Title	A Study on Visualization Techniques of an AR-based Context-Aware Assembly Support System in Object Assembly
Author(s)	Bui, Minh Khuong
Citation	大阪大学, 2018, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/69716
rights	
Note	

A Study on Visualization Techniques of an AR-based Context-Aware Assembly Support System in Object Assembly

January 2018

Bui Minh Khuong

A Study on Visualization Techniques of an AR-based Context-Aware Assembly Support System in Object Assembly

A dissertation
submitted to the
Graduate School of Information Science and Technology
Osaka University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

January 2018

Bui Minh Khuong

Thesis Committee:

Prof. Haruo Takemura (Osaka University)
Prof. Takao Onoye (Osaka University)
Prof. Kiyoshi Kiyokawa (Nara Institute of Science and Technology)
Assoc. Prof. Tomohiro Mashita (Osaka University)
Assoc. Prof. Yuichi Itoh (Osaka University)

List of Publications

Journals

1. Bui Minh Khuong, Kiyoshi Kiyokawa, Andrew Miller, Joseph J. LaViola Jr, Tomohiro Mashita and Haruo Takemura: "Context-related Visualization Modes of an AR-based Context-Aware Assembly Support System in Object Assembly," Journal of the Virtual Reality Society of Japan, Vol.19, No.2, pp.195-205, 2014.

2. Bui Minh Khuong, Kiyoshi Kiyokawa, Tomohiro Mashita and Haruo Takemura: "Hybrid Object and Screen Stabilized Visualization Techniques for an AR Assembly Support System" Journal of the Virtual Reality Society of Japan, Vol.22, No.2, pp.167-175, 2017.

International Conferences

Peer-reviewed

1. Bui Minh Khuong, Kiyoshi Kiyokawa, Andrew Miller, Joseph J. LaViola Jr., Tomohiro Mashita, and Haruo Takemura: "The Effectiveness of an AR-based Context-Aware Assembly Support System in Object Assembly" Proc. of the IEEE Virtual Reality 2014, pp. 57-62, Mar., 2014.

Non-peer-reviewed

1. Bui Minh Khuong, Kiyoshi Kiyokawa, Tomohiro Mashita, Haruo Takemura: "A Study on an AR-based Toy-block Assembly Support System with Automatic Error Detection" Proc. 6th Korea Japan Workshop on Mixed Reality (KJMR), Okinawa, Japan, Apr., 2013.

Theses

- Bui Minh Khuong, Kiyoshi Kiyokawa, Haruo Takemura: "A Study on Context-Aware Assembly Support System using Augmented Reality" Master's Thesis, Graduate School of Information Science and Technology, Osaka University, March.,2013.

Abstract

This thesis proposes and evaluates the effectiveness of visualization techniques for an augmented reality (AR) - based context-aware assembly support system in object assembly. In assembly support systems using AR, visualization techniques of guidance information have a very important role because they affect directly to users perception, task performance accuracy and mental workload. Although many AR-based assembly support systems have been proposed, few keep track of the assembly status in real-time and automatically recognize error and completion states at each step. Naturally, visualization techniques and their effectiveness for such context-aware systems remain unexplored.

Our test-bed system was built in the context of building block (LEGO) assembly that can automatically recognize assembly status, detect assembly errors and display context-aware guidance information corresponding to the recognized assembly status. It also helps users correct errors quickly and finish assembly tasks correctly.

In our first evaluation, we compared the performance of the test-bed system in different AR visualization modes proposed with a traditional assembly instruction style - paper manual in assembly tasks. Experimental results show that although subjects took longer to complete the assembly tasks with the test-bed system, accuracy was dramatically improved and subjects also felt that the visualization modes proposed were easier to understand and more useful than the traditional assembly style with a paper manual.

In our second evaluation, based on feedbacks in the first evaluation, we proposed some new forms of the traditional AR visualization mode - Overlay mode and evaluated them in assembly tasks. We found a visualization mode (partial-wireframe overlay mode), which has guidance information and the topmost layer of the virtual model rendered directly overlaying on the real model, had better user preference as well as efficiency of assembly tasks.

We conducted the third evaluation to explore effectiveness of two modes: one is the best visualization mode proposed in the first evaluation (the side-by-side mode) and the second one is the mode we found in the second evaluation (the partial-wireframe overlay mode). Our experimental results indicate that the first mode outperforms the second one under moderate registration accuracy and marker-based tracking.

Although the side-by-side mode has good performance and user preference, it is not alway available due to limits of object-stabilized visualization styles specifically under the context we concern in this study with big size of assembling models and narrow field-of-view head mounted displays (HMDs). In the last evaluation, we proposed and evaluated the effectiveness of hybrid object- and screen-stabilized visualization techniques as a solution for the limits with the object-stabilized visualization styles. Our experimental results indicate one of the two hybrid object- and screen-stabilized visualization modes pro-

posed that shows virtual target status of real assembling objects at a fixed position on the HMD screen with real-time pose updated has the best performance and user preferences under the context considered in this study.

Our experiments showed that the visualization techniques proposed in this thesis helped users to have a better perception about assembly tasks, increase task performance accuracy and reduce mental workload. We believe that our results provide useful insight into the design of visualization techniques for AR-based assembly support systems under moderate registration accuracy and marker-based tracking context.

Acknowledgments

I'd like to thank everybody without whom this thesis would not have been possible:

First and foremost, I would like to thank my supervisor and direct-supervisor Prof. Haruo Takemura and Prof. Kiyoshi Kiyokawa for having given me the opportunity to study in the area of augmented reality. During my time in the Takemura Laboratory which has a wonderful research environment, I learned a lot from research life and gained many experiences. I want to thank them especially for all invaluable support and guidance to me.

I would like to express my sincere gratitude to my direct-supervisor Prof. Kiyoshi Kiyokawa, for the countless hours of revisions and advice on my work as well as being very supportive and understanding during a difficult time.

I'd also like to thank Assist.Prof. Tomohiro Mashita, all staff and students of the lab for their help.

I am grateful to my parents, my brother, my wife and my little daughter who always support and encourage me in my studies.

Bui Minh Khuong
Osaka University
January 2018

Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background

Augmented reality (AR) - the technology that blends computer generated virtual objects with the real environment, is becoming more and more widely used in many areas in our life. One of the most promising applications of augmented reality is in the traditional manufacturing assembly domain. In manufacturing, while some assembly operations are automated, there are still a significant number of assembly operations that require manual human effort. In terms of assembly support systems, AR technology makes it possible to display digital information in the assembly subject's field of view, such as step-by-step instructions that are essential for the work.

In order to provide more natural hand free interaction, support for more complex, multi-step assembly tasks, better assist for users as well as improve labor efficiency and accuracy, smarter assembly support systems such as context-aware systems should be examined and treated as a main research trend of AR application in manufacturing domain.

In an assembly support system using AR, visualization factors play a very important role. Visualization in AR has the potential to resolve spatial ambiguities by displaying spatial indicators (such as arrows or spotlights) properly registered and directly overlaying the actual workpiece, freeing the user from the cognitive burden of relating actual locations on the workpiece to corresponding locations on a separate virtual model [41]. In comparison to visualization of conventional systems such as paper-based work instructions or multimedia information systems, visualization in AR applications can display information depending on the context (i.e., in reference to particular components or sub-assemblies). This helps to reduce search time as well as head and eye movements [25] in assembly and is thus able to increase productivity [40].

1.2 Problems

Although a number of visualization techniques for assembly support system using AR have been proposed, there is still no design standard for AR based assembly support systems. This is due to a variety of display devices' specifications, constrains of tracking sensors, content and requirement of assembly tasks etc. For these reasons, identifying what information should be provided, what representation of the information, the effectiveness of the presentation in object assembly, etc., remain unexplored.

