be defined in CAD files to be imported into the machine store beforehand. Thus, it is not possible to quickly add new components during planning. While interacting with a brick might be more intuitive than using mouse and keyboard, it is obviously less intuitive than moving true-to-scale 3D tangibles. Furthermore, the image rendered on the table is a 2D view, therefore, it is not directly possible to grasp the height relations on the table.

Our approach, DEPlaTa, also consists of two working areas, namely the planning table and a regular Graphical User Interface (GUI) offering a 3D view of the objects on the table. To support agile planning, the system does not depend on a predefined set of objects but allows users to create new tangibles on the fly. Furthermore, physical 3D models are used to help planners estimate heights and support multiple interactions at the same time.

2.1.2 Urp: Urban Planning and Design

A tangible system for urban planning and design called *Urp* was developed by Underkoffler and Ishii in 1999 [55]. It allows arranging multiple predefined models of buildings on a luminous workbench to plan urban areas. The locations of the individual tangibles are tracked by analyzing the positions of colored dots attached to them in a video stream. Users can simulate the shadows cast by the buildings at any time of the day by setting the time in a special clock tangible. Furthermore, the reflections cast by buildings can be simulated by tipping a facade of a building with a special tangible stick. The computed shadows and solar reflections are then rendered directly on the table. The wind flow (with eight possible directions) can be simulated to check if areas exist where opening doors might be extremely hard. Furthermore, distances can be computed by touching two points with a special tangible, enabling users to check proximity constraints. Figure 2.2 shows the shadow and solar reflection simulations. On the right image, we can see a user interacting with the system, here, by touching a side of a model with a special tangible to define the facade as being made out of glass.

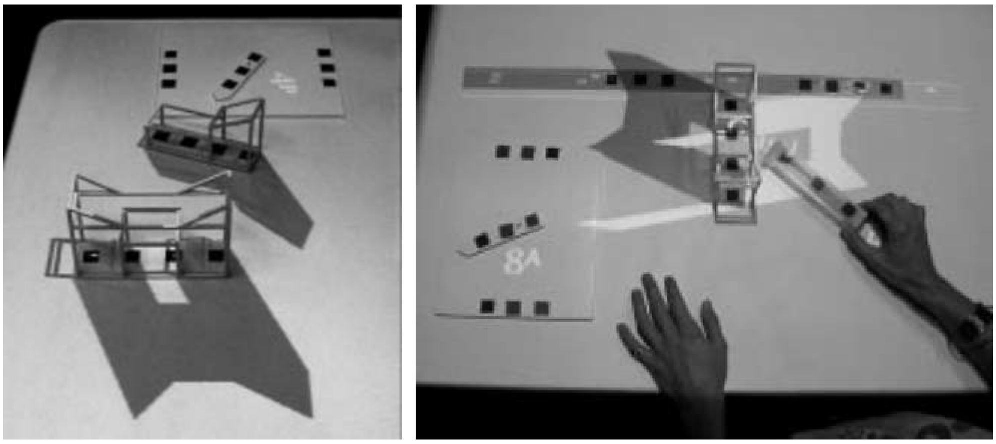


Figure 2.2: Shadows and solar reflections visualized by Urp [55].

The direct interaction in combination with the possibility to run simulations was appreciated by the architects and urban planners participating in a user study. They affirmed the usefulness of the system in client presentations and prototyping. Furthermore, about two hundred non-experts observed the system or interacted with it. The authors stated that the tangible approach apparently minimizes the "domain knowledge hurdle" through its simplicity of use, thereby allowing everyone to participate in the planning process. Still, the main disadvantage of the system is that it requires predefined and complex to build objects, which must be mapped to a 3D model in the system. Users cannot easily create new models on the fly, thereby hindering the creative planning process. Furthermore, it is not possible to save and load states to continue planning at a later point in time.

Analog to the presented system, DEPlaTa also supports planning with true-to-scale physical 3D models, however, it does not require a mapping to predefined digital models before starting the planning process. Instead of focusing on good support for a few simulations, we offer an export functionality in standard format, thereby allowing planners to run the simulations in an external program of their choice.

2.1.3 Luminous Table

Three years later, in 2002, an extension to Urp called *Luminous Table* was presented by Ishii et al. [24]. It allows to integrate multiple forms of physical and digital representations used during urban design. 2D drawings, 3D physical models and different kinds of digital models are overlaid in a single information space. Two video projectors and cameras located at the ceiling project dynamic digital simulations on the table and capture optical tags attached to the different representations. Apart from the integration of all kinds of media, the Luminous Table further extends Urp by improving the solar reflections simulation and adding support for traffic simulations. Furthermore, it can handle a more standard format of digital models than Urp and allows to save and load the state of the system. Another difference between the systems is that the Luminous Table uses a GUI as well as a TUI while Urp supported only tangible interaction. The reason for this transition is that the Luminous Table should also be used in real world applications where the table was often too crowded to place the extra tangibles properly. Figure 2.3 depicts a planning session using the presented system.

Figure 2.3: Combining different kinds of media for urban planning on the Luminous Table [24].

A study with eleven students of an urban design class was conducted after they worked with the system during a semester. While the participants again appreciated the physicality of the models, they still missed some flexibility, for example only one-way straight roads could be used. This missing support for agile planning is a drawback that any system depending only on predefined models has. Because of the system's simplicity, the potential to involve non-experts in the planning process was recognized by the participants. According to them, the Luminous Table facilitates collaboration because users can simply point to the physical objects when communicating. In contrast to usual computer interfaces, it also allows simultaneous work on a layout. However, participants also stated that the good support of the system regarding some urban planning problems exaggerated their importance compared to other unaddressed problems. Some participants even stated that the technology may have distracted from the actual work. This shows the risks of offering too much functionality. The Luminous Table theoretically supports automatic synchronization, however, the tracking algorithm often failed on large amounts of buildings such that users instead needed to synchronize manually.

The save and load functionality is also essential in rough factory layout planning because the planning process can last several weeks or months, therefore, DEPlaTa also supports this feature. Furthermore, we also use a hybrid GUI/TUI approach, because tasks like entering names are badly realizable using pure tangible interfaces. The evaluation shows that users require a lot of flexibility during planning, thus, DEPlaTa allows the use of quickly producible tangibles during planning. Furthermore, the study showed that offering good support for some simulations exaggerates their importance or even distracts from the actual task, therefore, we do not focus on simulations within our system but simply on offering export functionality which can be used to run simulations in expert software.

2.1.4 RFID Tangible Design Support System

Hosokawa et al. [22] present a tangible system supporting non-expert users in designing houses. Pre-built true-to-scale *tiles* and *plates* can be placed on a grid to create a miniature version of the rooms. Here, tiles are used to define the floor of the room and are made of the material they represent (e.g., wood or carpet). Plates are simply miniature wall pieces, possibly containing windows or doors of specific color, shape and material. By selecting and arranging the tiles and plates on a grid, users can define the layout of rooms, the positions, shapes and colors of windows and doors, as well as the materials and colors of floor and wall pieces. Using Radio-Frequency Identification (RFID) tags on the tangibles and a grid of RFID readers on the planning area, the authors implemented a system which automatically creates a 3D model of the miniature house. The 3D model is then rendered on a separate display and users can change the position and direction of view by moving and rotating a special camera tangible also equipped with a RFID tag and a six degree-of-freedom sensor. Figure 2.4 shows the planning area, a set of predefined tiles and plates and a rendering of the created 3D model.

Figure 2.4: Home design with the RFID Tangible Design Support System [22].

Since the tiles and plates are made of the corresponding real materials, users can not only see how their house will look, but also feel the different fabrics. By moving and rotating the camera tangible, it is possible to see how their layout would look from the perspective of a human walking through the house. Further, the resulting 3D model can be given to architects to understand the preference of their customers. In a user study, the participants stated that they enjoyed the direct manipulation and quickly understood the spatial implications of their changes. This shows that tangible systems are well-suited for planning systems.

However, the system also has some limitations: through the grid of RFID readers, only discrete positioning is supported. Furthermore, creating new tangibles is a lot of work since they must be physically built of the specific material, need to be modeled in 3D software with the corresponding textures and carry a RFID tag for the mapping of physical and virtual object. Thus, it is not possible to create new objects while planning. As only a limited amount of tangibles was offered to the participants of the study (e.g., only four different doors), they perceived the flexibility of the system as very limited. We deduce that RFID technology is a bad choice for rough factory layout planning, because the goal of versatility cannot be achieved if only predefined objects can be placed on a grid.

2.1.5 TinkerTable

Zufferey et al. [59] present a TUI for apprentices in logistics to increase their understanding of planning processes. In order to progressively acquire abstraction skills, two complementary interaction modalities are offered by the system called *TinkerTable*: a true-to-scale model with pre-built shelves defining a warehouse layout and paper-based forms called *TinkerSheets* to visualize data or control parameters used in simulations. Both are tagged with fiducial markers and can therefore be tracked and augmented by the TinkerTable as it has a camera and projector mounted above it. The simulation parameters are specified by placing tiny black disks on the TinkerSheets which are detected by computer vision algorithms.

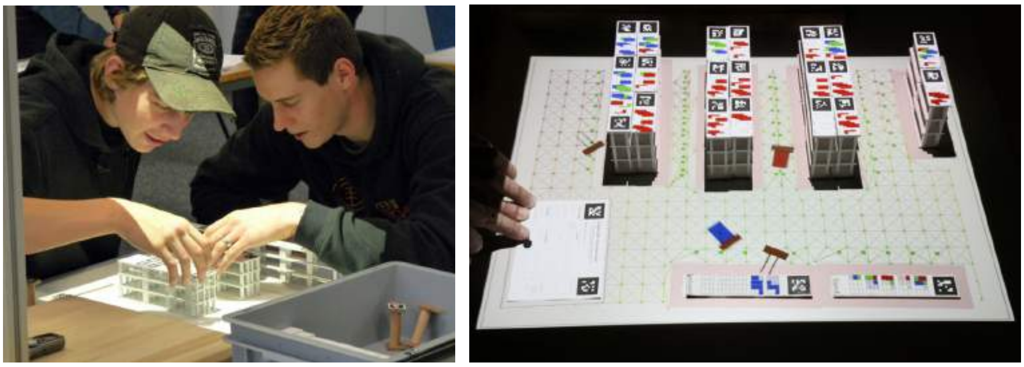


Figure 2.5: A TUI supporting apprentices in logistics called TinkerTable [59].

Using the system, apprentices can build warehouses by arranging the pre-built shelves on the table. Using the TinkerSheets, it is possible to run different kinds of simulations, for example determining the amount of accessible boxes in the shelves by specifying the type of the forklift and computing the efficiency of the layout. Apart from specifying parameters and starting simulations, TinkerSheets also hold additional information, such as the results of the simulations (e.g., visualized as graphs). Previously created layouts can be saved and reloaded using a dedicated TinkerSheet. Figure 2.5 shows apprentices during planning and a layout during simulation in which tiny forklifts are projected on the table. At the bottom left corner of the right image, we can see a TinkerSheet on which a parameter is specified through the black disk.

The system was evaluated in the classes of four teachers at two different schools. Multiple individual studies were conducted, specifically designed for participants in different stages of their apprenticeship. One task focused on defining a warehouse layout with the objective of maximizing the amount of accessible boxes by different types of forklifts. Another task was supposed to help apprentices understand the impact of a warehouse layout on work efficiency and teach them terms such as raw surface, raw storage surface and net storage surface. They could design a layout by placing the shelves on the table, see the computed net storage area and run a simulation on work efficiency. Generally, the evaluation shows that the tangible approach is well-suited for collaborative planning tasks as it made complex concepts easily understandable. This is important for rough factory layout planning as decision makers not directly involved in planning also need to understand planning states. Furthermore, the system seemed to facilitate collaboration as team members got quickly involved in discussions and apprentices were able to explain concepts to others. The physicality of the shelves helped apprentices understand the spatial relationships of a warehouse. While the TinkerTable seemed well suited for educational purposes and shows the possibilities that a similar system would allow for factory layout planning, it is not well-suited for that task by itself as it only offers a single size of tangibles, namely shelves.

2.1.6 Siemens IntuPlan

Siemens presents a tangible system called *IntuPlan* [44] (Intuitive Layout Planning), allowing users to place true-to-scale models of production and logistics components on a table. These models need to be available in advance, for example as 3D printed objects. Users then arrange the individual objects to create layouts optimizing e.g., the material flow. When a satisfying layout is found, it can be stored by taking a photo of it. By analyzing the positions of tiny markers attached to the topsides of the objects, a program developed by Siemens creates a virtual 3D model from the photographed scene. This 3D model can then be used for comparison with alternatives and for further processing such as running simulations. Until 2012, 15 factories around the world were designed using IntuPlan. The left image of Figure 2.6 visualizes how users plan production sites by simply moving the tangibles on the table. The right image shows a person taking a photo of the scene to be digitized and a 3D model created by the software projected on the wall behind him.

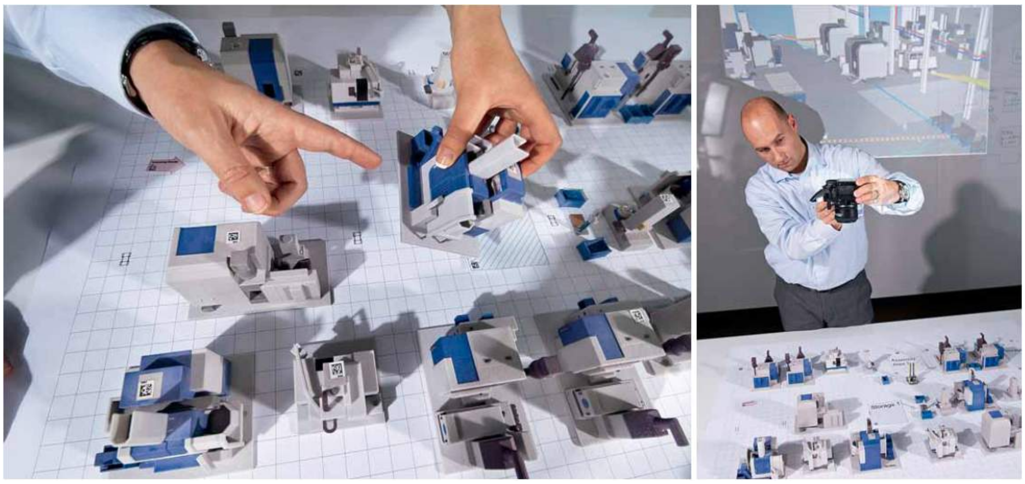


Figure 2.6: Planning production sites with Siemens IntuPlan [44].

Through the true-to-scale models and direct manipulation of the objects on the table, this approach is very simple to use, allows everyone to participate and facilitates collaboration. Using 3D printed objects generally is a good idea, as they increase the level of detail on the table, thus, we also want to offer users this possibility. However, as stated in the derived requirements, there should also be support for quickly created new components on the fly to facilitate creativity which is not possible with IntuPlan. Furthermore, the digital 3D model is only created when the user takes a photo, therefore, it does not adapt to changes in the real world automatically.

2.2 Tangible Augmented Reality

The concept of Tangible Augmented Reality was introduced by Billinghurst et al. [5]. It combines tangible approaches as described in the previous section with the concept of Augmented Reality (AR) where virtual objects are superimposed over real-world objects, typically viewed through hand-held or head-mounted displays. Thus, the advantages of TUIs, namely direct and intuitive manipulation are fused with those of Augmented Reality, specifically providing a spatially seamless display.

2.2.1 VOMAR

Billinghurst et al. [4] present four Tangible Augmented Reality prototypes, one of them being a virtual scene assembly application called *VOMAR*. Users wearing head-mounted displays see available objects superimposed on a real book. Using a cardboard paddle, they can pick up virtual objects by placing the paddle nearby. In order to arrange the object on the workspace defined by a large sheet of paper, users simply tilt the paddle such that the object slides off. The object can then be moved by pushing it with the paddle. Furthermore, it is possible to delete an object from the paddle by shaking it and from the workspace by hitting it. Technically, this is realized using computer vision algorithms and tracking the positions of the paddle, book and workspace through attached markers. In the background, all interactions are mapped to a simple CAD program. Figure 2.7 shows the overlaid book and workspace during selection of an object and placement in a room.

Figure 2.7: A tangible AR virtual scene assembly prototype called VOMAR [4].

The system visualizes objects in the same quality as CAD software but allows easier handling than such programs. Nevertheless, only predefined objects can be used as they must be first designed in a CAD program and added to the book, thus, it is not possible to define new objects during planning the way it is possible in DEPlaTa. Furthermore, the paddle as a universal interaction handler is an indirection compared to the direct manipulating of physical objects used in our approach and therefore less intuitive. Last, we do not want to require users to wear head-mounted displays during planning because they can hinder collaboration in shared environments [34].

2.2.2 AR Planner

Wang [56] presents another Augmented Reality system with a tangible interface called *AR Planner*, supporting users in planning construction worksites. Every user wears a head-mounted display and uses a paddle for tangible interaction with virtual objects. Analog to the previous approach, a set of predefined virtual 3D models exists from which the user can select different objects (see Figure 2.8 left). The interaction space is defined by a large sheet of paper filled with numerous fiducial markers (see Figure 2.8 right). When wearing the head-mounted display, this interaction space is overlaid with a rendering of the planned construction site. The user can select, place and manipulate the elements using a paddle as it was done in the previous work. However, the system offers more features than VOMAR such as validating planning results to improve the quality of the resulting plans. For example, collision detection was implemented even for moving elements such as trucks. Furthermore, adjacency constraints and safety margins can be defined, rendered as bounding boxes and evaluated at runtime. Apart from this, different simulations can be run such as throughput estimation, material flow or manpower requirements. Users can export the designed layout to Virtual Reality Modeling Language (VRML) format and re-open it in 3D-realtime rendering systems, external simulation programs or CAD software.

Figure 2.8: A tangible AR system for construction site planning called AR Planner [56].

While the paddle interaction might be easier to use than a real CAD program, it still misses the direct interaction that systems with real tangibles such as DEPlaTa can offer. Furthermore, the authors point out how problematic the paddle can be in crowded scenes, where it is hard to select a single object. Through the export functionality to standardized format, the created model can be used in further planning steps, a feature also supported by our system. As all previously presented systems, AR Planner also misses the main contribution of our planning system, namely the possibility to quickly create new objects instead of relying on a predefined set of elements. As with the previous system, we do not want to force our users to wear head-mounted displays as they can hinder collaboration [34].

In our opinion, the main advantage of tangible AR systems, namely the high quality of the visualization, is less important in rough layout planning than during fine planning. Since the approach has many drawbacks compared to purely tangible approaches as stated above, a system using head-mounted displays without physical tangibles is an unsuitable choice for rough factory layout planning. However, augmenting reality by projecting information on tangibles as done by Dalsgaard and Halskov [12], which can be seen as a less immersive form of Tangible AR, seems like an appropriate approach for rough factory layout planning. In their work, the tangibles could not only be textured on their top-side, but also from the different sides, a feature which could in theory be used to enhance the level of detail of the physical representatives (cf. Figure 2.9). However, the side augmentation is probably unfeasible in crowded scenes. Furthermore, they only supported projections on some predefined shapes, making it impractical for rough factory layout planning.

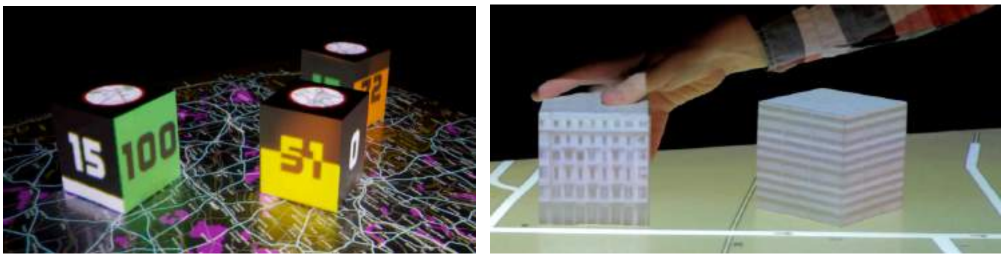


Figure 2.9: Projections on tangibles to achieve a higher level of detail [12].

2.3 Virtual Reality Approaches

Apart from purely tangible systems where users plan in the physical world and tangible AR systems where virtual objects are overlaid over real world elements, there is also the possibility to plan completely in the virtual world. While CAD software also allows planning in a virtual world, a better perspective and understanding of the created scenes can be achieved through fully immersive modeling environments [23]. The concept of applying Virtual Reality (VR) technology to manufacturing processes, called virtual manufacturing, was already introduced in 1995 [35]. Mujber et al. [37] state that through virtual manufacturing planning failures can be detected earlier which leads to a cost and time reduction. Furthermore, users can interact and change the environment at runtime and validate plans by simulation. This improves the users' understanding of the created layouts. The authors also argue that unskilled users can participate more easily in the planning process compared to design using CAD software as they are mapped inside the manufacturing site and can therefore immediately grasp the implications of their changes. Figure 2.10 shows a user wearing a head-mounted display and a virtual factory environment.

Figure 2.10: An immersive VR environment and a virtual factory used in virtual manufacturing [37].

Despite these advantages, VR systems only work if digital models of all shop-floor items are available since they need to be rendered. Through small time delays between the actual head movements of users and the detection and updating of the rendering in head-mounted displays, many users experience simulator sickness when using VR systems [32]. Furthermore, we argue that not seeing the other planners hinders communication in such a highly collaborative task as factory layout design. Taken together these drawbacks with the non-neglectable costs of VR systems, we think that tangible approaches are the better choice for our purpose.

2.4 Comparison to Our Approach

Table 2.1 compares the presented related planning systems with our approach which will be presented in detail in Chapter 3. It also summarizes the advantages and drawbacks of the individual approaches. The digital model was synchronized with the current planning states in all systems except Siemens IntuPlan [44], where users needed to take a photo to be analyzed by the software first. As we track the individual objects, DEPlaTa's planning state will always be backed by a synchronized digital model. The RFID Tangible Support System [22] did not allow continuous positioning because tracking was realized by a RFID grid on which tagged objects were placed. Being able to continuously position and rotate shop-floor elements is a crucial aspect of rough factory layout planning, therefore, we allow users to place the tangibles anywhere they prefer and track them using image-based methods (with a camera having sufficiently high resolution). In contrast to BUILD-IT [43], VOMAR [4] and AR Planner [56], DEPlaTa offers direct interaction with the tangibles without intermediate handlers like a paddle. Furthermore, our approach allows users to export layouts in a standardized format, enabling them to run simulations in external software and use the created model during fine planning. While many of the presented systems used a 3D renderer in the background and allowed such exports, the early systems Urp [55] and the

Luminous Table [24] did not offer this possibility. BUILD-IT [43], the Luminous Table [24] and the TinkerTable [59] have save and load functionality, however, their implementations differ: in BUILD-IT [43], the manufacturing system was defined by rearranging projections on a table, therefore, a simple load restored the whole planning state. In contrast, reloading a state with the Luminous Table [24] or the TinkerTable [59] only restored the state of the program, but offered no support to rebuild the physical model. DEPlaTa assists users in rebuilding stored states by projecting the positions of the tangibles on the table. Last and most important, none of the presented planning systems supported agile planning where users can quickly produce and use new tangibles. Instead, they relied on pre-built physical and digital models which were simply rearranged. In contrast, DEPlaTa allows users to create and place arbitrarily shaped objects on the table and automatically creates a digital 3D model, thus, supporting agile planning.

	BUILD-IT [43] '97	Urp [55] '99	Luminous Table [24] '02	RFID TSS [22] '08	TinkerTable [59] '09	IntuPlan [44] '12	VOMAR [4] '08	AR Planner [56] '07	DEPlaTa '15
Automatic Synchronization	✓	✓	∗	✓	✓	✗	✓	✓	✓
Continuous Positioning	✓	✓	✓	✗	✓	✓	✓	✓	✓
Direct Manipulation	✗	✓	✓	✓	✓	✓	✗	✗	✓
Export to Standard Format	✓	✗	✗	✓	✗	✓	∗∗	✓	✓
Restoring Old States	✓	✗	(✓)	✗	(✓)	✗	✗	?	✓
Using non-predefined Objects	✗	✗	✗	✗	✗	✗	✗	✗	✓

∗ = yes, but it did not work in bad lighting conditions
∗∗ = not stated explicitly, but should be possible
(✓) = reloading digital state, but no support for rebuilding physical state
? = not stated in the paper

Table 2.1: Comparison of the presented planning systems to our approach.

Chapter 3
System Overview

3.1 General Idea

This chapter gives an overview of DEPlaTa, the Digitally Enhanced Planning Table for rough factory layouts, which aims to fulfill the requirements gathered in the requirements analysis (cf. Section 1.3) by combining features of related planning systems (cf. Section 2.4) with new functionalities. The main idea of the system is to provide a seamless integration [25] of digital concepts into analog rough factory layout planning. Thus, the advantages of analog and digital planning as presented in Section 1.2 are combined without assimilating their drawbacks.

Traditional tables are well-suited as workspaces for many collaborative tasks such as planning, scheduling, design and layout [54]. Therefore, DEPlaTa supports analog planning with physical true-to-scale models on a large table. To integrate the advantages of digital planning into this analog tabletop planning approach, every physical object is backed by a digital counterpart. In contrast to the presented related works, no digital objects need to be modeled before the planning process; instead, they are automatically generated by DEPlaTa: the tangibles are 3D-scanned at runtime without notable additional effort for the users. Every translation and rotation of the physical objects is then mapped to the digital representative such that the analog and digital models are always synchronized. Such a Tangible User Interface (TUI) allows parallel input, leverages our well-developed skills for physical object manipulations and facilitates interaction through directness and multi-person collaborative use [19]. In contrast to CAD software which requires long training phases [18], all stakeholders can participate in the planning process independent of their computer literacy.

3.2 System Components and Features

This section discusses the concept of DEPlaTa in more detail by presenting the components and features of the system and arguing to which extent they fulfill the gathered requirements R1-R7 (cf. Section 1.3). We start by describing the set-up of the system and the analog planning process. Then the digital extensions and the possibilities to interact with DEPlaTa are presented.

3.2.1 Set-up

For designing a plant layout, the people involved in a planning session stand around a large table on which they arrange tangibles representing parts of the factory's interior. A depth-camera located at the ceiling is used to create a digital 3D model of the planning state. Depending on the properties of the room, a regular computer screen or a projection on a wall displays a Graphical User Interface (GUI), allowing users to see a rendering of the created 3D model and offering possibilities to enhance it with additional information. Furthermore, the table area and the tangibles on the table can be augmented with digital information by a top-mounted projector. Figure 3.1 depicts the tangible tabletop environment which builds the basis for good collaboration as demanded by requirement R2.

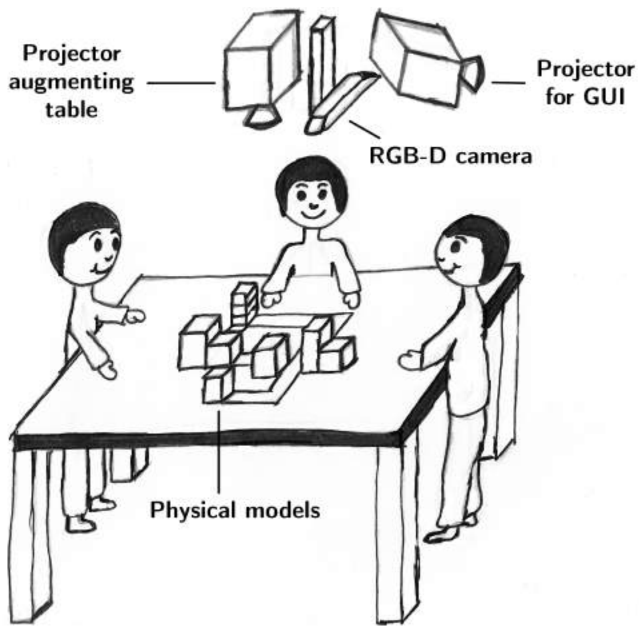


Figure 3.1: The setup of the system.

3.2.2 Tangibles

The tangibles on the table should be true-to-scale 3D objects as they increase spatial awareness (R3). To support agile planning (R4), where new objects need to be created on the fly, the physical representatives of machines, shelves or workbenches ought to be quickly producible. While DEPlaTa can theoretically deal with any solid material, we used Styrofoam for our experiments because it is cheap and easily manipulable with a hot-wire cutter or a knife (see Figure 3.2a). Furthermore, the system does not require digital 3D models corresponding to the physically cut objects which reduces the time needed for set-up and preparation phases (R4). Another advantage of Styrofoam is that its white surface offers a good contrast when projecting on the tangibles. The tangibles allow multiple users to modify the arrangement at the same time, thereby supporting collaborative group work (R2). Due to the simplicity of creating and arranging physical models, even users who are not familiar with DEPlaTa can participate in the planning process, thus, fulfilling requirement R1.

3.2.3 Tape

Apart from Styrofoam which is used for the objects within a plant, users can also define the overall layout (e.g., walls, pillars, doors) and paths throughout the factory. All they need to do is to place adhesive tape (see Figure 3.2b) at the specific positions on the table, which is extremely simple (R1) and can be done by multiple users in parallel (R2). Different colors can be used for different purposes, for example blue for the layout of the building and red, green and yellow for different types of paths depending on the vehicles that can use them. Since the tape is adhesive, it does not accidentally shift when rearranging the tangibles. Nevertheless, it can be rapidly placed and removed and is very cheap.

(a) Styrofoam for the tangibles. (b) Tape for the building's layout and paths.

Figure 3.2: Styrofoam and tape can be used to define machines, workbenches, pathways or material flow.

3.2.4 Digital Model

As previously stated, the physical model is backed by an automatically created digital model which can be used as a basis for further planning steps, e.g., simulations such as material flow or throughput time (R5). The digital model reflects all parts of the analog model, namely the tape and the tangibles. It is automatically created in the background from the data acquired by the camera. The only additional step compared to purely analog planning is that users have to stick optical markers encoding IDs on the tangibles for technical reasons (cf. Figure 3.3). However, since these markers can be pre-printed on adhesive paper, the additional effort of attaching them is negligible. As we also print the ID next to the markers in human readable form, the attachment of markers also enables users to reference objects by their IDs in discussions. After attaching the markers, no user interaction is required for the recognition process since it runs completely in the background to ensure a continuous work flow (R5). The objects are scanned once when they are initially placed. Afterwards, when a user moves a tangible, the movement and rotation of its marker is detected and used to apply the same transformation to its digital counterpart. Thus, the digital model is automatically synchronized with the physical model (R5).

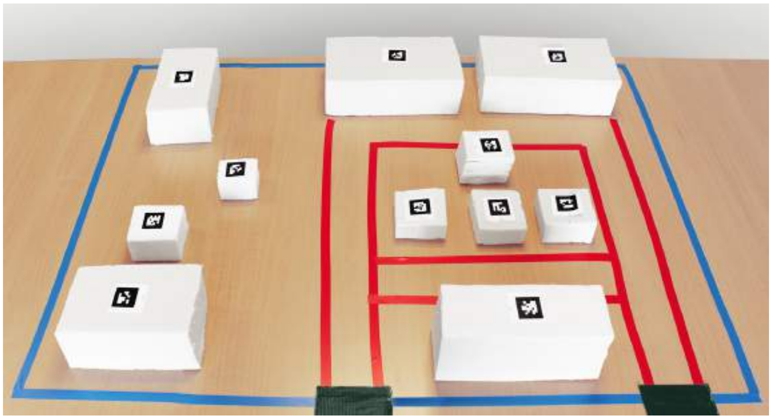


Figure 3.3: Physical planning with DEPlaTa.