1.3 Proposals and Contributions

In the scope of this study we focus on usual conditions such as using head mounted display (HMD) devices with narrow field of view, marker-based environment, 3DOF tracking on the table with moderate registration accuracy, those are mostly used in assembly in practice. The goal of this study is to explore the best representation for guidance information of a context-aware assembly support system using AR which supports the best performance in assembly tasks.

In Chapter 1, we introduced the background of the study, problems, our proposed solutions as well as our findings and contributions for the research topic.

In Chapter 2, we did a survey on related works and highlighted the difference between our approaching with methods were introduced in the related works.

In Chapter 3, we described details about the test-bed assembly support system that we used in this study to evaluate our experiments.

In Chapter 4, we focus on our first attempt to suppress the impact of moderate registration accuracy to effectiveness of assembly tasks. We proposed two AR visualization modes, one visualizes a virtual object - a clone of the real object that is assembled whose structure and pose are updated in real-time to match those of the real object. This visualization mode is considered to be able to get rid of the effect of poor registration accuracy. Another visualization mode is displaying wire-frame virtual guidance information directly onto the real object. We conducted a comparative evaluation with a traditional assembly instruction style - paper manual in assembly tasks. Experimental results show that although subjects took longer to complete the assembly tasks with the test-bed system, accuracy was dramatically improved and subjects also felt that the visualization modes proposed were easier to understand and more useful than the traditional assembly style with a paper manual.

In Chapter 5, based on feedbacks in the first evaluation, we proposed some new forms of the traditional AR visualization mode - the overlay mode. In the first evaluation, participants encountered two problems with the full-wireframe overlay mode. The first one is low visibility of the real object due to a overlaid full-wirefame virtual content. The second one is the confusing visualization due to poor registration. Overlay mode is a traditional AR visualization style. Even though side-by-side mode had better performance in the first evaluation in Chapter 4. However, the two problems were encountered with the overlay mode mention above can be mitigated by improving the overlay mode. This gives us a hope to compare the improved overlay mode again to the side-by-side mode. Under bottom up assembly style which we considered in this study (a layer should be finished before starting a new layer above it), only a portion of the wire frame that connects the top part of the real object with guidance information (the partial-wireframe) will take better efficiency than displaying a full virtual connection wire frame between real object and virtual guidance information. The experimental results supported our hypothesis.

In Chapter 6, we described the third evaluation to explore effectiveness of two modes: one is the best visualization mode proposed in the first evaluation (the side-by-side mode) and the second one is an overlay mode with the best performance we found in the second evaluation (the partial-wireframe overlay mode). Our experimental results indicate the first mode outperforms the second one under moderate registration accuracy and marker-based tracking. It may still be the case that an overlay mode has the potential to reduce spatial ambiguity by overlaying instructions directly onto the real object, however it seems that this is highly sensitive to misalignment, latency, or conflicting depth cues. At least for our test-bed, having a spatial separation between the virtual model and the real model led to significantly better performance in every aspect.

In Chapter 7, we proposed and evaluated the effectiveness of hybrid object- and screen-stabilized visualization techniques as a solution for the limits with object-stabilized visualization styles. Although the best performance visualization style in the third evaluation,the side-by-side mode, has good performance and user preference, it is not always available due to limits of object-stabilized visualization styles specifically under the context we concern in this study with large assembling models and narrow field-of-view head mounted displays (HMDs). We conducted the fourth evaluation between the side-by-side mode and the hybrid modes proposed in this Chapter to evaluate their effectiveness in object assembly. Our experimental results indicate one of the two hybrid object- and screen-stabilized visualization modes proposed that shows virtual target status of real assembling objects at a fixed position on the HMD screen with real-time pose update has the best performance and user preferences under the context considered in this study.

In the Chapter 8, we summarized the findings, clarified and reinforced our hypotheses as well as gave the conclusions for usefulness of the study as well as contributions for the design of visualization techniques for AR-based assembly support systems.

Chapter 2
Related Works

2.1 Assembly Support Systems using Augmented Reality

Assembly support systems using augmented reality have been studied for many years. One of their main purpose is to make it possible to provide essential digital information in the assembly subject's field of view, such as step-by-step instructions in order to support and improve the effectiveness of their work. One of the most well-known assembly support systems in this regard is the assembly support system for cable harnesses at Boeing [13]. Engineers at Boeing have implemented wearable computers and augmented reality systems to aid workers in the assembly of airplanes (Figure 2.1). Their augmented reality project was designed to display pertinent instructions and diagrams in front of the manufacturing workers, who use the information to work on or assemble pieces of the aircraft. A wearable computer is used to render wire frame diagrams or text instructions at the arm's length in front of the user next to the work piece. The user looks at the piece and they see a diagram or text telling them what to do next and how. Since rendering complex images is not required, a wearable computer with low graphics capability suffices. One of the main challenges associated with using AR and a wearable system for this application is registering the overlaid instruction information relative to the work piece so that it stays on the target work piece precisely regardless of user motion. In order to solve this problem, Boeing engineers have worked on a real-time video-based tracker. A small, head-mounted video camera detects visual markers on the work piece and the computer estimates the relative pose information and displays the diagram and text on the work piece.

Since then, many research works on AR assembly guidance have been reported. Reiners et al. [45] developed an AR prototype system for assembling door locks on cars (Figure 2.2). The system was built using common off-the-shelf hardware, a standard SGI O2 with a 180 MHz R5k processor and 128MB memory. The machine has good video capabilities and reasonably fast rendering. A voice-command driven interface was used. It runs on a separate machine, a standard Intel-based laptop running Windows 95 and IBM VoiceType based speech recognition software. It is connected to the O2 via RS-232, which is adequate for the transmission of the short recognized commands. It uses CAD data taken directly from the construction and production database.

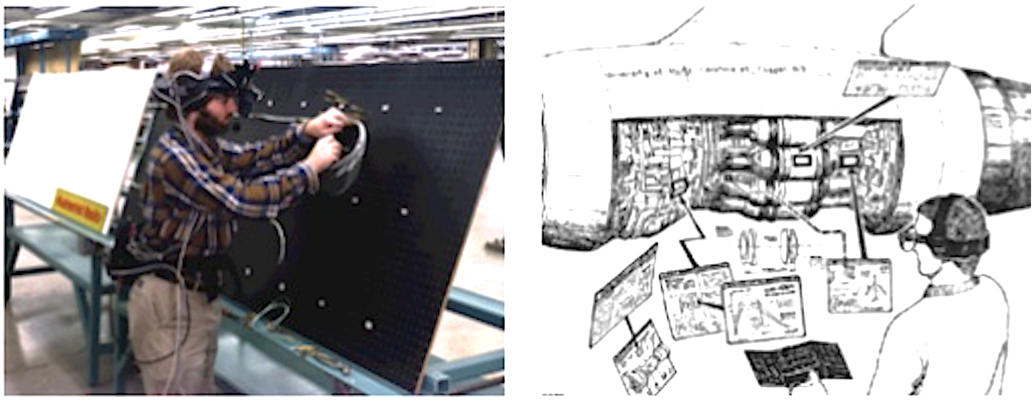


Figure 2.1: Aircraft assembly support system at Boeing [13]. Adam Janin demonstrates Boeing's prototype wire bundle assembly application.

This allows the system to be integrated into the existing infrastructure. An optical tracking system was designed and implemented using low cost passive markers that is fast enough for HMD use. Zauner et al. [57] developed a prototype system for AR in the assembling of furniture in which step-by-step instructions were given to users to help them complete the assembling (Figure 2.3). The system uses an HMD to display guidance information - a mixture of traditional 2D and 3D contents combining with audio to guide the user through the assembly process. ARToolKit [32] is used for tracking, and reference markers are attached to the various furniture parts thus allowing the system to determine the point and order of assembly. Salonen and Saaski [48] proposed AR assembly systems that focus on the implementation of an AR assembly system in a real setting in factory by integrating design for assembly (DFA) software tools to the design systems (CAD/PDM/PLM). The majority of the product data is created in design systems (CAD) and stored to a PDM/PLM system (Figure 2.4). The CAD model is exported to a standard STEP (ISO 10303 1994) format file, that includes a product structure and 3D models of the parts. The system uses markers and a HMD to display guidance information. The markers pasted on the rotating metal plate are used for tracking the physical units' orientation. Assembled parts are shown to the worker task by task according to the work phase instructions, overlaid on the physical unit. Syberfeldt et al. [54] described a study of using the concept of augmented reality for supporting assembly line workers in carrying out their task optimally. By overlaying virtual information onto real world objects and thereby enhance the human's perception of reality, and augmented reality makes it possible to improve the visual guidance to the workers. A prototype system is developed based on the Oculus Rift platform and evaluated using a simulated assembling task (Figure 2.5).