Cutting Styrofoam precisely can be difficult and should not be required for rough factory layout planning. As an example, edges are often cut more crookedly than intended and objects that are supposed to be cuboids usually do not have perfectly parallel sides. However, these errors should not be reflected in the digital model. Therefore, DEPlaTa offers a mechanism to automatically correct the digital shape accordingly such that the digital model is cleaned up. The recognition process with all underlying algorithms is described in detail in Section 4.2. It is robust to user interaction during the scanning process, for example, it can deal with translations and rotations before the model is built completely or notices when a tangible is still held in a user's hand.

3.2.5 Digital Enhancement

To add meaning to the created digital model, it can be enhanced by assigning names and annotations to the whole model or to individual parts of it (R5). For example, the name of the model might simply state the hall that is being reorganized while the description might contain the advantages and drawbacks of that layout or the persons involved in the planning process.

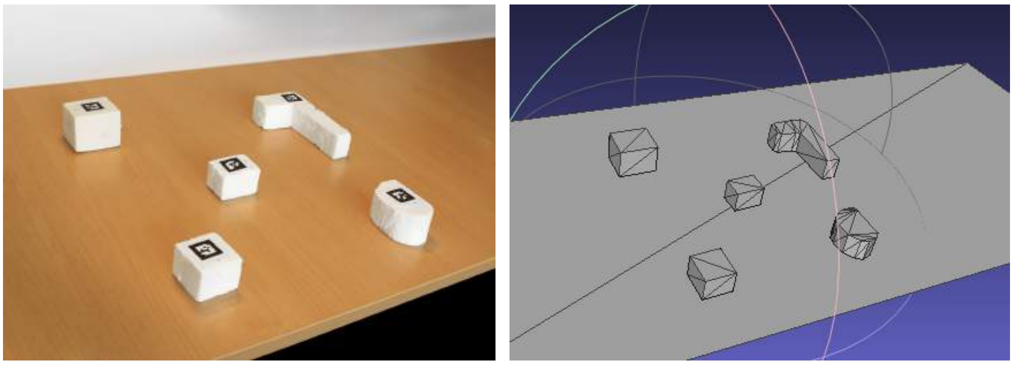
However, using only names and annotations is not practical when multiple instances of the same machine exist within a factory because the user would have to enter the information for each instance separately which is time-consuming. We therefore offer a layered approach, where conceptual ideas are detached from the actual representations. Apart from the created 3D objects, an abstract concept layer representing all characteristics which are not directly defined by the physical shape is introduced. A concept can either be a single property of the represented real-world object or a prototype defining complete elements of the shop-floor. For example, a property could be the energy consumption of a machine or the groups of people that are allowed to use it. Thus, properties can exist in parameterized form (water-consumption and the specific amount) or in unparameterized form (master craftsman). To help users find a specific property within a possibly large set, DEPlaTa supports grouping. As an example, the different forms of energy consumption (e.g., water or electricity) could be grouped to impose structure. A prototype is now simply a named collection of property instances (e.g., a "100t press" which can be used by "master craftsmen" and "team leaders"). Both, individual properties or prototypes can be attached to the digital representatives to add meaning to the objects.

Once created, concepts are permanently stored and can be reused over multiple planning sessions. Existing prototypes and properties can then be assigned to objects with a single markup step. The same prototype can be attached to multiple representatives and thereby mark them as being of the same type. However, machines of the same type might not be completely equivalent. Therefore, it is possible to further specify this object by adding and deleting properties or adapting the values of parameterized properties after a prototype was assigned to an object.

3.2.6 Version Control

During planning, many possible layouts are created, digitally enhanced and examined. When planning purely analogously, the old planning state is lost whenever the users rearrange the tangibles. Due to the amount of considered layouts, it is not unlikely that an intermediate state turns out to be the best. However, the possibilities to store states in purely analog planning are very limited: users can take a photo or manually create a digital model. In the former, it is difficult to exactly recreate a state while the latter can be very time consuming. To ensure a continuous work flow, planners might therefore choose to digitize only the final state of the planning process such that it can be used for simulations (e.g., material flow or throughput time) and for fine layout planning.

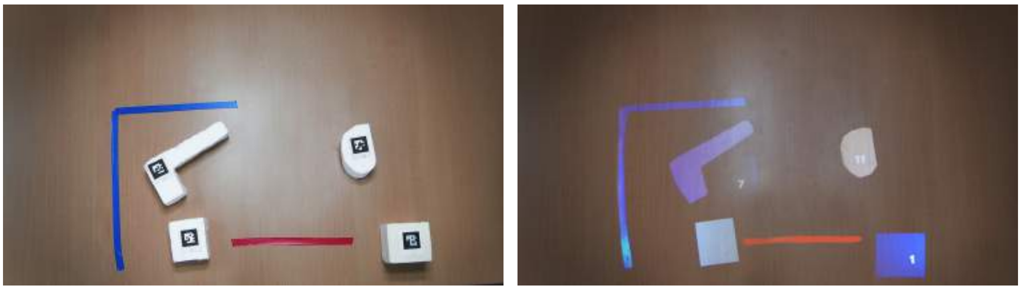
Since DEPlaTa automatically creates a 3D model, it can support the planning process by offering store and load functionality (R6). Users can save intermediate states at any time and continue planning later on. To satisfy the aforementioned need for a digital model that can be used for simulations and for fine layout planning, DEPlaTa also offers an export functionality to a standard format. We decided to use the Wavefront .obj format as it can be read by almost any 3D application. Apart from manually storing a layout, DEPlaTa also automatically saves planning states in the background. Otherwise, layouts could be lost because users forget to store them or because at the time of planning, a layout did not look promising. Instead of trying to recreate the lost state from memory, users can then simply reload it. Figure 3.4 shows a physical layout and its exported model visualized in external 3D software.



(a) Physical model. (b) Exported digital model.

Figure 3.4: An analog model on the table and the created digital model opened in MeshLab [10].

When importing, the old digital state of the system is immediately recreated and the digital model is rendered in the GUI. In contrast to the related works which also use true-to-scale physical models, DEPlaTa helps users rebuilding those physical states. This is achieved by projecting the 2D shapes of the objects and the locations of colored tape at the correct positions on the table (cf. Figure 3.5). All there is left to do for the users is to place the tape and objects on those rendered positions. When an object is placed correctly, its 2D rendering is switched off to help the user find the correct spot. This feature can also be used to compare an old state to the current physically planned state (R6): by reloading the old state, it is rendered on top of the current layout, thereby visualizing the differences. Since the layouts are stored in standard format, they can also be compared in more detail in external 3D software where simulations can be run to quantify the quality of the layouts. Furthermore, the export and import functionality can be used to continue planning at completely different locations: for example, a user can store the state at location A and send the exported files via email to location B, where users either create the tangibles by cutting them out of Styrofoam or by using a 3D-printer to directly print the sent files. The only thing that remains to be done is placing the created objects on their projection at location B and continue designing the plant there.



(a) Physical model. (b) Projection for rebuilding.

Figure 3.5: A physical model and a projection helping users to rebuild the state.

3.2.7 Graphical User Interface

Figure 3.6 shows DEPlaTa's Graphical User Interface with overlays for the different sections. A larger version of the image can be found in Appendix A. The middle area shows a rendering of the created digital 3D model while the right area shows the individual components of the model in a list-view and offers the possibility to digitally enhance them (cf. Section 3.2.4). The tape is rendered in the tracked color by default, however, the user can change the displayed color at any time. Since tape can also be used for the outline of the building, users can define the height of areas covered by tape and thereby create walls in the digital model. All individual objects have the ID encoded by their markers rendered on top of them to help users with the mapping between the analog and digital model. Each object has a random color assigned to it which is used for rendering and shown in the list-view, thus, reinforcing this mapping. The list-view is divided into the model itself with name and description, a section for the objects and a section for the tape. In the object section, users can edit the names, descriptions, properties and prototypes of the objects. The tape section allows to define the displayed color and height. Already the small artificial example of Figure 3.6 uses a lot of the available space in the list-view. To make the GUI practical even for large layouts, all parts of the list-view can be expanded and collapsed to show only the necessary information for the users. Since it might be hard to a find an object in a long list, it is also possible to expand and select an object by simply clicking on the digital representative in the rendering.

The lower left area of the GUI allows to create, edit and delete properties and prototypes as explained in Section 3.2.5. The properties can be encapsulated in groups which can again be expanded and collapsed to impose structure. When creating a prototype, the user can enter a name, select a color and the set of properties. Furthermore, values can be assigned to the parameterized properties (e.g. water-consumption gets the value "30 l/min"). To assign prototypes and properties to individual objects, users can simply drag them from their corresponding list and drop them on the respective object in the list-view. Alternatively, they can edit an object in the list-view by clicking on a button and select a prototype or

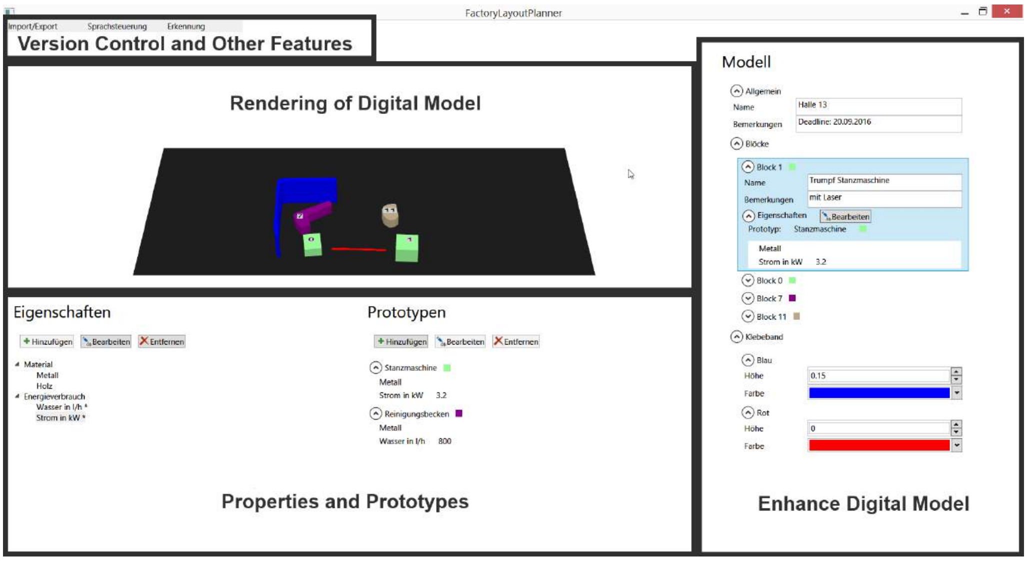


Figure 3.6: DEPlaTa's Graphical User Interface with overlays for the different sections.

property from a drop-down menu. In this editing mode for individual objects, it is also possible to specialize an object with an attached prototype by adapting the selection of properties and their assigned values. When a prototype is assigned to an object, its digital model is rendered in the color of that prototype. Thus, users can immediately see when multiple objects represent the same type of machine.

Apart from using the GUI to add meaning to objects, users can also use voice commands (cf. Section 3.2.8). The menu bar at the top of the window allows users to enable and disable the speech recognition and input frequently used words to increase the recognition quality. Furthermore, the menu offers functionality to start and pause the object recognition process, manually export and import digital planning states and define where models should be automatically stored to and the time interval at which the storing process should be triggered (cf. Section 3.2.6).

3.2.8 Speech Input

The whole process of creating and synchronizing the digital model is performed in the background. As we want to reduce interactions with the GUI to a minimum, users also have the possibility to enhance the digital model via speech input (R7). Names and annotations can be assigned to the whole model or individual parts of it. Furthermore, properties and prototypes can be added to the individual representatives. To specify the object being enhanced, users can simply state the ID which is printed next to the marker.

A custom grammar that is dynamically updated is used to improve the recognition quality and offers a lot of flexibility. Instead of allowing only a single sentence structure to perform a task, multiple variations are allowed. For example, users can say "annotate", "comment", "note", "remark" or many other options to add an annotation to an object (which they might call "block", "object", "device" or "machine"). At the same time, the grammar limits the recognized statements to those containing objects being physically on the table and to prototypes that are actually stored in the database. For assigning labels and descriptions to individual objects or the whole model, arbitrary inputs are allowed since we use a dictation grammar. However, some domain-specific words are wrongly recognized when dictating text for names or annotations. Therefore, the GUI offers the possibility to insert words which are added to the custom grammar and thereby increase the recognition quality.

Chapter 4
Implementation

This chapter shows how DEPlaTa was implemented. The software is divided into two individual components (cf. Figure 4.1). The model recognition component is responsible for creating and updating a digital model from the physical planning state on the table. To recognize the Styrofoam objects and the colored adhesive tape, shapes must be extracted, noise in the acquired data needs to be reduced, 3D meshes must be generated and changes to the physical model must be detected and reflected in the digital model. Since these tasks should be solved in a very performant matter, we decided to implement the model creation process in C++. The model interaction component then uses the created digital model and renders it in the GUI where users can also enhance it digitally as described in the previous chapter. Furthermore, the speech recognition, version control and renderings for the projector are part of this component. We chose to implement this component in C# using Windows Presentation Foundation (WPF) because it offers great support for creating GUIs. The communication between the two modules is realized by a socket connection.

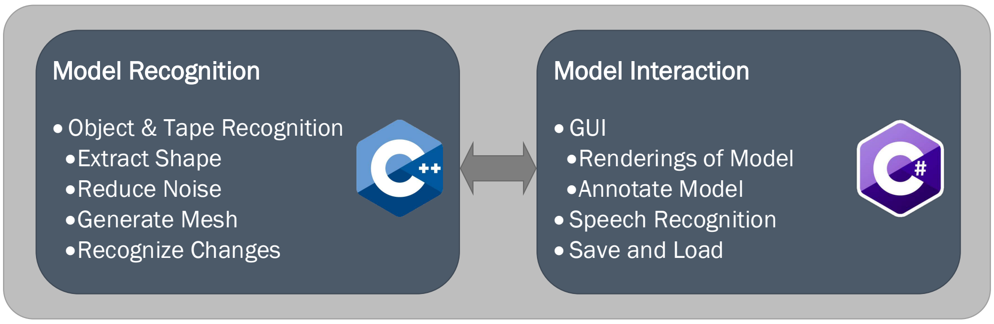
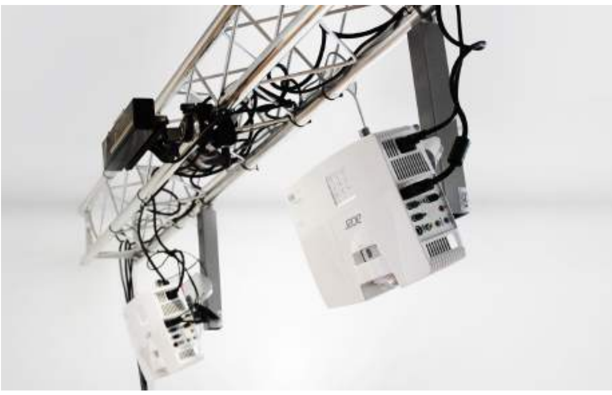


Figure 4.1: System components[3].

The remainder of this chapter consists of three parts: first, the created hardware prototype is shown. Then the algorithm used for object and tape recognition is presented and finally the implementation of the model interaction component is explained.

4.1 Hardware Prototype

A hardware prototype consisting of a large standing table and a computer monitor with mouse and keyboard was built. Above the table a truss carrying a Microsoft Kinect v2 and two projectors is attached to the ceiling (cf. Figure 4.2a). The low-cost Kinect v2 is used because it offers color (RGB) data for the colored tape recognition, depth data for object recognition and a microphone array for speech recognition (cf. Figure 4.2b). The depth data is acquired by emitting infrared (IR) light and measuring the time it takes until it is reflected by a surface and received by an IR sensor inside the Kinect. Since the distance between the projectors and the table surface is relatively small (143 cm), we used short-throw projectors. We do not require a high resolution for our setting, therefore, we chose the cheap Acer S1283e. The Kinect can track an area of 200 cm by 80 cm at this distance. Due to this length, two projectors are required to cover the whole space. Additionally, a table and a modern computer with a screen, mouse and keyboard are required.