STARMATE, a project funded by the EU (IST-1999-10202), aims at speci-

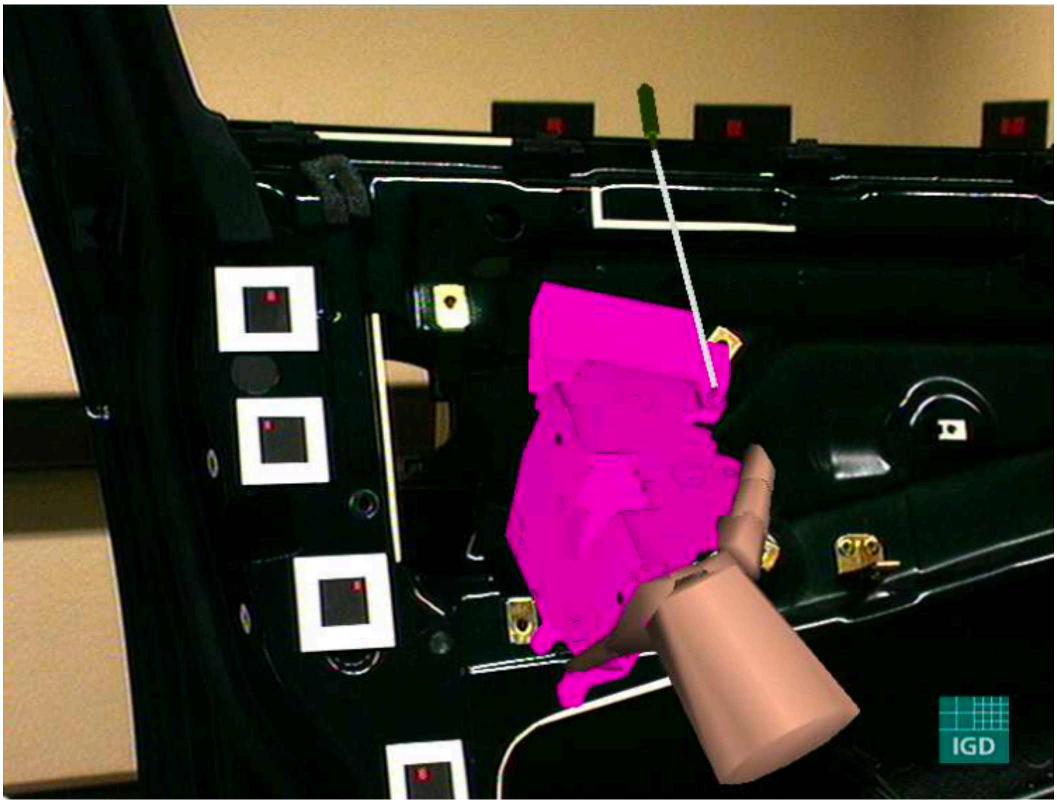


Figure 2.2: Car's door locks assembly using augmented reality [45].

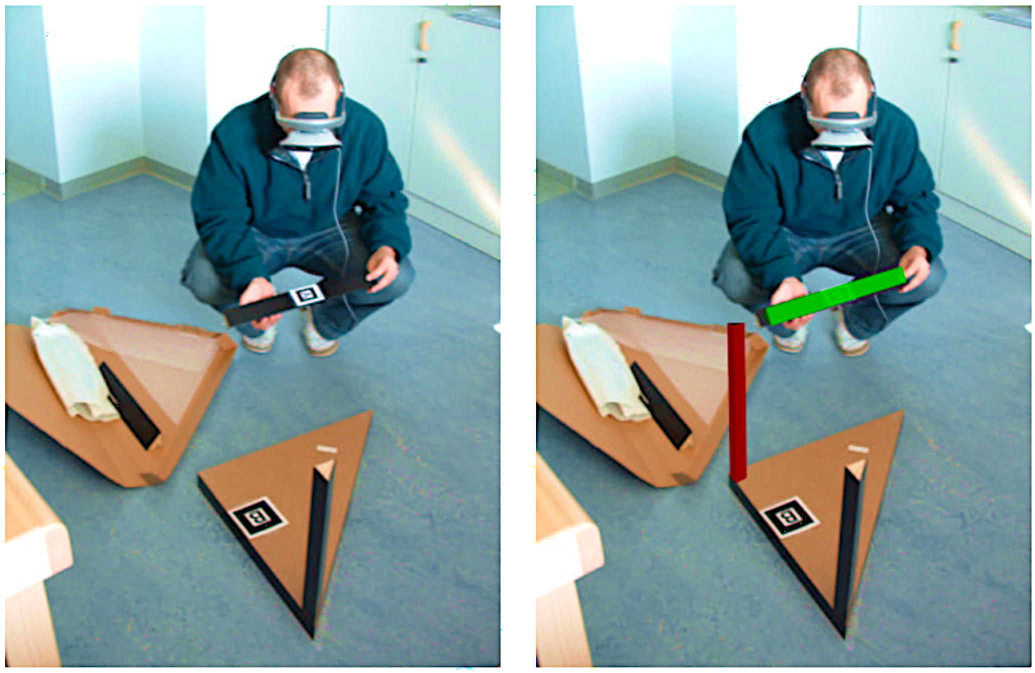


Figure 2.3: Furniture assembly using augmented reality [57].

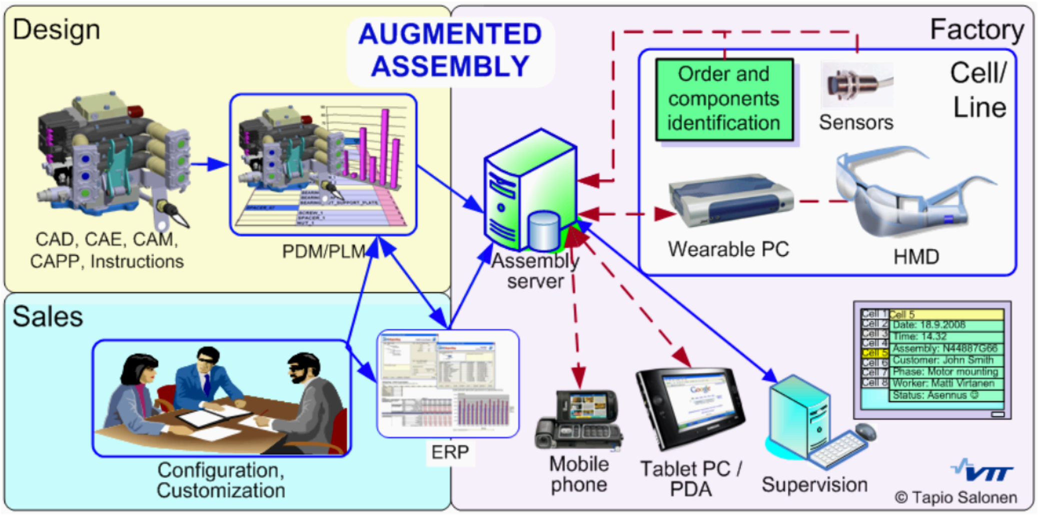


Figure 2.4: Salonen et al.'s proposed augmented reality based information processing architecture [48].

fying, designing, developing, and demonstrating a prototype dedicated to computer guided maintenance of complex mechanical elements using augmented reality techniques. The system will provide two main functionalities: User assistance for achieving assembly/disassembly and maintenance procedures, and workforce training of those procedures. A user equipped with a see-through HMD, a microphone and headphones, has to perform some tasks on a mechanical element [50]. The see-through mode of the HMD allows the user to see the real image of the scene, which is augmented by a computer generated virtual image, containing additional information (Figure 2.6) [50]. Tracking system consists of an optical infrared stereo tracker, which can be combined with an inertial tracker to overcome some weaknesses of a full optical solution.