(a) System setup.

(b) Microsoft Kinect v2[4].

Figure 4.2: DEPlaTa's apparatus and the Kinect v2 sensor.

[3]Logos taken from:
`https://cmsresources.windowsphone.com/devcenter/common/resources/images/games/tech/CPlusPlus.png`
`https://cmsresources.windowsphone.com/devcenter/common/resources/images/games/tech/csharp.png` [both last accessed 06/09/15].

[4]Image taken from:
`http://image.slidesharecdn.com/kinectv2introductionandtutorial-141114042655-conversion-gate01/95/kinect-v2-introduction-and-tutorial-6-638.jpg%3Fcb%3D1415940331` [last accessed 06/09/15].

4.2 Model Recognition

This section explains how a physical planning state is digitized. First, we describe how objects are recognized from the Kinect's depth data and then we show the process of recognizing colored tape from the RGB stream.

4.2.1 Object Recognition

Figure 4.3 gives an overview on the object recognition process. Whenever an object with a new marker is placed on the table, the process starts by acquiring the heightmap of the KinectFusion [27] algorithm as it is more consistent than the raw depth data of the sensor. Since an individual 2.5D (2D shape and height information) model per object is required, we need to separate the objects. This is done by first removing all points at table height and then using image processing and computer vision algorithms to extract the individual shapes. Afterwards, noise in the detected shape is reduced and the height of the object is computed. Finally, a mesh is created from the shape and height information. In the end of the section, we show how the algorithm deals with noise and users interacting with the tangibles while the model is initially created and describe how the created model adapts to changes in the physical world.

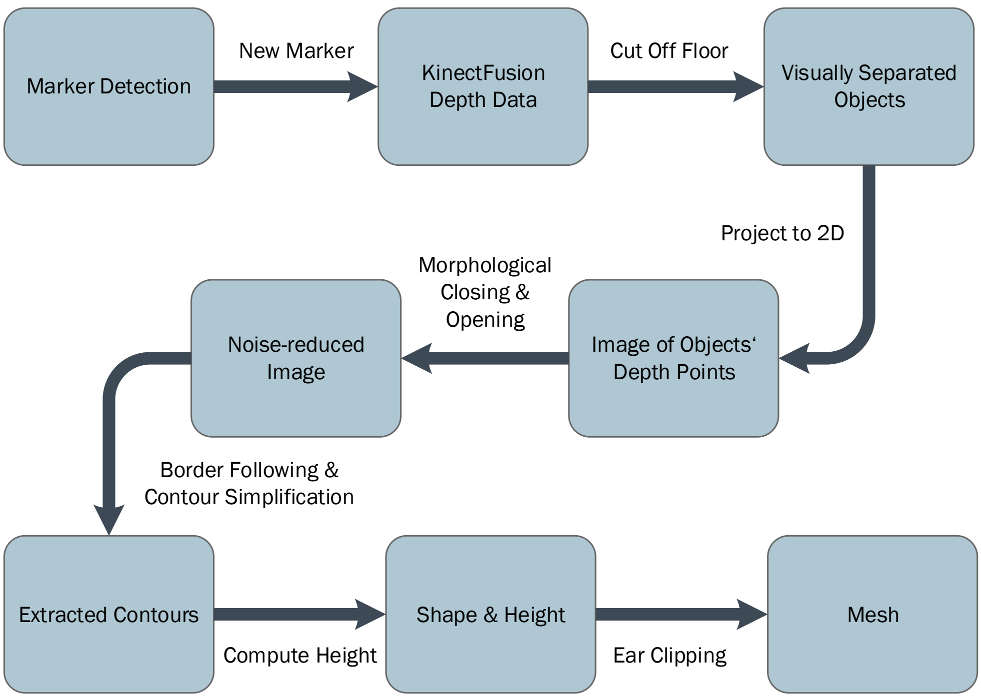
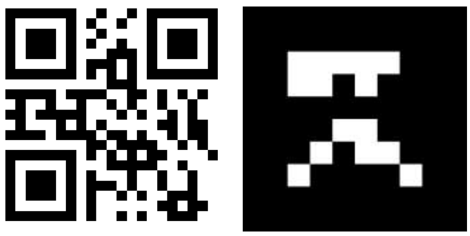


Figure 4.3: Overview of the object recognition process.

4.2.1.1 Marker Detection

Each object is tagged with an optical marker which can easily be detected from the camera's color stream. There are multiple reasons for the markers: first, each object only needs to be scanned once, since we store each digital model together with its marker during the creation and later simply map all translations and rotations of the marker to the digital object. This saves computation time and thereby makes the system more responsive. Second, a newly detected marker implies that a previously unseen object is placed on the table, thus, we know when the scanning process needs to be triggered. Last, markers allow to easily get separated digital models even if two objects are placed directly next to each other. If we created a new digital model all the time, distinguishing a single large object from several small ones in the heightmap would be very difficult. By requiring users to initially place each object on its own, we get separated digital models even if they are moved next to each other afterwards.

We use fiducial markers instead of commonly known Quick Response (QR) codes (cf. Figure 4.4), because they are better suited for our approach: QR codes focus on encoding a lot of data, whereas the goal of fiducial markers is to be easily recognizable [17]. The drawback of fiducial markers is that they encode only little data, however, a simple number is sufficient for our identification purpose.



(a) QR Code. (b) Fiducial Marker.

Figure 4.4: A QR code in comparison to a Chilitag fiducial marker [6].

We decided to use the Chilitags 2 library [6] for multiple reasons: in contrast to most libraries, it returns not only the position, but also the orientation of the markers which we need to recognize rotations of the objects. Furthermore, tags can be recognized even if they cover only 20 pixels on the image [6], thus, allowing the use of relatively small markers at the distance between the Kinect and the table. The library was developed for a setup where a camera records a table which is augmented by projections [7], therefore, the marker detection is robust to rapidly changing illumination. In this setting, Chilitags 2 also proved to be very precise, efficient and reliable over a longer period of time which suits our scenario well. An image showing the Kinect's whole field of view with the detected markers overlayed can be seen in Figure 4.5.

To reduce CPU usage, markers are analyzed every 50 ms, which is frequent enough to let the system feel interactive. By comparing the newly detected markers with the lastly analyzed markers, the system checks whether any markers

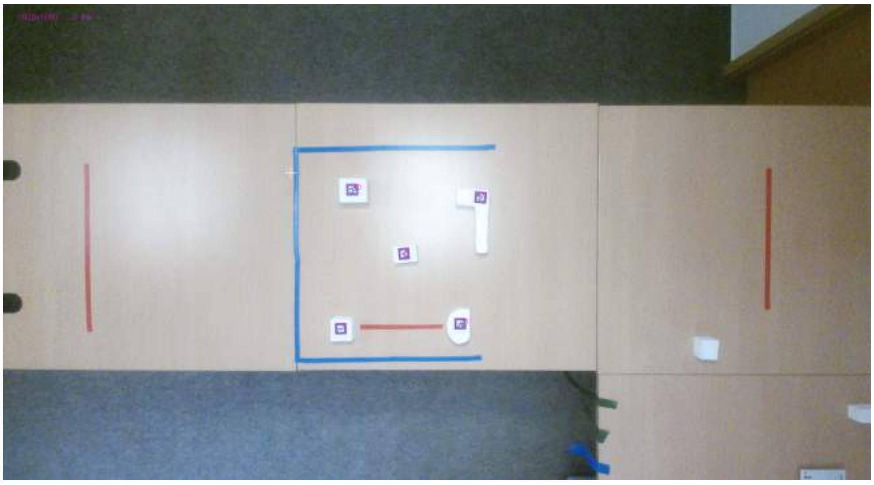


Figure 4.5: Marker detection using the Chilitags 2 library [6].

were added or removed or if any marker was moved by more than 3 pixels. The model is only adapted if one of these conditions is fulfilled. Since the Kinect's RGB camera automatically chooses its exposure and tends to overexpose images, the fiducial markers often look like black squares with white interior. We therefore put darkening foil in front of the camera which prevents such overexposures.

The fewer pixels on the image are covered by a marker, the more often markers remain undetected. Those temporarily failed recognitions result in flickering in the created model because every time the marker is removed, the digital model of the corresponding object is also removed. With a fixed camera, only two approaches exist to prevent these flickering effects: the marker size can be increased or the markers can be cached. Bigger markers can be problematic since they define the minimal required object size. Caching the markers, i.e. removing them only if they are not detected for several successive frames, leads to a delay until we recognize that an object was actually removed. Using 3.25 by 3.25 cm sized markers and caching for 9 frames turned out to be good compromise between the two for our setting.

4.2.1.2 KinectFusion Depth Data

When a new marker is detected, we know that a previously unseen object was placed on the table. To create a 3D model of the object, depth data is required. However, the raw depth map returned by the Kinect sensor usually has many holes at locations where it was unable to measure depth. There are several approaches to solve this problem: Piumsomboon et al. [42] use OpenCV's[5] inpainting method which guesses the missing points from the correctly captured data in the neighborhoods of the holes (spatial smoothing). Wilson [57] also uses spatial smoothing, however, he also integrates over time to fill up holes with

[5] `www.opencv.org` [last accessed 05/09/15].

correctly captured data from the past (temporal smoothing). In contrast to spatial smoothing, temporal smoothing does not guess missing values and is therefore unbiased. However, the integration over time introduces a delay until changes are recognized. Since small delays of a couple of frames are irrelevant for our scenario, we use a temporal smoothing approach. Specifically, we decided to use KinectFusion [27, 28, 38] since it was specifically designed for the Kinect sensor and returns a more consistent and less noisy depth measurement than the live data.

Even though KinectFusion was made for fusing frames from different perspectives, the integration over time without moving the sensor also yields a higher quality depth map than the raw data. Since the algorithm is also stable in different indoor lighting conditions and can deal with changing dynamic scenes, it can also be used in our setting where we encounter lots of user interaction and lighting changes through the projections. Thus, we want to use the depth data created by KinectFusion instead of the raw depth data for our tracking algorithm. Since the algorithm runs in real time and runs on the Graphics Processing Unit (GPU), it does not slow down our tracking procedure.

Figure 4.6 depicts a mesh created by KinectFusion[6]. The first image shows the table from above where only the outlines of the objects are visible, while the second image shows a side-view of the 3D model. Note that the created mesh does not separate the objects from one another but simply returns one big mesh containing everything the Kinect sees. This is not practical for further use because the individual objects of our created model should be easily movable and replaceable in CAD software. Also note that there is still a lot of noise which should be removed in our 3D model. For example, one can see the markers as bumps even though they are flat in reality. Furthermore, note that the vertical sides of the objects are divergent. If the sensor was moved around the objects in smaller distance, this noise would vanish, however, expecting the users to do so would be a strong distraction from the actual planning process. Therefore, we need to remove such noise automatically from the acquired data.

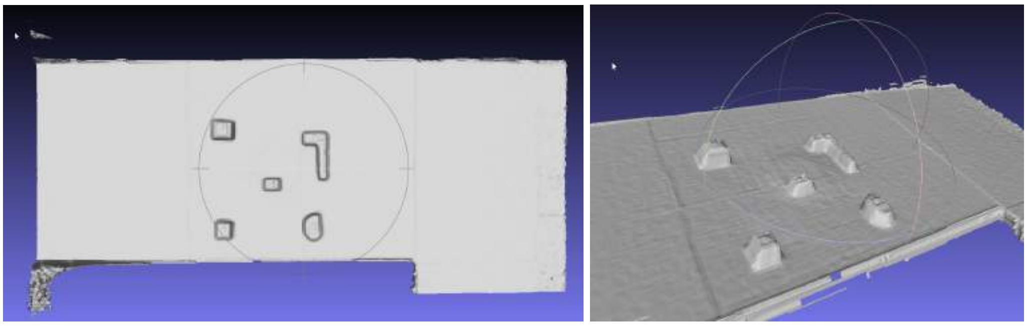


Figure 4.6: A mesh created by KinectFusion [27] from above and from a zoomed-in perspective (Rendered in MeshLab [10]).

[6]Parameters used: depth minimum = 1 m, depth maximum = 1.45 m; voxels per meter = 128; voxels in x, y, z = 384, 128, 384; integration weight = 50.

4.2.1.3 Object Separation

As already mentioned, we need to extract individual objects from the mesh created by KinectFusion to be able to easily arrange them in CAD software or exchange the model of a block by a more detailed CAD model if available. To achieve this, a similar approach as Corbett-Davies et al. [11] is used which starts by removing the table through thresholding. Specifically, we crop every point which is less than 5 mm above the table, beside the table (e.g., a user) or too far away from a newly detected marker to be part of a new object. Removing as many points as possible minimizes the data that needs to be handled and therefore increases the efficiency of the algorithm. We require users to initially place each object without a direct neighbor to ensure that newly placed objects are separated after removing the table. Figure 4.7 shows the remaining mesh after cropping as described above. Note that even though the individual objects are now separated in space, they all still reside in the same data structure from which they need to be separated.

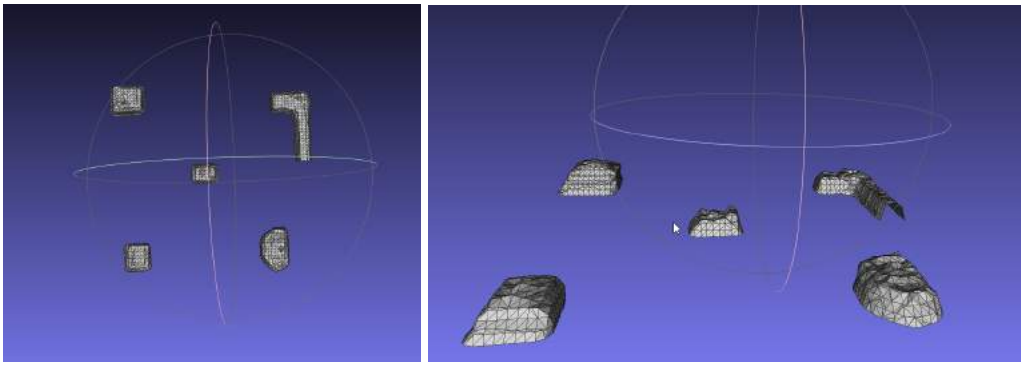


Figure 4.7: The remaining mesh after cropping.

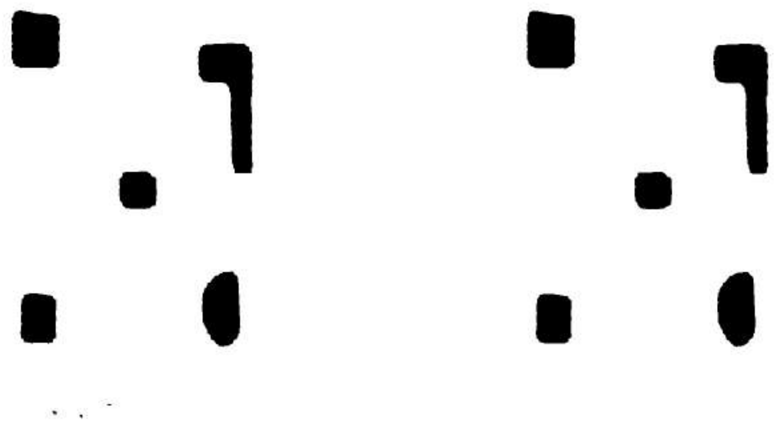
4.2.1.4 Object Shape Extraction

In the next step, the shapes of the objects are extracted for the following two reasons: first, the objects need to be separated from each other in memory and second, noise such as the bumpy surface and divergent sides should be removed. Creating cleaner real 3D models than KinectFusion [27] is difficult because the algorithm already performs a lot of noise reduction through temporal smoothing. However, for creating rough factory layouts, usually only 2.5D objects are built because cutting real 3D objects out of Styrofoam is very difficult. That is, they can have arbitrary, even concave, shape in the plane, however their floors and ceilings are flat. In contrast to simple 2D planning, the visible height of a machine or shelf can help imagining how it will look in the factory. In this simpler 2.5D setting, we can reduce noise significantly.