In order to enhance man machine communication with more efficient and intuitive information presentation, Andersen et al. [12] proposed a proof-of-concept system based on stable pose estimation by matching captured image edges with synthesized edges from CAD models for a pump assembling process (Figure 2.7) [12]. The system is created for aiding a pump assembling process at Grundfos, one of the leading pump producers. Stable pose estimation of the pump is required in order to augment the graphics correctly. This is achieved by matching image edges with synthesized edges from CAD models. To ensure a system which operates at interactive-time the CAD models are pruned online and a two-step matching strategy is introduced. Online the visual edges of the current synthesized model are extracted and compared with the image edges using chamfer matching together with a truncated L2 norm. A dynamic visualization of the augmented graphics provides the user

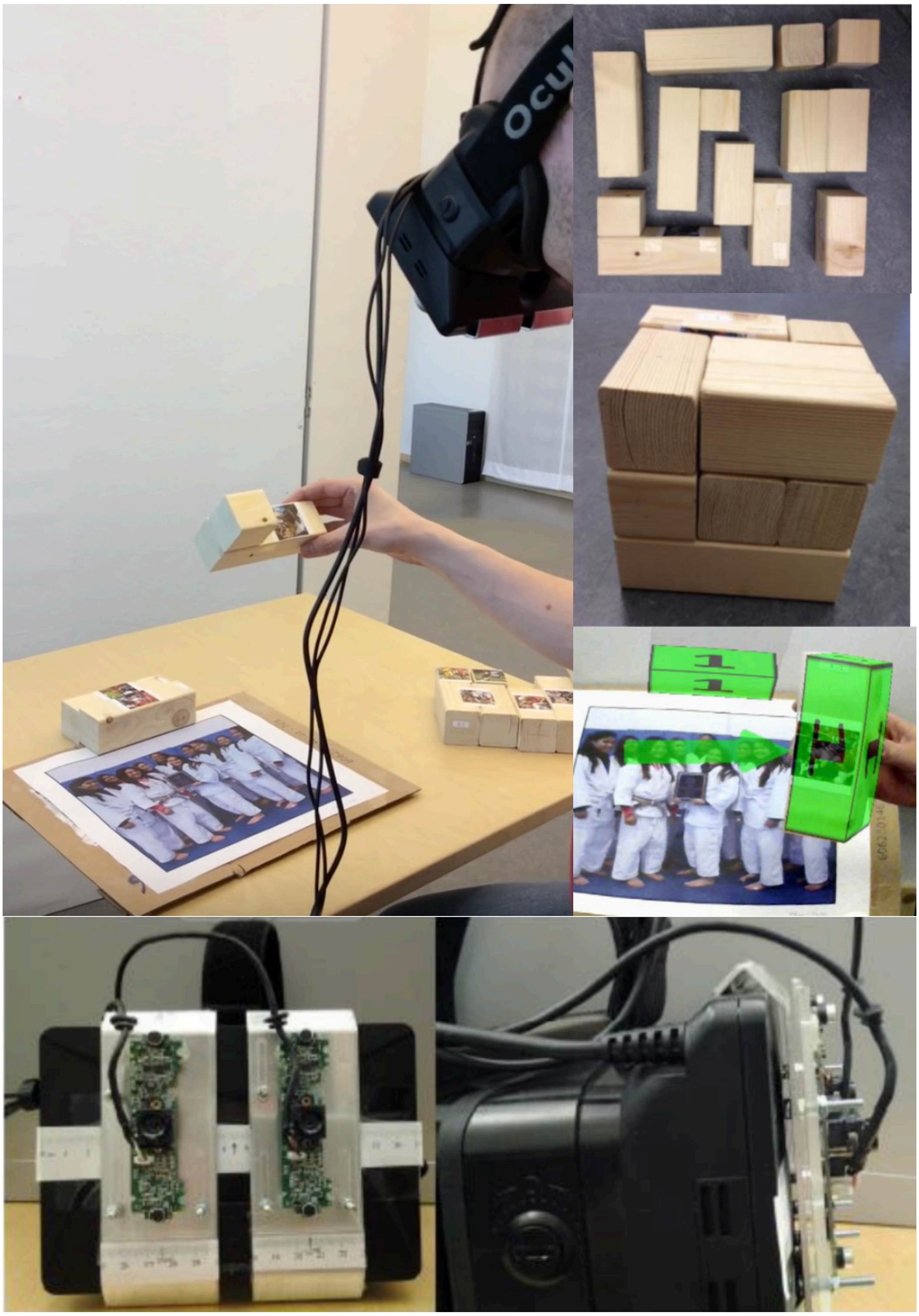


Figure 2.5: Visual assembly support system [54].

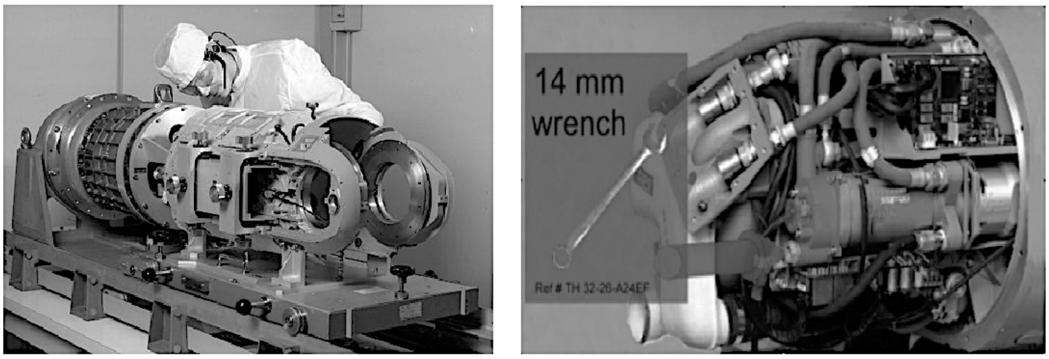
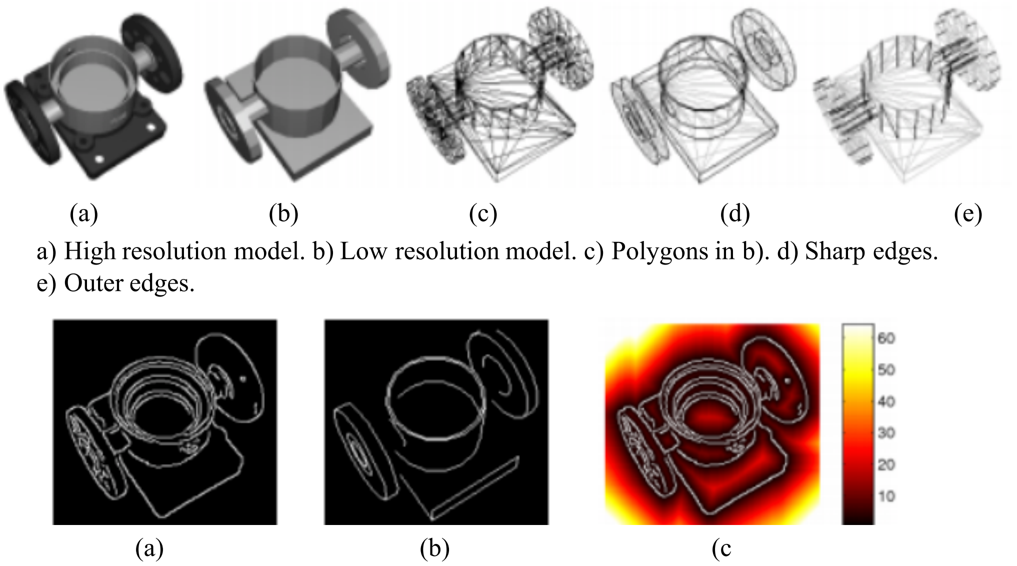


Figure 2.6: The STARMATE in use [50]. User working on a machine (left), and an augmented view of the user (right).

with guidance. Usability tests show that the accuracy of the system is sufficient for assembling the pump. Zhu et al. [60], proposed a wearable AR mentoring system to support assembly and maintenance tasks in industry specifically assist in maintenance and repair tasks of complex machinery, such as vehicles, appliances, and industrial machinery. The system combines a wearable optical see-through (OST) display device with high precision 6DOF pose tracking and a virtual personal assistant (VPA) with natural language, verbal conversational interaction, providing guidance to the user in the form of visual, audio and locational cues. The system is designed to be heads-up and hands-free allowing the user to freely move about the maintenance or training environment and receive globally aligned and context-aware visual and speech instructions (animations, symbolic icons, text, multimedia content, speech). The user can interact with the system, ask questions and get clarifications and specific guidance for the task at hand (Figure 2.8) [60].

Previous works have shown the potential of AR assembly guidance systems providing multi-media and interactive instructions to improve the performance of the users. However, limitations exist in current AR systems when assisting users with complex assembly processes. Such issues include time-consuming authoring procedures, integration with enterprise data, intuitive user interfaces, etc. Future work should examine the appropriateness of AR guidance for more complex, multi-step assembly tasks. In addition, although interactive AR assembly guidance has improved the traditional step-by-step guidance systems by providing pertinent information according to the user's requirement, the interaction scheme may disturb or interrupt the user's ongoing assembly task. Therefore, it is imperative to work on detecting and recognizing the users' actions or assembly status in order to provide more natural hand free interaction, better assist for users as well as improve labor efficiency and accuracy.



a) High resolution model. b) Low resolution model. c) Polygons in b). d) Sharp edges. e) Outer edges.

a) Detected edges. b) Synthesized edges. c). Distance transform with the detected edges superimposed.

Figure 2.7: Feature extraction and synthesis of objects assembling [12].

2.1.1 Context-Aware AR Assembly Support Systems

Smarter assembly support systems such as context-aware systems have been researched and become a main trend in recent years. ARVIKA (Figure 2.9) [2] project provides a context-sensitive system to enhance the real field of vision of a skilled worker, technician or development engineer with timely pertinent information. Figure 2.10 illustrates a sub-project of ARVIKA in aerospace industries.

ARTESAS (Advanced Augmented Reality Technologies for Industrial Service Applications) [1], a follow-up of ARVIKA project, aims to provide augmented reality systems to be used in complex industrial service situations. Using a see-through display, in a car assembly/disassembly scenario, the mechanic is provided with additional information about the mechanic parts in his or her view in a perspectively correct manner (Figure 2.11). 3D augmented information is displayed using a marker-less tracking system. Model-based tracking with CAD model data of the objects, real-time structure-from-motion (SfM) using a fish eye camera, inertial sensor which is used to measure head rotation have been used in combination (Figure 2.12).

COGNITO (Cognitive Assistance and Training System for Manual Tasks in Industry) whose main purpose is to design a personal assistance system, in which augmented reality is used to support users in task solving and manipulation of objects (Figure 2.13). Due to its sensing and learning capability, the COGNITO system automatically creates workflow references by observing a

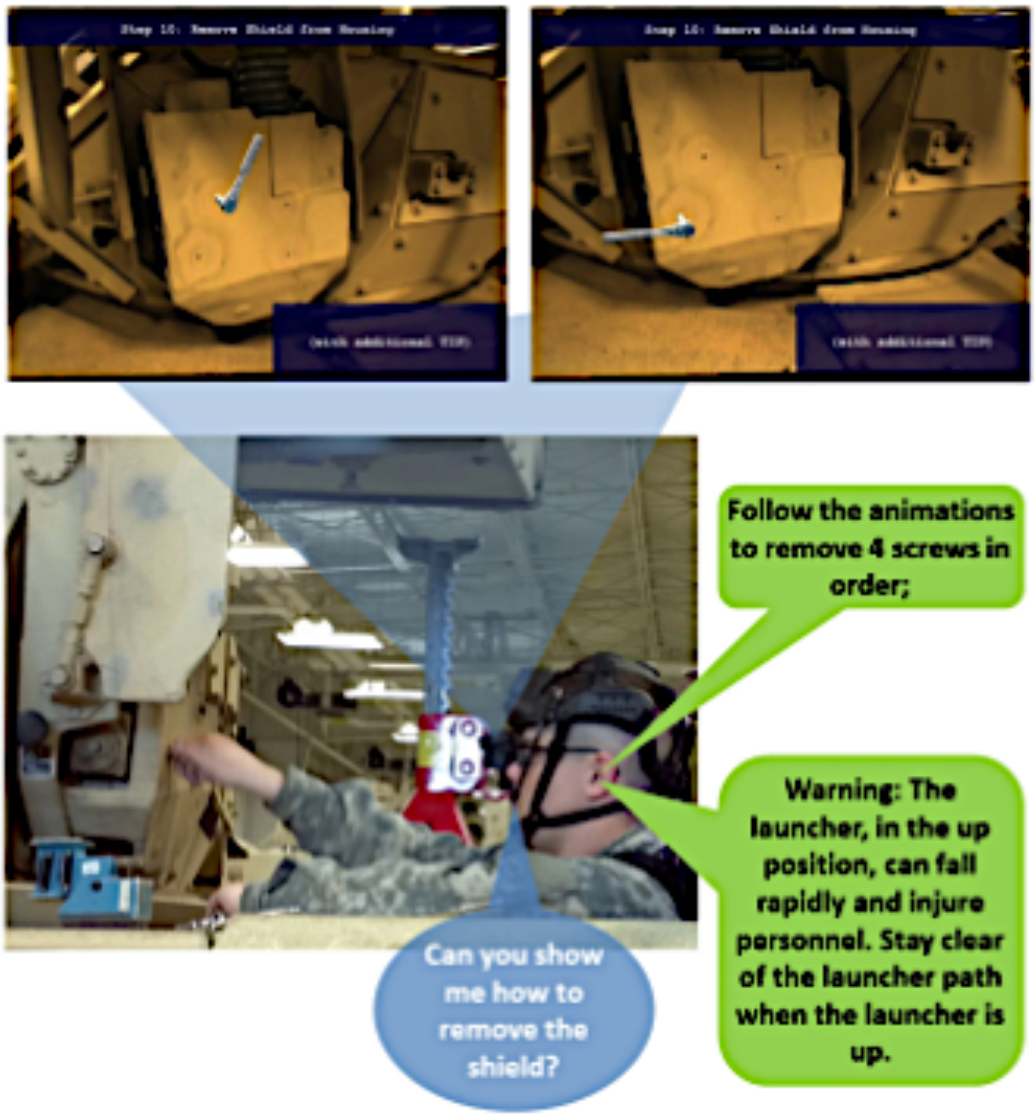

Step 10: Remove Shield from Housing
(with additional TIP)
Step 10: Remove Shield from Housing
(with additional TIP)
Follow the animations to remove 4 screws in order;
Warning: The launcher, in the up position, can fall rapidly and injure personnel. Stay clear of the launcher path when the launcher is up.
Can you show me how to remove the shield?

Figure 2.8: Concept of the proposed AR-Mentor system [60]: The user communicates verbally to the AR-Mentor system using a microphone; The AR-Mentor system understands the user and provides audible (speaker) and visual instructions (OST glasses).