Analogously to Corbett-Davies et al. [11], we perform the next few steps completely in 2D. Since their work focused on interacting with tracked objects in an Augmented Reality environment, they used the efficient algorithm presented by

Chang et al. [8] to separate the objects. Since real-time behavior is not as important in our scenario, but we have the additional requirement of noise reduction, we use a slightly different approach. We start by projecting all points to a 2D plane by ignoring their z-coordinate. A black and white image is created by a linear mapping of the objects' points from real world coordinates to black pixels in an image with white background.

If all points within the shape of an object were black and all those in the background were white, we could easily find the contours by border following [45]. To achieve this, we have to ensure that the projected black points are connected. Therefore, we apply morphological opening [20] on the binary image which connects black areas being close together. Figure 4.8a shows the binary image after morphological opening. As you can see, there is still some noise in the lower left area of the image. Such artifacts can be removed by performing the opposite of a morphological opening, namely a morphological closing which connects white areas being close together. After opening, the image looks like Figure 4.8b.

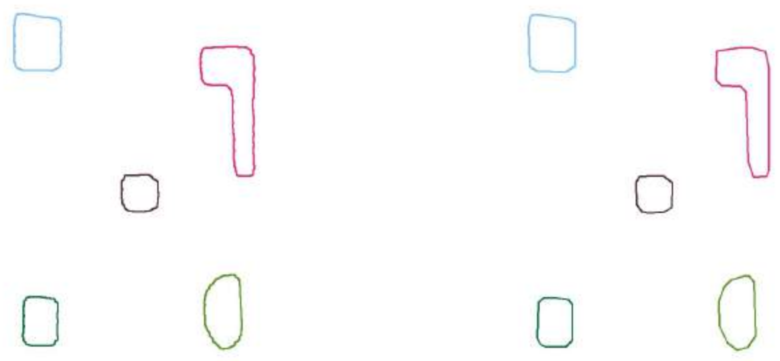


(a) Morphological opening. (b) Morphological closing.

Figure 4.8: The morphological opening of the objects projected to 2D and the closing of the opened image to reduce noise.

The next step is to use the border following algorithm by Suzuki and Abe [52] for extracting only the outermost borders. It is an extension to the simple border following algorithm explained and proven in [45]. Since objects with holes are not required for rough factory layouts, inner contours can only exist if there is so much noise that the morphological image operations do not close all holes. The contours extracted from Figure 4.8b are depicted in Figure 4.9a.

By mapping the detected markers to the same space, we can check how many markers are contained in each found contour. If there is exactly one newly detected marker in a contour, we have found its corresponding object. If more than one marker is found in a contour, multiple objects were placed next to each other. If one of these markers is a newly detected one, the user is prompted to



(a) Extracted contours. (b) Simplified contours.

Figure 4.9: The contours are extracted using Suzuki and Abe's [52] algorithm and then simplified by the Douglas-Peucker algorithm [15] (the colors were chosen randomly).

place new objects on their own for an initial build. If no marker is in a contour we can ignore it because it is just a hand or some other obstacle on the table which should not be tracked since it does not have a marker on it.

4.2.1.5 Model Cleanup

As can be seen in Figure 4.9a, the found contours are not perfectly straight. To remove this fine noise, we use the Douglas-Peucker algorithm [15] which straightens contours[7]. The simplified contours can be seen in Figure 4.9b.

Another shape simplification algorithm was implemented because it is difficult to cut Styrofoam really precisely. For example when cutting a cuboid, the sides are usually not perfectly parallel. Additionally to these errors produced by the users, the noise in the data slightly changes the shape. As an example, consider the three leftmost blocks in Figure 4.9b. When cutting, they were meant to be cuboids, however, their resulting shapes are not. Furthermore, the object in the lower right is meant to be convex, but contains tiny indentations. For these cases, our algorithm uses convex hulls or Oriented Minimal Bounding Boxes (OMBBs) to further simplify shapes. An OMBB is the smallest arbitrarily oriented rectangle enclosing a polygon. Our tracking algorithm builds the convex hull and OMBB of each object and compares their areas to the area covered by the original contour. If the OMBB is less than 10% bigger than the original polygon, the OMBB is used instead of the original contour. If this is not the case, but the convex hull area is less than 5% bigger than the contour area, the convex hull is used. These percentages are a trade-off between simplifying the model and guessing wrongly what the user might have tried to cut.

[7] We used an epsilon of 0.008.

An image of a contour and its corresponding convex hull and OMBB can be seen in Figure 4.10. To compute the convex hull, we implemented the gift wrapping algorithm [29]. In general this approach is outperformed by algorithms such as Graham Scan [21], however, in our scenario where only few points define the convex hull, gift wrapping is extremely fast because all individual computations are very efficient. For computing the OMBB, we implemented the rotating calipers algorithm [47, 53] which exploits the fact that an edge of the convex hull coincides with an edge of the OMBB. The other three edges are then chosen to touch the convex hull on at least one point on each edge. Figure 4.10 on the right shows all possible oriented bounding boxes for the convex hull. The smallest of these rectangles defines the OMBB.

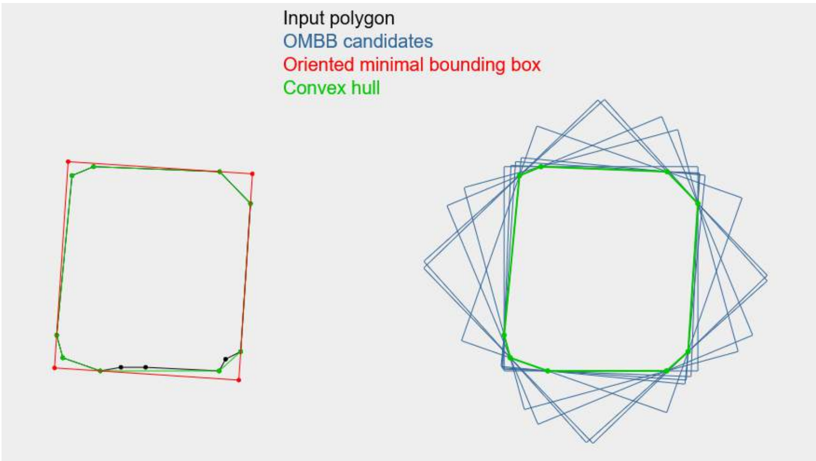


Figure 4.10: The convex hull and oriented minimal bounding box of a tracked object[8].

4.2.1.6 Computing Height

Now that the outlines of the objects' shapes are calculated in the plane, the only thing missing to create 2.5D digital models are their heights. We created and analyzed height histograms of the original 3D points corresponding to the shapes at millimeter precision and searched for a heuristic returning the height from such a histogram. Empirical observations revealed that the physical height corresponds to or is very close to the first peak in the histogram (if ordered by descending height). Thus, all points above this peak seem to be noise, while most points below are part of the vertical, divergent walls. Since exceptions to the rule occur when there is a lot of noise, we implemented a fallback using the highest value when no peak exists within the first four intervals. Figure 4.11 shows a histogram where the x-axis represents the height in centimeters and the red bar marks the value chosen according to our heuristic. This might not seem intuitive

[8]Image created using an open-source Javascript OMBB implementation `https://github.com/geidav/ombb-rotating-calipers` [last accessed 06/09/15].

at a first glance but can be easily explained: the first few medium height bars represent the top side of the tangible where points are strongly scattered. Further to the right are a few really large peaks with almost nothing between them, which correspond to the points captured at the divergent walls. The high density at those few points relates to the fact that KinectFusion [27] scans at evenly spaced position, therefore, with so little change in x and y as we have at the divergent walls, we only have very few different z values as well.

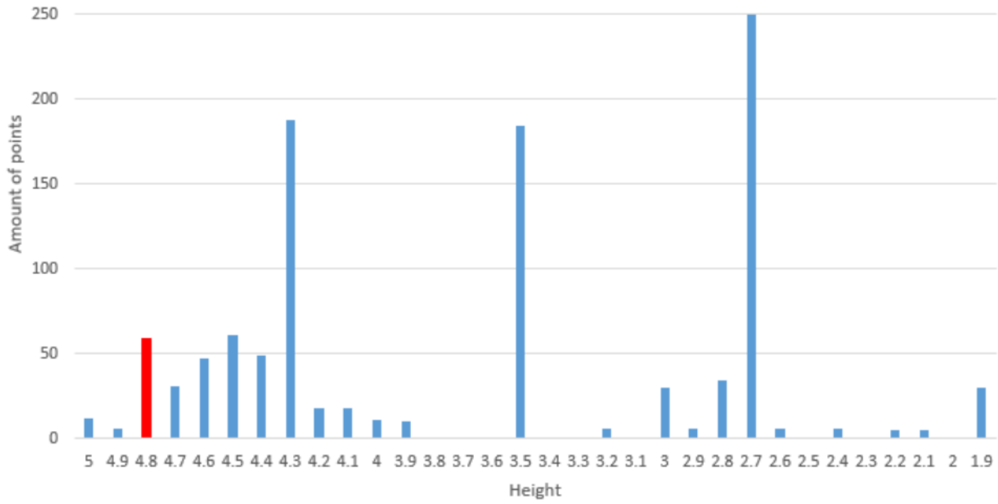


Figure 4.11: Height histogram of an object at millimeter precision: the value of the red bar is selected by our heuristic.

4.2.1.7 Building Objects

Once we have the cleaned up contour and the height of each object, we can easily create meshes of the objects, that is, collections of simple 2D polygons in 3D space describing a complex 3D shape. We decided to use triangle meshes because they can easily be handled in memory: all points are stored in a long array and each three consecutive points define a triangle. Thus, the memory locations of all individual triangles are known which would not directly be possible when using general polygon meshes. The vertical meshing is done in a straight forward manner: each two points on the contour represent a rectangle in 3D which is spanned by two points on the lower and two on the upper edge of the object. This rectangle is simply meshed using two triangles. Since meshing horizontally is more complicated because the polygons can be concave, we use the ear clipping algorithm [36] for triangulation in this case. Many other meshing approaches exist: for example, Keil and Snoeyink [31] presented an algorithm which yields optimal quality meshes but is slow. Chazelle [9] showed that meshing in linear time is possible, however, the algorithm is very complex to implement and uses operations which are computationally very expensive, thus, the polygon needs to have many vertices until the algorithm beats other approaches in terms of runtime. A relatively fast approach without such complex operations was presented by Berg et al. [3], but unfortunately, it yields poor quality meshes. We found ear clipping [36] to be a good trade-off between complexity, runtime and mesh quality.

The resulting mesh can be seen in Figure 4.12. Note that the three physical blocks indeed became digital cuboids as their OMBBs were used. Furthermore, the round object is represented by its convex hull as it differed only slightly from the computed contour.

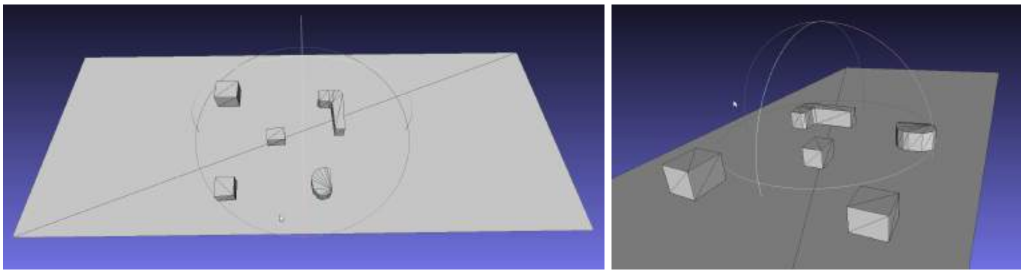


Figure 4.12: The resulting meshed objects.

4.2.1.8 Reducing Noise

As previously stated, the recognition process must be able to deal with lots of noise originating from measurement failures, changing lighting conditions or shadows coming from the planners. Another problem occurs when a user is still holding an object in his hand while it is being processed, or when it is moved before the recognition process is finished. The previously presented approaches for noise reduction (e.g., caching markers, using the KinectFusion data, simplifying contours) work well if the object is placed without any interaction, however, they cannot deal with hands or movement. To solve these problems and increase the recognition quality, objects are built multiple times until the created digital model is approximately the same twice in a row. Here, approximately means that the location is equivalent, the height differs by less than 2 mm, the areas of the contours by less than 5% and the orientations of the digital objects by less than 2.5 degrees. Again the chosen values represent a trade-off between precision and computation time. If too small, it might take a long time until two successive digital models are approximately the same, but when they are, we can be sure that the created digital model is very close to the physical one. In contrast, if too large, even very dissimilar digital objects are considered the same and therefore the recognition must be triggered less often.

4.2.1.9 Update Mechanism

All the previous sections in this chapter described how a digital model is created when a new marker is detected. Taken together, all these steps are quite computationally expensive (usually 1-3 s on a modern computer). Therefore, the created objects are cached together with the ID encoded by their markers such that they only need to be built once.

We want to apply all transitions and rotations of a marker to the corresponding object. This was implemented in two steps: for storing a new objects, we translate it such that the center of its marker is positioned on the origin and a side of the marker coincides with the x-axis. When changes to the position or orientation of a marker are detected, we simply translate the cached digital model to the new position and rotate it according to the angle between a side of the marker and the x-axis by multiplying each point with the rotation matrix corresponding to the angle.

When an object is removed from the table, it is also removed from the digital model (with a small delay since the markers are cached for several frames, cf. Section 4.2.1.1). Nevertheless, we do not remove the digital data from cache. This way, markers can be occluded for some seconds without the need to re-scan them afterwards. Furthermore, the temporary decision to remove a certain machine can be reverted without re-scanning the object.

4.2.2 Tape Recognition

Apart from planning with blocks (e.g., for machines), users can use differently colored tape to define the outline of the building, the pathways within the factory, entrances or material flow (cf. Section 3.2.3). Thus, DEPlaTa must be able to create a digital model of the tape on the planning table. We first present an algorithm extracting the shape of the tape and then show how DEPlaTa deals with occurring flickering effects by caching found contours.

4.2.2.1 Tape Shape Extraction

First, the RGB frame of the Kinect is accessed and cropped to the region of interest, namely the area tracked by KinectFusion [27] to minimize the amount of data that needs to be analyzed. Since slight changes in lighting can have high influence on RGB values, it is hard to track a color in different lighting conditions without re-calibrating. To prevent long set-up phases (cf. requirement R4), DEPlaTa recognizes tape in the Hue Saturation Value (HSV) color space [50] instead because it is more robust to lighting changes. The hue component of this color space corresponds to our natural color perception. Thus, a color can be calibrated by setting the respective interval of allowed hue values and defining a range of saturation and value to reflect the different lighting conditions.

With given intervals of hue, saturation and value for each color, a simple thresholding is performed on the cropped image converted to HSV color space. To reduce noise, morphological closing and opening operations are performed (cf. Section 4.2.1.4). For the color blue, the thresholded and mophologically processed image can be seen in Figure 4.13. Not only the shape of the tape but also four of the markers were recognized as blue by the procedure. These artifacts could be removed by tightening the minimum and maximum H, S and V values or by performing stronger morphological operations which remove noise of this size. However, the first approach would make the system less stable in different lighting conditions while the second approach would delete small contours entirely. Therefore, another filtering is performed later when the contours are mapped back to real world coordinates, which removes all contours at the positions of objects because those must be noise.

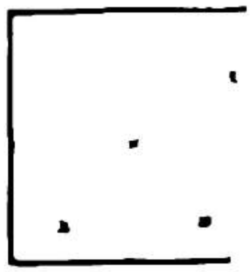


Figure 4.13: The HSV image thresholded by the given minimal and maximal H, S and V values. Also morphological closing and opening are applied.

Now, as during object recognition, Suzuki and Abe's border following algorithm [52] is applied. Afterwards, the found contours are again simplified using the Douglas-Peucker algorithm [15]. The result can be seen in Figure 4.14[9].