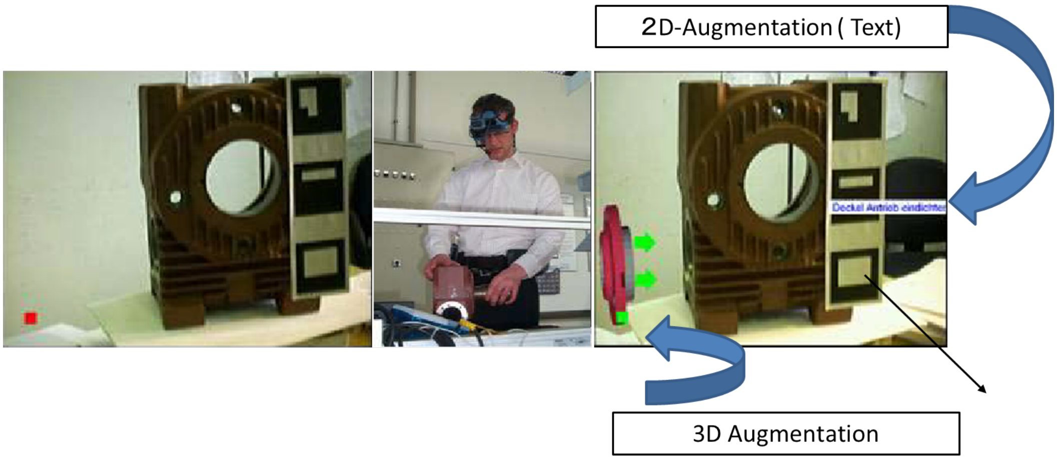


Figure 2.9: Step-by-step assembly augmentation in ARVIKA [2]. Using 2D computer graphics and 3D data models for the augmentation of the process.

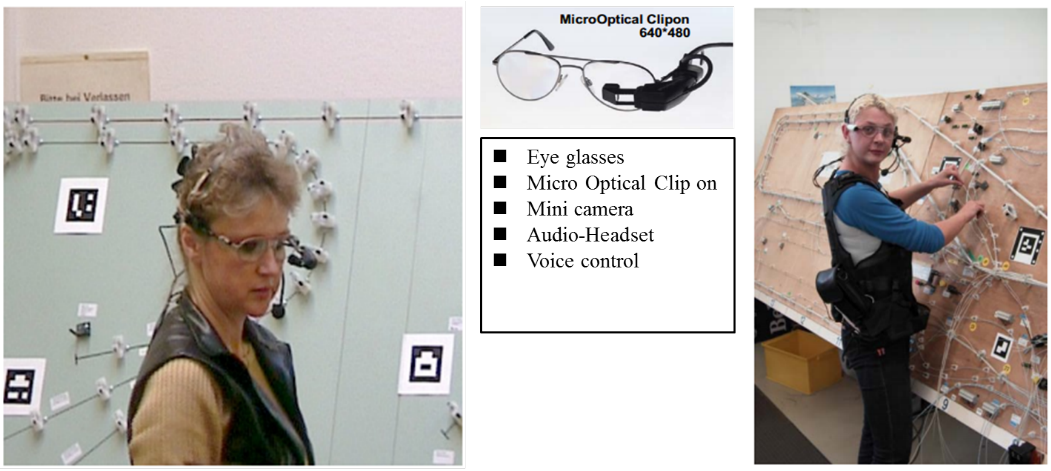


Figure 2.10: Sub-project of ARVIKA in aerospace industries [2].

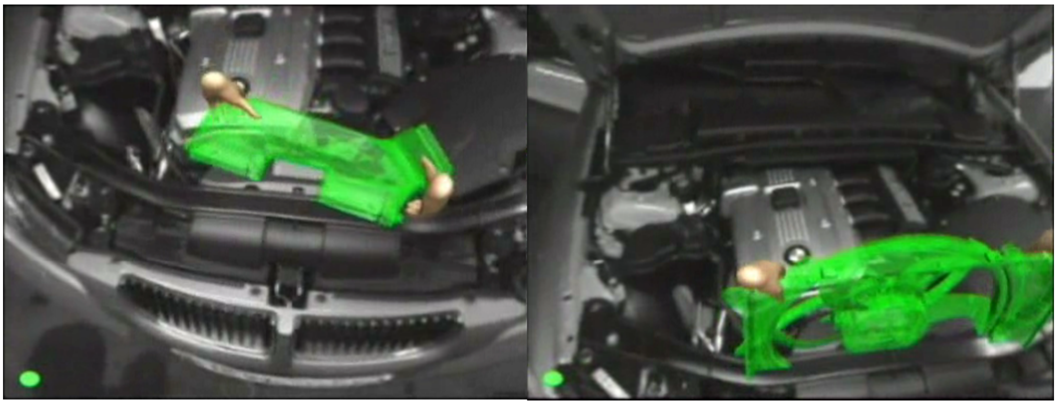


Figure 2.11: Demonstration in a car assembly/disassembly scenario in ARTE-SAS [1]. 3D augmentation information is displayed using marker-less tracking system.

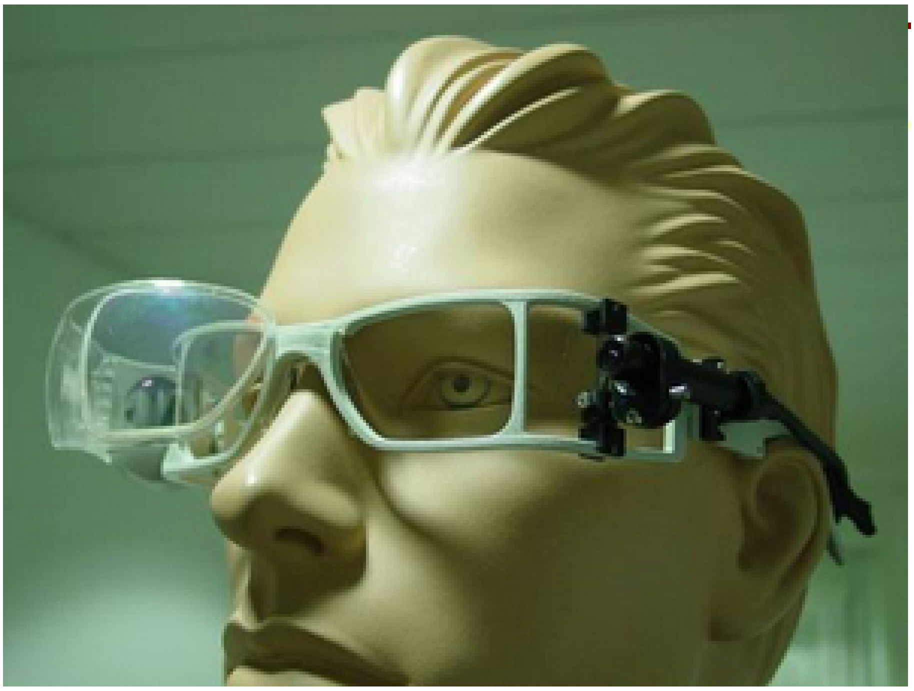


Figure 2.12: Structure-from-Motion system using a fish eye camera [1].

shown task in the learning mode. After a workflow has been learnt, the system can be run in the playback mode, in which it explains the previously learnt task to the operator. The system compares the user activity in real-time with the workflow reference and provides adequate feedback (Figure 2.14). The low level sensor processing handles the measurements from a BSN (on-body sensor network) and provides estimates of the positions of the operator, his or her hands, and relevant objects in the environment. A monocular HMD provides the system feedback and user assistance information.

In order to implement the context-aware AR systems, 3D models of workspace scenes are often required, whether for registration of the camera pose and virtual objects, handling of occlusion, or authoring of pertinent information. In our previous work [34] we introduced a test-bed system that displays guidance information and error detection information corresponding to recognized assembly status. It was used as the test-bed system for evaluations in this paper. We will explain more details about the test-bed system in the Chapter 3.

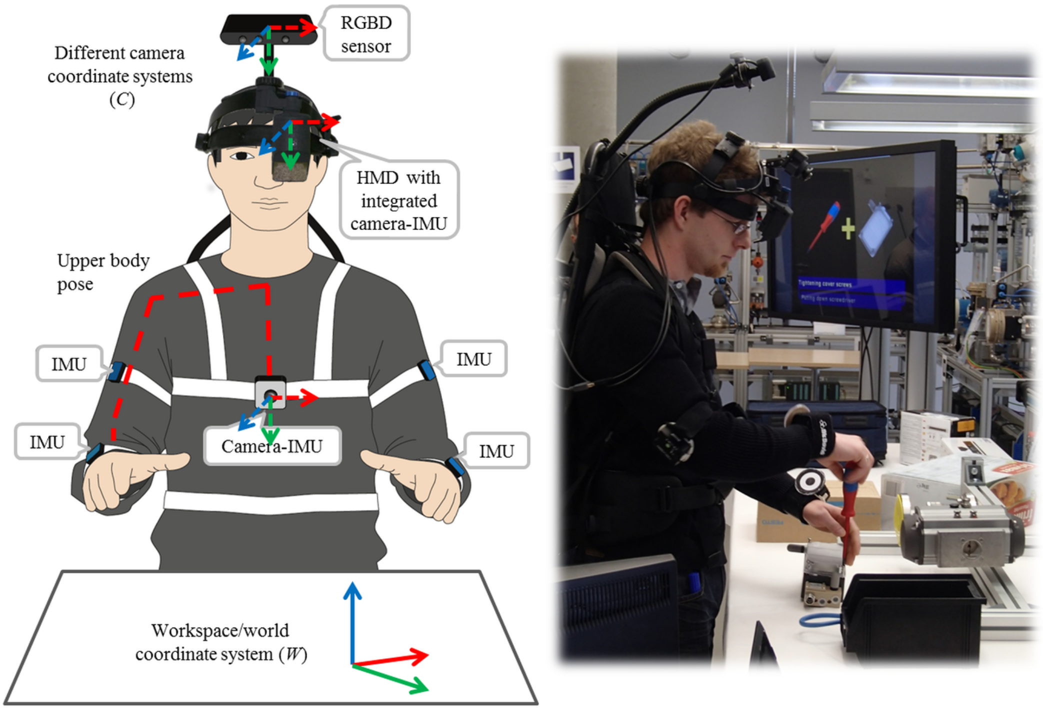

RGBD sensor
Different camera coordinate systems (C)
HMD with integrated camera-IMU
Upper body pose
IMU
IMU
IMU
IMU
Camera-IMU
Workspace/world coordinate system (W)

Figure 2.13: A usage scenario in the COGNITO system [3].

2.2 Typical Architecture of an AR Assembly Support System

2.2.1 Generalized AR System Design Concepts

In 2005, Bimber and Raskar [16] proposed a generalized design for an AR-based assembly support system (Figure 2.15).The generalized design is comprised of the following four layers of components:

Base Level. This is the lowest level of design, and it includes hardware and software designs for tracking objects in the real world, displaying objects and information to the user. Most research effort to date in the field of AR has occurred at this level.

Intermediate Level. This level of design, implemented mostly in software, is responsible for interacting with the user, providing authoring tools for creating AR content, presenting and arranging generated content.

Application Level. This level, implemented entirely in software, consists of the overarching AR application. It serves as the primary interface to the user.

User Level. This level represents the end user' system. This level is included in the generalized design to emphasize the users role in the application. AR applications are highly dependent on human design issues, which must be considered independently of aggregate interface design approaches.

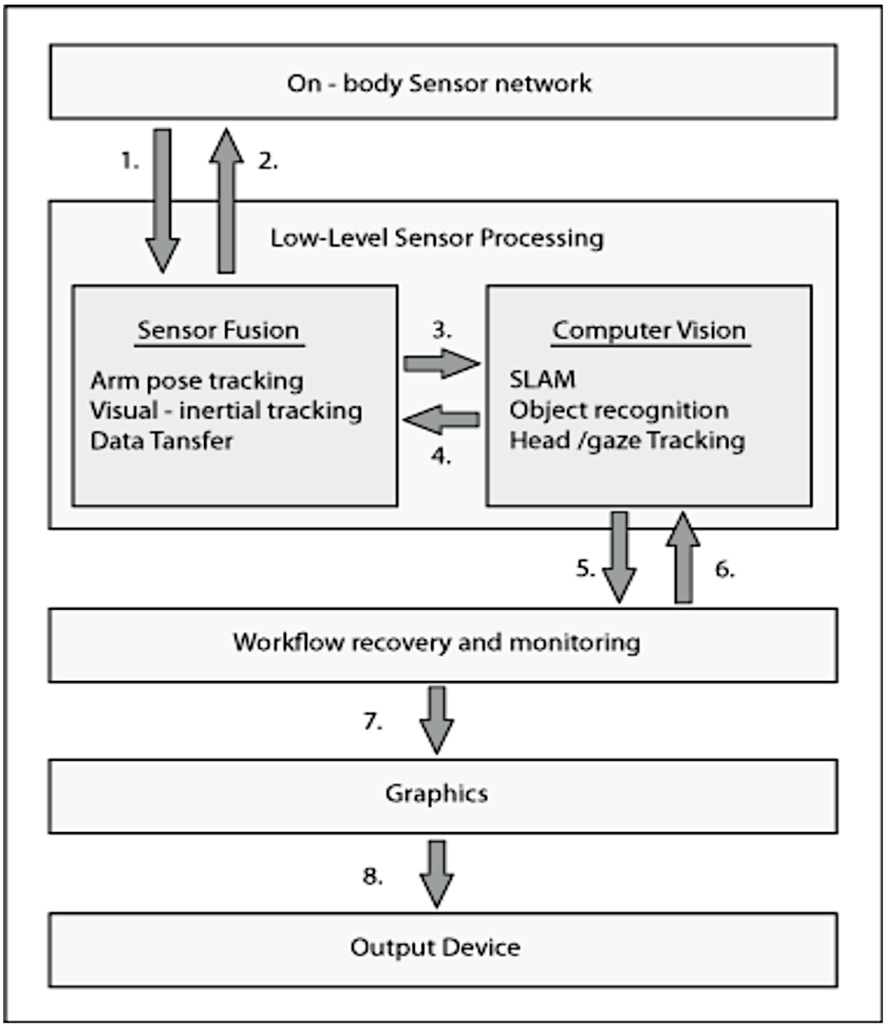


Figure 2.14: Workflow of the COGNITO system [3].

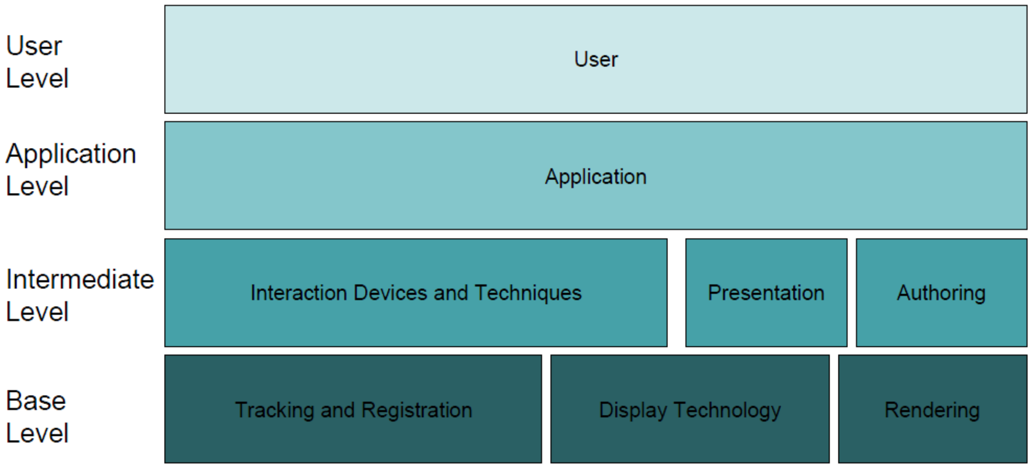


Figure 2.15: Generalized AR Design [16].