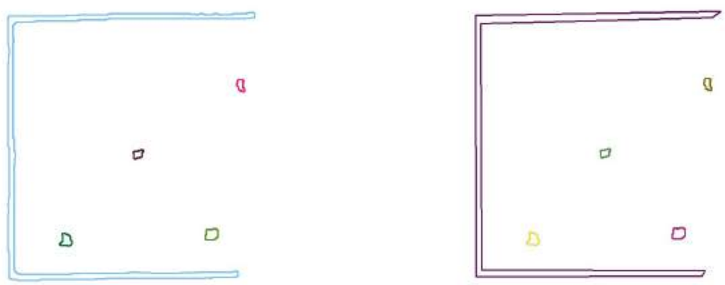


Figure 4.14: The contours and the simplified contours found for blue during tape recognition.

[9]We used an epsilon of 0.003.

The found contours are then mapped back to the coordinates in the uncropped image and from there to real-world coordinates by using the Kinect Software Development Kit (SDK). Analog to object creation, meshes are produced from the resulting coordinates in physical space by applying the ear clipping algorithm [36]. Figure 4.15 shows the resulting mesh for the blue tape. The whole tape recognition process is triggered at 50 ms intervals to reduce processor usage.

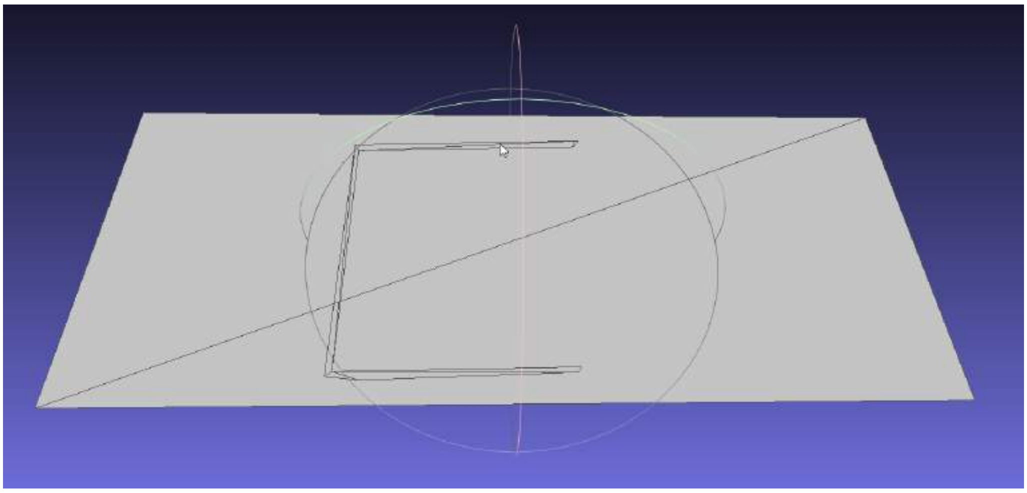


Figure 4.15: The final mesh created for the blue tape.

4.2.2.2 Tape Contour Caching

The algorithm presented above works, but we encounter flickering effects due to noise. In contrast to object recognition, the problem cannot be solved by attaching markers to the tape, because users might decide to enlarge the walls or transportation paths of their building. To deal with these flickering effects, DEPlaTa uses a custom caching mechanism: it stores the last used contours and compares them to the ones detected in the current frame. For each contour, the algorithm determines whether the old or new contour should be used. Thus, matches between the contours need to be found in the cached and current set. Since a large contour might be detected as several small ones which we would like to combine, a match is actually a subset of the cached contours and a subset of the current contours (cf. Figure 4.16a). To find these matching subsets, the algorithm first searches for overlaps of each pair of cached and current contours and then combines the trivial pairs to sets.

If a user extends a stripe of the tape, the new longer version and the cached shorter version would still be matched because they overlap in the whole cached part (cf. Figure 4.16b). In this case, we cannot simply use the cached version even though we found a match. To decide which version of the contour shall be used, two aspects are considered: first, the accumulated size of the contours must

(a) A valid match. (b) An invalid match.

Figure 4.16: Matches between cached and currently found contours.

be almost the same (allowed difference: 5%). This ensures that enlargements as in the last example are recognized. However, if this is not the case, the sets of contours might still represent the same tape: it could happen that an end piece is recognized as a single point instead of two because the Douglas-Peucker algorithm [15] oversimplified the contour (cf. Figure 4.17). In this case, the area is only half as big in as the actual area, thus it would be considered as an invalid match and the new contour would be used. To prevent this effect, a second criterion is checked whenever the areas do not match. It simply analyzes whether all new points have a corresponding cached point within a neighborhood of 10 pixels and the other way around. Here, the example of Figure 4.16b does not meet the criteria because the rightmost point does not have a partner. However, the example of Figure 4.17 is considered a match due to this criteria.

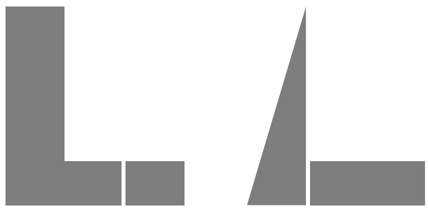


Figure 4.17: A falsely collapsed edge.

When a found match is indeed a match according to the above criteria, we decide whether to take the cached or the current set of contours based on the following rules:

1. If possible, use the description consisting of fewer individual contours (e.g., in Figure 4.16a the left set of contours would be used)
2. In case the areas match: if a description has an even amount of vertices while the other has an odd amount, use the even one. This ensures that an even number of triangles is used, which should be the case if built correctly, otherwise we might encounter the oversimplified case explained before (cf. Figure 4.17)
3. Take the contour with less overall vertices as it is simpler

In contrast, if no match in the cached contours is found or the areas and the points of the matches differ, we simply use the new contour. Thus, newly placed tape or extensions to existing tape are always recognized.

The cache removes the flickering after a few iterations, because once a simple description for the overall shape is found (i.e., a single contour with an even and small amount of points), this simple description is used until actual physical changes occur.

4.3 Model Interaction

Previously, we discussed the first component of our system which is responsible for creating a digital model. This section now presents the implementation details of our second system component handling all interactions with the created model. For example, users can digitally enhance, store and load models using this component. Thus, the GUI, the renderings for the projectors and the speech recognition are implemented here. Since this component contains few interesting algorithms, we only describe which programming languages and SDKs were used and then explain how the projectors were calibrated and how a mapping from real world coordinates to pixels in the projection areas was realized.

4.3.1 Component Setup

We decided to implement this component in C# using Windows Presentation Foundation (WPF) because it allows to easily create natively looking interfaces and the .NET framework offers a lot of functionality to rapidly develop the required features. The store and load functionality is implemented by simply serializing the classes to JavaScript Object Notation (JSON) and recreating them from the stored strings. However, the digital models themselves are exported to Wavefront .obj format to be usable in external software. All values which need to exist across planning session (e.g., properties, prototypes or menu settings) are stored in a SQLite database. For the speech recognition, we decided to use the speech SDK offered by Microsoft, because it is customized for the microphone array of the Kinect v2 sensor.

4.3.2 Projector Calibration

To align the two projectors with the Kinect's field of view, calibration is required when the system is initially set up. Figure 4.18 visualizes the different spaces used during calibration. Note that the KinectFusion [27] algorithm tracks a smaller area than the Kinect's RGB camera. The first step during calibration is to determine the areas of the projections which enlighten the tracked area of the table without overlap (i.e. the green and blue boxes in Figure 4.18). Then, the six points defining the borders of these two projection areas on the Kinect's RGB image need to be entered into the program. To map a real-world 3D point to a projector, it is first mapped to the Kinect's RGB space using the SDK's mapping functionality. Then, the corresponding projector and pixel of its projection can be determined by a simple linear mapping using the entered points.

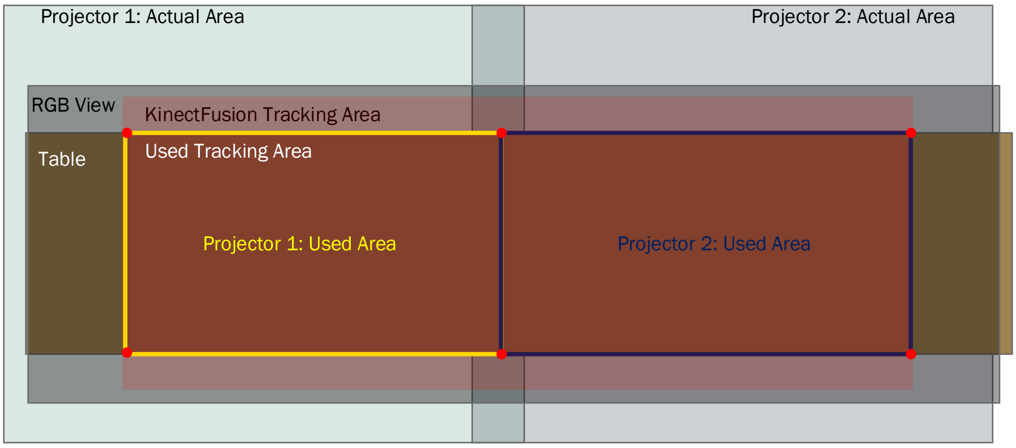


Figure 4.18: The different spaces used for calibrating the projectors.

Chapter 5
Evaluation

We conducted an evaluation to inspect the usability of our digitally enhanced planning approach and its individual features and test whether our approach facilitates or hinders the planning process. Additionally, the evaluation provides insights in possible system improvements and useful extensions.

5.1 Participants

A purposive sampling [1] approach was used, selecting experts on planning factory work floors for domestic cooking appliances. All participants were employees of the same company where the initial requirements analysis was conducted, thus, they could assess whether the initial requirements were implemented as expected. Currently, rough factory layouts in this company are planned completely analogously by arranging Styrofoam, wooden models and Playmobil figures on a large table. Since this approach is very similar to our concept, the participants were able to judge DEPlaTa in comparison to their current pure analog approach.

To receive a realistic impression of the planning process and gather meaningful usability feedback [39], the evaluation was conducted with five planning experts which is a typical amount of persons involved in a planning session. Three of them already participated in the initial requirements analysis, however, the other two were completely new to the topic. All participants were male ($M_{age} = 33.6$, $SD = 7.7$) and had experience in rough factory layout planning ranging from 1.5 to 15 years ($M_{exp} = 5.9$, $SD = 5.4$). The five experts were also experienced in CAD planning but preferred different programs. Nevertheless, they clearly focused on analog planning: four participants stated that they plan between 75 and 100 % of the time with the previously described analog approach whereas only one participant planned digitally in most cases (85 %). Thus, the experts were also good candidates to judge the usefulness of our digital extensions.

5.2 Apparatus

To conduct the evaluation, we set up our system at the participants' company in the room where planning is usually done. Thus, the planners were accustomed to the environment and could work with their usual tools (e.g. a hot-wire cutter for the Styrofoam). The planning space was significantly larger than our first prototype and covered an area of $1.56 \times 1.40\ m$. Since the table was not as long compared to the original prototype and the distance to the ceiling was larger, a single projector was sufficient. Furthermore, the white surface of the table was even better suited than our wooden table as the increased contrast improved the visibility of the projections. Instead of a computer monitor for the GUI, we attached the computer to a SmartBoard which was already available in the room. The increased screen size made the GUI visible for all participants independent of their current position at the planning table. Figure 5.1 shows the planning table with the SmartBoard in the background. Apart from a recalibration procedure, no adjustments to the software were necessary to work in this new setup. This ease of setting up DEPlaTa at another location shows the flexibility of our approach, as required by R4 of the initial requirements analysis.

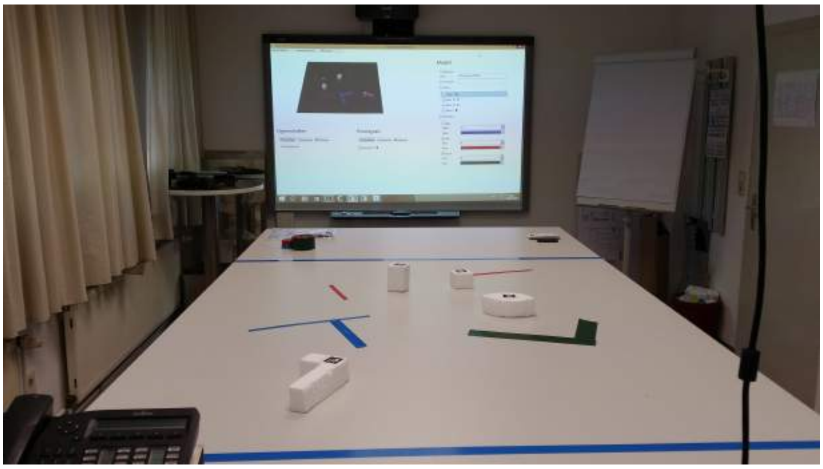


Figure 5.1: The system setup for the evaluation.

5.3 Method

For the evaluation, we first handed out a questionnaire to gather demographic data, ask the participants about their planning background and assess drawbacks with their current purely analog planning approach.

Afterwards, a planning task was presented by the participants' superior. It was a task which has not yet been planned in an analog session, but which was familiar to all participants and actually needed to be addressed in the near future. Using a realistic scenario minimized the risk of receiving non-reliable results which might have happened if we had created an artificial task. The superior illustrated the task on a flip chart and stated the operating resources and the optimization goals: ensuring an optimal value flow, protecting workers from noise and easy extension possibilities for future growth. Two very large machines were fixed and could not be rearranged while all others were flexible. Furthermore, the participants did not only have the option to rearrange the existing machines, but could also decide to use new machines that are currently not present. A 2D layout of the factory was printed on a large board as a reference. After presenting the topic, the superior left the room.

We then briefly presented DEPlaTa and all its features before the participants started planning the task presented above. The planning session itself lasted approximately one hour. After that, a post-session questionnaire was handed out and a semi-structured group interview was conducted to analyze the strengths and weaknesses of the system and find possibilities for improvement. We asked about the general usability and the quality and usefulness of the created digital model. Furthermore, we analyzed each existing feature (e.g., object recognition, speech commands) separately to gather specific improvement suggestions and assess useful extensions to the existing functionality of DEPlaTa.

Two experimenters observed the whole evaluation and took notes. Another experimenter led the interview and answered arising questions during the planning phase. Furthermore, he was prepared to lead the planning experts to specific features in case they would not use them on their own. To make sure that no valuable feedback would stay unnoticed, the whole session was videotaped with a camera focusing the planning table.

5.4 Results

We present the results of the evaluation in four categories: first the workflow during the planning phase is described. Then we report the participants' impressions of our system with respect to their currently used analog planning approach. Afterwards, we discuss possible improvements to existing functionality and finally we present potential extensions to the system which were proposed by the participants.

5.4.1 Workflow during the Planning Phase

In the beginning of the planning phase, the participants calculated a scale in which the factory fits on the table and then attached the tape representing the walls to the table. Most of the needed objects were cut with a hot-wire cutter in the beginning of the planning session and placed on the table such that DEPlaTa scanned and cached them. For calculating the correct dimensions, the participants used the 2D factory layout and adapted the dimensions to the scale used on the table. One person then created prototypes and assigned them to the digital representatives of the newly placed objects which he also named. Properties were not used because they would be superfluous in this initial planning stage according to a participant.

During the first couple of minutes, the participants experimented a lot with the system and tested the tracking capabilities. However, after this short phase, they mostly planned analogously while the system was running in the background. We observed a clear distribution of roles when adapting the layout: one participant was cutting the Styrofoam objects, while another attached the pre-printed markers on the tangibles and placed them on the table such that they were initially recognized by the system. The GUI was handled by another participant and the paths were also primarily placed by a single person. Only for larger changes, he received help from the other participants. Nevertheless, all except for one participant were strongly involved in the planning itself. This parallel processing clearly shows how DEPlaTa supports collaboration, especially compared to CAD software. It is also interesting to note that the participants regularly pointed to Styrofoam pieces while talking about machines and directly moved them to other locations to propose alternative layouts. Apart from the Styrofoam objects, Playmobil forklifts and figures were used. To analyze the behavior of the system, the participants attached markers to these objects as well. It did not bother the participants that the created digital models were in 2.5D. For the pathways, the right and left boundaries were taped in the correctly scaled distance which was recognized very well by the system. The participants used all features on their own, for example they saved a version of their plan unsolicitedly. Afterwards, they cleared the whole table and started the planning process anew. Thus, two different layouts were planned for the given task. The second state, which can be seen in Figure 5.2, was preferred by the participants. Further observations and how they might be improved are presented in Section 5.4.3.

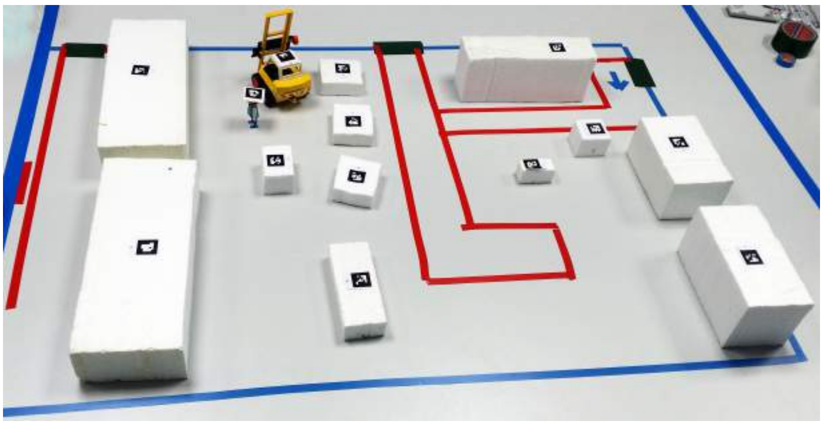


Figure 5.2: Final planning result created during the evaluation.

5.4.2 Comparison with the Experts' Current Approach

The participants identified several problems with their current analog planning approach: four reported that the manual digitization is time-consuming and sometimes difficult. According to one participant, it would take approximately two hours to reach the level of detail that DEPlaTa offers in just seconds. Since the planners do not want to interrupt a planning session for this amount of time to digitize an intermediate state, they currently only transfer final planning states to the digital domain. This makes it really hard to archive and document the planning process. Furthermore, only the few digitized states can be analyzed and compared in simulation software. Another drawback of purely analog planning as stated by the participants is the transportation of a planning state to another location. Here, our system would also help rebuilding the state by projecting the positions of the objects on the table.

The additional effort of attaching adhesive markers to the tangibles was perceived as unproblematic and would not distract from the actual planning process according to all participants. One participant stated that the major advantage of our system is the facilitation of creative planning because it allows to start on a clean planning table and test alternative layouts without being afraid to lose old states. Once a good layout is found with their current purely analog planning approach, the planners usually keep it and only try to optimize it further. Since they have no possibility to quickly store a layout, they fear to completely remove everything and start anew. The simple restoring mechanism of our system therefore allows more flexible planning. While this would also be possible in CAD systems, these programs would hinder collaboration according to the participants. Only the person handling the computer would effectively do all the planning, while the others just watch without participating. Furthermore, it was stated that rearranging objects on the table is more elegant than in CAD software. The analog Styrofoam model also offers the advantage that it can be easily shown to decision makers who prefer simple analog models over CAD models.

5.4.3 System Improvements

Generally, all participants were able to use the system without noticeable problems regarding the main planning task. However, we found a few usability aspects that can be improved:

Object Recognition

The current setup in which the camera used for tracking is located at the ceiling, resulted in not completely accurate digital models when introducing objects at the very edges of the planning area. Since the table used for the evaluation was significantly wider than the one of our laboratory setup (80 cm vs. 140 cm), we did not know about the diminishing model quality in advance. This large table covered the whole viewport of the camera which resulted in noisy data at the border due to the tilted perspective. We told the participants that the model quality can be increased by avoiding initial placements at the edges. This worked well for the evaluation, however, as pointed out by a participant, a problem might arise when only the edges of the table are non-occupied.

A solution to this potential problem could be to have a designated scanning area outside of the actual planning space where the initial scanning process is performed. The recognition quality might also improve in general with this approach as the distance between the camera and table can be reduced which makes the data more precise. A standard RGB camera could then be used for the marker and tape recognition. If this camera has a higher resolution than the Kinect, the size of the fiducial markers could be decreased, thereby allowing extra small objects which are sometimes used by the participants. Furthermore, an even larger table might be used in combination with multiple RGB cameras whereas using multiple Kinect v2 sensors with one computer is not possible due to technical restrictions by Microsoft. Since most tangibles were created before starting the planning process, the extra scanning step at the dedicated scanning area would probably be acceptable.

Even though the recognition could be improved as previously stated, all participants were already content with the quality of the created digital model in its current form.

Tape

Using colored adhesive tape for the outlines of the building or different paths was generally perceived positively. However, the participants stated that its usage could be improved when reloading stored versions because it was cumbersome to remove. This feeling might have been reinforced because one of the four supported tapes was strongly adhesive duck tape which even left adhesive residues on the table. A simple solution would be to use colored paper stripes, however, these might easily shift on the table. One participant proposed the use of a table with a whiteboard surface and colored markers to draw on it. Furthermore, we learned that seven colors are consistently used throughout the whole planning process, each with a different meaning. DEPlaTa could easily handle this amount of colors if calibrated accordingly.

For the evaluation, the system was calibrated to easily distinguish the different colors and also to work in many lighting conditions. This was achieved by thresholding to relatively large intervals of hue, saturation and value for extracting the shape of the tape. However, it resulted in the participants' sleeves and hands regularly being detected as tape. With ten hands and sleeves as well as a huge amount of objects on the table, the system slowed down significantly such that displacements were only recognized after several seconds. This could be solved by a tighter calibration. However, none of the participants mentioned the delay as a problem as no real-time constraint exists.

Speech Recognition

The planning experts really appreciated the general availability of a speech recognition because they liked the idea of interacting with the system without using the GUI directly. Since the recognition was untrained to the participants' voices and the microphone was located at the ceiling and also received the sounds of the neighboring projector, the recognition was not optimal. We discussed potential solutions, namely using a table microphone or headsets to improve the recognition quality. The planning experts stated that they would be willing to use any of these alternatives to increase the recognition quality. Especially the participant handling the GUI expressed the wish to improve the speech recognition. However, he also noted that the time needed to handle the GUI would decrease over time once the prototypes are stored in the database and need only be attached without prior creation. Furthermore, the possibility of creating a set of standard objects which can be used across planning sessions was recognized: once created objects with assigned name and prototype can simply be put in a box and then used for multiple layouts without any need to use the GUI or speech recognition. Therefore, the importance of the speech recognition would decrease over time.

GUI

The GUI was perceived as clearly structured with a sufficient amount of functionality. According to the participants, more options would only complicate interaction and hinder the planning process. Features like running simulations would be executed in separate external software on the created digital model. Only a couple of small improvements were proposed. For example, it became obvious that the amount of prototypes after multiple planning sessions would be above 100, thus, a filtering mechanism to find a prototype in the long list would be helpful. Furthermore, users should be able to enter the meaning of a tape (e.g., wall, path for forklift) once to store it in a database such that it is available in further planning sessions.

5.4.4 Feature Requests

Since the participants had no prior experience with a combined analog and digital planning solution, it is not surprising that extensions to the features of the initial requirements analysis were requested after using DEPlaTa for the first time. We derived the following feature requests from the planning experts' statements during the semi-structured group interview:

FR1 Projections to the real world
Digital information should be projected onto the real world. For example a floor plan is usually available digitally and could be projected on the table in the correct scale. Thus, users would no longer be required to mark the outlines using adhesive tape. If no digital floor plan is available, projecting a grid in the correct scale would also help the planners aligning the tangibles. Furthermore, the assigned names and prototypes could be directly rendered on the tangibles to impose meaning to the physical world. While projections on large objects could be easily realized, a problem might arise for projections on small objects in crowded scenes where only little space is available.

FR2 Scale conversion tool
A scale conversion tool should be implemented which converts real world measurements to the selected scale. This would speed up the creation of the physical model by supporting users cutting tangibles out of Styrofoam and placing the tape.

FR3 Extended Export
All additionally entered information should be included in the exported files to be utilized in other 3D programs. Even though the Wavefront OBJ format can be universally read by almost any 3D software, it does not support all digital enhancements that DEPlaTa offers (e.g., prototypes or descriptions). Therefore, we should use another format in the future.

FR4 Special Tangibles for Material Flow
Since the planners need to analyze material flow very frequently, predefined objects for simple visualization should be provided. According to the participants, simple wooden arrows with fiducial markers attached to them would be sufficient. The digital model of an arrow is then simply loaded and placed at the correct position instead of actually tracking the shape.

FR5 Better Connection to CAD Software
A better connection to CAD software should be created. As an example, it is often the case that some machines cannot be moved when re-designing a plant. Instead of creating Styrofoam representatives, a CAD model of the machine should be read and projected on the table. Furthermore, when a model created by DEPlaTa is changed in CAD software, these changes should be represented in the system. For example, DEPlaTa should detect new digital objects and support the physical creation process by displaying the dimensions for cutting and the position where the new tangible should be placed. Another requested feature is to load more detailed digital 3D models into DEPlaTa which are then used for rendering in the GUI. Furthermore, a tangible could also be top-augmented with a rendering of the detailed model to better visualize material inputs and good outputs. However, the practicability of this feature is questionable as the planning experts nowadays have only very few digital models of their machines.

5.5 Discussion

Even though the study only analyzed a planning process snapshot, we have already received valuable feedback onto how our system can support experts. The results of our evaluation clearly indicate that DEPlaTa helps users to easily test multiple design alternatives which facilitates creativity. In contrast to CAD software, the system also supports collaborative planning. The quality of the created digital model is satisfying for the participants' needs and the additional effort of attaching markers does not disturb the planning process. Approximately two hours of work are saved each time a state needs to be stored. Furthermore, we received insights about possible extensions to the system and found out how existing functionality can be improved.

The main limitation of our study is that only a single planning session was observed. To receive further insights into the usefulness of DEPlaTa, a long-term evaluation is required. Especially the possibilities to easily store and load layouts can only be assessed in detail if used over several weeks or even months. While evaluating the system with a group of five planning experts reflected a realistic scenario and allowed insights into the collaborative aspects of our system, only a single group of experts ever used the system. It would also be interesting to conduct the evaluation with further participants to gather more opinions on the system. Especially the opinions of experts in digital planning who are not used to analog planning could provide further insights. These new participants should ideally work at another company or at least in another production domain to ensure external validity of the results.

Chapter 6
Conclusion and Future Work

In the following section, we summarize the work done in the course of this thesis. Afterwards, possible extensions to the system and future research directions are presented.

6.1 Summary

This section summarizes the results of this thesis and explains how we fulfilled the initially stated research goals (cf. Section 1.4) which are repeated here for a better overview.

Conceptual design of a rough factory layout planning system

To develop a concept for a rough factory layout planning system, we first analyzed currently used pure analog and pure digital approaches and noticed that they show severe drawbacks. Motivated by these problems and by an initial requirements analysis, we wanted to create a system bridging the gap between the analog and digital worlds [14] by combining their advantages without assimilating their drawbacks. We showed related planning approaches with the same goal, however, they only allow rearranging a set of predefined objects. As the main contribution of this thesis, we want to support the creation of new objects during planning.

Our tangible factory layout planning tool called DEPlaTa allows analog planning with models cut out of Styrofoam or other materials for representing machines, workbenches or supply areas. Furthermore, colored adhesive tape can be used to define the walls of a building or paths throughout a factory. Since the objects and tape can be quickly cut and easily arranged on the table, our approach supports rapid prototyping and allows even untrained users to participate in the planning process. The physicality of the 3D objects motivates users to directly interact with the model and facilitates spatial awareness during planning. Multiple users standing around the planning table can extend and adapt the physical model at the same time which is a crucial aspect for collaborative group work.

Apart from these analog functionalities, the system offers features that are currently restricted to digital solutions such as CAD software. This is achieved by automatically creating a digital model of the arbitrarily shaped objects and tape on the table. To impose meaning to the digital representatives, users can add names and descriptions or use so called prototypes and properties. The latter two can be created once and can then be attached to multiple objects even across planning session with a single markup step in the GUI or via speech commands. The digitally enhanced model can be exported in a standard format for archiving, running simulations in external software or as a basis for fine planning. When re-importing a saved layout, an easy and precise rebuilding mechanism which renders the locations of the individual blocks and tape directly on the table helps users rebuilding the physical state. The system automatically recognizes correctly placed tangibles and switches off the corresponding projections. To the best of our knowledge, no other planning system exists which automatically creates a digital model of physical representatives on the table and supports the reconstruction of former physical states.

Implementation for the usage in a planning environment

To implement the previously described concept, we built two hardware prototypes of the system: one in our laboratory and a second one for the evaluation. Each prototype consisted of a large planning table, a Microsoft Kinect v2 mounted above the table and one or several projectors to augment the table. We developed an algorithm which uses the noisy real world data acquired from the Kinect to create individual digital models of the objects and the colored tape on the table. The resulting digital layout is constantly updated in the background without disturbing users in their analog planning. Furthermore, the model creation process is robust to user interaction and takes care of crookedly cut edges to produce a clean digital model. A GUI and a speech recognition were implemented to allow users to enhance and manage the digital models.

Investigation of the planning experts' interest in the system

We conducted an evaluation of the system at a large German manufacturer with five experts on planning rough factory work floors. We found that the participants much appreciated the combination of analog and digital planning. The analog aspects of DEPlaTa, which are similar to their current planning approach, facilitate groupwork and help in discussions with decision-makers. According to a participant, our digital extensions reduce the time needed to create a digital model of a layout from approximately two hours to just seconds. Using DEPlaTa, the experts now store intermediate states which was not done previously due to the time consuming manual digitization process. According to the participants, the main advantage of our system is that it facilitates creativity by helping users to easily test multiple design alternatives without losing potential good states. The quality of the created digital model is completely satisfying for the participants' needs and the additional effort of attaching markers to the representatives does not disturb the planning process. Furthermore, we found possible extensions to the system and improvements to existing functionality.

6.2 Future Work

As a next step, DEPlaTa should be adapted to the results of the evaluation. Thus, the found improvement possibilities of existing features should be implemented, for example, smaller markers should be used in combination with higher resolution cameras. Furthermore, the newly proposed extensions to the system, such as projecting floor plans on the table, should be integrated into DEPlaTa. The improved and extended system could then be evaluated in a long-term study to examine the usefulness of features like store and load in a planning process lasting several weeks. Ideally this extended evaluation would be conducted at multiple companies to ensure external validity of the findings.

Furthermore, it would be interesting to adapt the concept of DEPlaTa to other planning tasks than rough factory layout planning. Depending on the planning task, the tracking algorithm might need to support real 3D objects. This could be realized by moving the Kinect in rails above the table such that the KinectFusion [27] algorithm captures the object from all sides and creates a higher quality mesh. Optimally, our current recognition algorithm would still be used for physical 2.5D objects as it yields clean digital models. Only for real 3D, we would extract the object from the mesh created by KinectFusion [27]. The classification when to use which of the two approaches might be realized using a machine learning algorithm on the height histograms of the objects.

Another interesting aspect would be to extend the system such that it allows planning at multiple locations at the same time. Objects created at one location could simply be 3D-printed at the other location and then placed on a projection appearing at this location. All translations and rotations performed by the users in one place then need to be visualized at the second planning location. In a next step, the objects could even be moved and rotated automatically at the remote location whenever a motion is recognized. This could be realized by emitting ultrasonic shock waves or generating air pressure to give small impulses to the Styrofoam blocks and move them along the table. These approaches require no user interaction to synchronize the analog models across locations. However, the implementation might be difficult, especially in crowded scenes.

Appendix A
Graphical User Interface

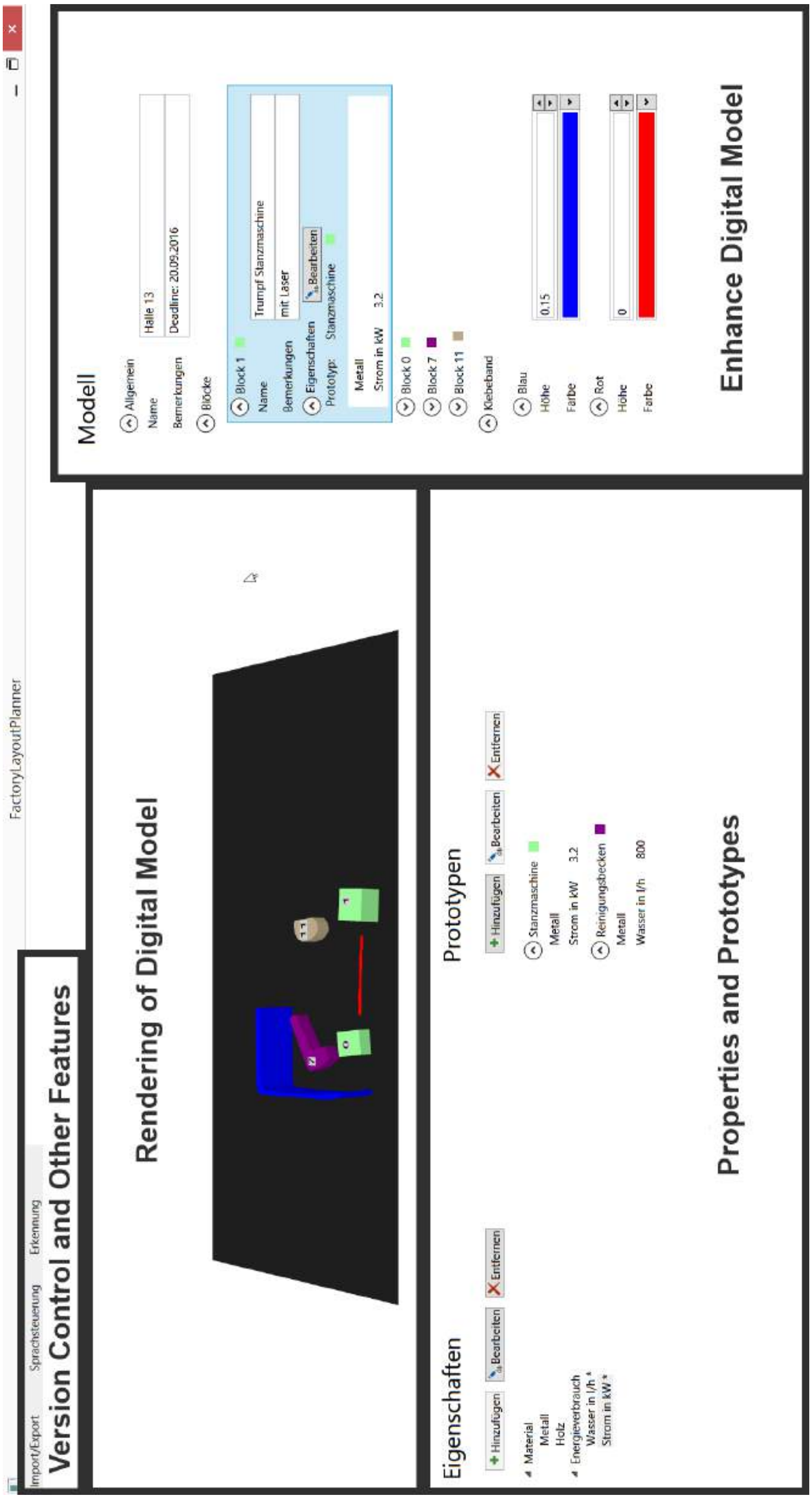


Figure A.1: DEPlaTa's Graphical User Interface with overlays for the different sections.

Bibliography

[1] BABBIE, R. *Survey Research Methods*. Wadsworth Publishing Company, 1990.

[2] BASTIAN-KÖPP, D. *Cooperative Design of Manufacturing Systems in SMEs: Development of an Integration Approach and a Groupware-based Cooperation Concept*. Informationstechnische Systeme und Organisation von Produktion und Logistik. GITO-Verlag, 2009.

[3] BERG, M.; CHEONG, O.; VAN KREVELD, M. AND OVERMARS, M. *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag TELOS, 2008.

[4] BILLINGHURST, M.; KATO, H. AND POUPYREV, I. Tangible Augmented Reality. `http://www.csie.nuk.edu.tw/~ayen/teach/ar/ref/Tangible%20Augmented%20Reality.pdf` [Online; last accessed 25/08/15].

[5] BILLINGHURST, M.; KATO, H. AND POUPYREV, I. Collaboration with Tangible Augmented Reality Interfaces. In *Proceedings of the 9th International Conference on Human-Computer Interaction* (2001), HCII '09, pp. 797–803.

[6] BONNARD, Q.; LEMAIGNAN, S.; ZUFFEREY, G.; MAZZEI, A.; CUENDET, S.; LI, N.; ÖZGÜR, A. AND DILLENBOURG, P. Chilitags 2: Robust Fiducial Markers for Augmented Reality and Robotics., 2013. `http://chili.epfl.ch/software` [Online; last accessed 02/08/15].

[7] BONNARD, Q.; VERMA, H.; KAPLAN, F. AND DILLENBOURG, P. Paper Interfaces for Learning Geometry. In *Proceedings of the 7th European Conference on Technology Enhanced Learning* (2012), EC-TEL '12, pp. 37–50.

[8] CHANG, F.; CHEN, C. AND LU, C. A Linear-time Component-labeling Algorithm using Contour Tracing Technique. *Computer Vision and Image Understanding 93*, 1 (2004), pp. 206–220.

[9] CHAZELLE, B. Triangulating a Simple Polygon in Linear Time. *Discrete & Computational Geometry 6*, 1 (1991), pp. 485–524.

[10] CIGNONI, P.; CALLIERI, M.; CORSINI, M.; DELLEPIANE, M.; GANOVELLI, F. AND RANZUGLIA, G. MeshLab: an Open-Source Mesh Processing Tool. In *Proceedings of 6th Eurographics Italian Chapter Conference* (2008), EG-Italy '08, pp. 129–136.

[11] CORBETT-DAVIES, S.; GREEN, R. AND CLARK, A. Physically Interactive Tabletop Augmented Reality Using the Kinect. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand* (2012), IVCNZ '12, pp. 210–215.

[12] DALSGAARD, P. AND HALSKOV, K. Tangible 3D Tabletops: Combining Tangible Tabletop Interaction and 3D Projection. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction* (2012), NordiCHI '12, pp. 109–118.

[13] DOIL, F.; SCHREIBER, W.; ALT, T. AND PATRON, C. Augmented Reality for Manufacturing Planning. In *Proceedings of the Workshop on Virtual Environments* (2003), EGVE '03, pp. 71–76.

[14] DONG, S. AND KAMAT, V. R. Collaborative Visualization of Simulated Processes Using Tabletop Fiducial Augmented Reality. In *Proceedings of the Winter Simulation Conference* (2011), WSC '11, pp. 828–837.

[15] DOUGLAS, D. H. AND PEUCKER, T. K. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10*, 2 (1973), pp. 112–122.

[16] DRIRA, A.; PIERREVAL, H. AND HAJRI-GABOUJ, S. Facility Layout Problems: A Survey. *Annual Reviews in Control 31*, 2 (2007), pp. 255–267.

[17] FIALA, M. ARTag, a Fiducial Marker System Using Digital Techniques. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), CVPR '05, pp. 590–596.

[18] FIELD, D. A. Education and Training for CAD in the Auto Industry. *Computer-Aided Design 36*, 14 (2004), pp. 1431–1437.

[19] FITZMAURICE, G. W.; ISHII, H. AND BUXTON, W. A. S. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1995), CHI '95, pp. 442–449.

[20] GONZALEZ, R. C. AND WOODS, R. E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[21] GRAHAM, R. L. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. *Information Processing Letters 1*, 4 (1972), pp. 132–133.

[22] HOSOKAWA, T.; TAKEDA, Y.; SHIOIRI, N.; HIRANO, M. AND TANAKA, K. Tangible Design Support System Using RFID Technology. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction* (2008), TEI '08, pp. 75–78.

[23] IQBAL, M. AND HASHMI, M. S. J. Design and Analysis of a Virtual Factory Layout. *Journal of Materials Processing Technology 118*, 13 (2001), pp. 403–410.

[24] Ishii, H.; Ben-Joseph, E.; Underkoffler, J.; Yeung, L.; Chak, D.; Kanji, Z. and Piper, B. Augmented Urban Planning Workbench: Overlaying Drawings, Physical Models and Digital Simulation. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality* (2002), ISMAR '02, pp. 203–211.

[25] Ishii, H.; Kobayashi, M. and Arita, K. Iterative Design of Seamless Collaboration Media. *Communications of the ACM 37*, 8 (1994), pp. 83–97.

[26] Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1997), CHI '97, pp. 234–241.

[27] Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A. and Fitzgibbon, A. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (2011), UIST '11, pp. 559–568.

[28] Izadi, S.; Newcombe, R. A.; Kim, D.; Hilliges, O.; Molyneaux, D.; Hodges, S.; Kohli, P.; Shotton, J.; Davison, A. J. and Fitzgibbon, A. KinectFusion: Real-time Dynamic 3D Surface Reconstruction and Interaction. In *ACM Special Interest Group on Graphics and Interactive Techniques Talks* (2011), SIGGRAPH '11, p. 23.

[29] Jarvis, R.A. On the Identification of the Convex Hull of a Finite Set of Points in the Plane. *Information Processing Letters 2*, 1 (1973), pp. 18–21.

[30] Kagermann, H.; Wahlster, W. and Helbig, J., Eds. *Securing the Future of German Manufacturing Industry: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0, Final Report of the Industrie 4.0 Working Group*. Forschungsunion im Stifterverband für die Deutsche Wirtschaft e.V., 2013. `http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf` [Online; last accessed 22/08/15].

[31] Keil, J. M. and Snoeyink, J. On the Time Bound for Convex Decomposition of Simple Polygons. *International Journal of Computational Geometry and Applications 12*, 3 (2002), pp. 181–192.

[32] Kennedy, R. S.; Lane, N. E.; Berbaum, K. S. and Lilienthal, M. G. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology 3*, 3 (1993), pp. 203–220.

[33] Koch, V.; Kuge, S.; Geissbauer, R. and Schrauf, S. *Industry 4.0 – Opportunities and Challenges of the Industrial Internet*. strategy&, 2015. `http://www.strategyand.pwc.com/media/file/Industry-4-0.pdf` [Online; last accessed 22/08/15].

[34] KOLLENBERG, T.; NEUMANN, A.; SCHNEIDER, D.; TEWS, T.; HERMANN, T.; RITTER, H.; DIERKER, A. AND KOESLING, H. Visual Search in the (Un)Real World: How Head-mounted Displays Affect Eye Movements, Head Movements and Target Detection. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (2010), ETRA '10, pp. 121–124.

[35] LIN, E.; MINIS, I.; NAU, D. S. AND REGLI, W. C. Contribution to Virtual Manufacturing Background Research. *Institute for Systems Research, University of Maryland* (1995).

[36] MEISTERS, G. H. Polygons Have Ears. *The American Mathematical Monthly 82*, 6 (1975), pp. 648–651.

[37] MUJBER, T. S.; SZECSI, T. AND HASHMI, M. S. J. Virtual Reality Applications in Manufacturing Process Simulation. *Journal of materials processing technology 155*, 1 (2004), pp. 1834–1838.

[38] NEWCOMBE, R. A.; IZADI, S.; HILLIGES, O.; MOLYNEAUX, D.; KIM, D.; DAVISON, A. J.; KOHLI, P.; SHOTTON, J.; HODGES, S. AND FITZGIBBON, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality* (2011), ISMAR '11, pp. 127–136.

[39] NIELSEN, J. AND MOLICH, R. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI conference on Human factors in Computing Systems* (1990), CHI '90, pp. 249–256.

[40] OLSON, G. M. AND OLSON, J. S. Groupware and Computer-Supported Cooperative Work. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, Jacko, J. A. and Sears, A., Eds., The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Associates, Inc., 2003, pp. 583–595.

[41] PENTENRIEDER, K. *Augmented Reality based Factory Planning*. Ph.D Thesis, Technische Universität München, 2009.

[42] PIUMSOMBOON, T.; CLARK, A. AND BILLINGHURST, M. Physically-based Interaction for Tabletop Augmented Reality Using a Depth-sensing Camera for Environment Mapping. In *Proceedings of the 26th International Conference on Image and Vision Computing New Zealand* (2011), IVCNZ '11, pp. 161–166.

[43] RAUTERBERG, M.; FJELD, M.; KRUEGER, H.; BICHSEL, M.; LEONHARDT, U. AND MEIER, M. BUILD-IT: A Computer Vision-based Interaction Technique for a Planning Tool. In *Proceedings of HCI on People and Computers XII* (1997), HCI '97, pp. 303–314.

[44] ROHLING, G. Building a Common Framework. *Pictures of the Future*, Spring (2012), pp. 111–112.

[45] ROSENFELD, A. Connectivity in Digital Pictures. *Journal of the ACM 17*, 1 (1970), pp. 146–160.

[46] SCHUH, G.; POTENTE, T.; WESCH-POTENTE, C.; WEBER, A. R. AND PROTE, J. Collaboration Mechanisms to Increase Productivity in the Context of Industrie 4.0. *Procedia CIRP 19* (2014), pp. 51–56.

[47] SHAMOS, M. I. *Computational Geometry*. Ph.D Thesis, Yale University, 1978.

[48] SHARIATZADEH, N.; SIVARD, G. AND CHEN, D. Software Evaluation Criteria for Rapid Factory Layout Planning, Design and Simulation. *Procedia CIRP 3* (2012), pp. 299–304.

[49] SMARTF-IT CONSORTIUM. SmartF-IT: Cyber-physical IT Systems to Master Complexness of a New Generation of Multi-adaptive Factories, June 2013. `http://www.smartf-it-projekt.de/?lang=en` [Online; last accessed 22/08/15].

[50] SMITH, A. R. Color Gamut Transform Pairs. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques* (1978), SIGGRAPH '78, pp. 12–19.

[51] SUZAKI, K. *The New Manufacturing Challenge: Techniques for Continuous Improvement*. Free Press, 1987.

[52] SUZUKI, S. AND ABE, K. Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing 30*, 1 (1985), pp. 32–46.

[53] TOUSSAINT, G. T. Solving Geometric Problems with the Rotating Calipers. In *Proceedings of IEEE Mediterranean Electrotechnical Conference* (1983), MELECON '83.

[54] TUDDENHAM, P. J. *Tabletop Interfaces for Remote Collaboration*. Ph.D Thesis, University of Cambridge, 2009.

[55] UNDERKOFFLER, J. AND ISHII, H. Urp: A Luminous-tangible Workbench for Urban Planning and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1999), CHI '99, pp. 386–393.

[56] WANG, XIANGYU. Using Augmented Reality to Plan Virtual Construction Worksite. *International Journal of Advanced Robotic Systems 4*, 4 (2007), pp. 501–512.

[57] WILSON, A. D. Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *Proceedings of Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems* (2007), TABLETOP '07, pp. 201–204.

[58] WOLFF, I. AND SCHULZE, S. *Industrie 4.0 – Cyber Physical Systems in der Produktion*. Clustermanagement IKT.NRW, 2013. http://www.cps-hub-nrw.de/images/publish/download/Industrie40_CPS_in_der_Produktion-SchriftenreiheIKTNRW.pdf [Online; last accessed 22/08/15].

[59] ZUFFEREY, G.; JERMANN, P.; DO-LENH, S. AND DILLENBOURG, P. Using Augmentations As Bridges from Concrete to Abstract Representations. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (2009), BCS-HCI '09, pp. 130–139.