Based on the generalized design, a typical AR assembly system is illustrated in Figure 2.16 [55]. There are six function modules, namely, video capture, image analysis and processing, tracking process, interaction handling, assembly information management and rendering are illustrated as the kernel modules to constitute the main loop of an AR assembly system. In addition, for each important function module, e.g., display or camera, there is a pop-up note to illustrate its characteristics, disadvantages, classifications, etc. Important pathways for data transferring on which the six kernel modules rely on are shown in color, e.g., input data pathways are in blue, while output data pathways are in orange.

2.2.2 Components for an Assembly Support System

As above, there are six function modules are kernel modules of an AR based assembly support system. Among them, the most essential components are tracking, displaying and content authoring.

2.2.2.1 Tracking

Tracking refers to the process of continuously determining an object's position and orientation in physical space. This fundamental task, inherent to all AR applications, provides important information for determining the state of objects (including the user) and their interactions. Tracking real world objects is vital to registering virtual objects with their real-world counterparts. Tracking techniques generally fall into three categories: sensor-based, vision-based, and hybrid tracking techniques [59].

Sensor-based tracking techniques are based on sensors such as mechanical, inertial, electromagnetic and ultrasonic sensors. *Mechanical tracking* systems

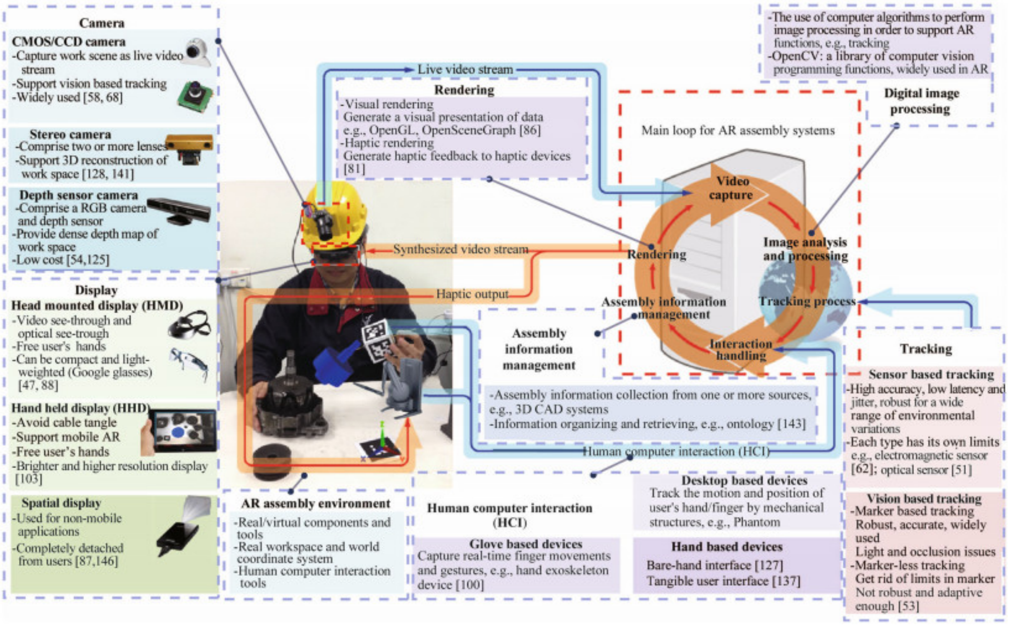

Camera
CMOS/CCD camera
-Capture work scene as live video stream
-Support vision based tracking
-Widely used [58, 68]
Stereo camera
-Comprise two or more lenses
-Support 3D reconstruction of work space [128, 141]
Depth sensor camera
-Comprise a RGB camera and depth sensor
-Provide dense depth map of work space
-Low cost [54,125]
Display
Head mounted display (HMD)
-Video see-through and optical see-trough
-Free user's hands
-Can be compact and light-weighted (Google glasses) [47, 88]
Hand held display (HHD)
-Avoid cable tangle
-Support mobile AR
-Free user's hands
-Brighter and higher resolution display [103]
Spatial display
-Used for non-mobile applications
-Completely detached from users [87,146]
Live video stream
Rendering
-Visual rendering
Generate a visual presentation of data e.g., OpenGL, OpenSceneGraph [86]
-Haptic rendering
Generate haptic feedback to haptic devices [81]
Synthesized video stream
Haptic output
-The use of computer algorithms to perform image processing in order to support AR functions, e.g., tracking
-OpenCV: a library of computer vision programming functions, widely used in AR
Digital image processing
Main loop for AR assembly systems
Video capture
Image analysis and processing
Tracking process
Interaction handling
Assembly information management
Rendering
Assembly information management
-Assembly information collection from one or more sources, e.g., 3D CAD systems
-Information organizing and retrieving, e.g., ontology [143]
Human computer interaction (HCI)
Tracking
Sensor based tracking
-High accuracy, low latency and jitter, robust for a wide range of environmental variations
-Each type has its own limits e.g., electromagnetic sensor [62]; optical sensor [51]
Vision based tracking
-Marker based tracking
Robust, accurate, widely used
Light and occlusion issues
-Marker-less tracking
Get rid of limits in marker
Not robust and adaptive enough [53]
AR assembly environment
-Real/virtual components and tools
-Real workspace and world coordinate system
-Human computer interaction tools
Human computer interaction (HCI)
Glove based devices
Capture real-time finger movements and gestures, e.g., hand exoskeleton device [100]
Desktop based devices
Track the motion and position of user's hand/finger by mechanical structures, e.g., Phantom
Hand based devices
Bare-hand interface [127]
Tangible user interface [137]

Figure 2.16: Typical architecture of AR assembly system [55].

use mechanical linkages tied to potentiometers and other electronic devices to determine position and orientation. These systems provide extremely accurate position and orientation information, often serving as the baseline for tracker performance tests, and were used in early AR systems including the first system developed by Sutherland [53]. However, a principal disadvantage of these trackers is the need for the mechanical apparatus that supports linkages and sensors, constraining the user's freedom of motion. *Inertial tracking* systems, which leverage the principle of linear and angular momentum conservation, represent an important tracker class, given their compact size and ability to function without an external source. However, since these systems tend to drift over relatively short periods of time on their own, they are most useful when configured as part of hybrid systems. *Electromagnetic tracking* systems employ sensors that determine their position and orientation inside an artificially generated electromagnetic field. These systems are relatively inexpensive and provide relatively high accuracy. However, these trackers are highly susceptible to interference from ferrous metal, and magnetic fields, especially in dynamic environments in which potential sources of interference are not stationary. *Ultrasonic tracking* systems calculate position by measuring the time ultrasonic signals from a set of transducers take to reach a set of microphones, assuming the speed of sound to be known. These systems are also relatively inexpensive and provide a high level of accuracy, for example, the 3D position and orientation of a set of three rigidly mounted microphones can be determined based on the computed time of flight (and hence distance) to each microphone from each of a set of three rigidly mounted transducers.

Ultrasonic systems were among the earliest trackers used in AR research, including Sutherland's. However, they require a direct line of sight, can suffer from interference from echoes and other sources, and have a relatively low update rate. The only sufficiently high-quality systems currently available that use ultrasonic technology do so in combination with other technologies to address these deficiencies, and will therefore be discussed below with other hybrid trackers. [47] provides a good review of sensor based tracking. Sensor-based tracking techniques are widely used in many advanced assembly support systems such as ARTESAS [1] (inertial sensors) or COGNITO [3] (inertial sensors).

Vision-based tracking techniques can use image processing methods to calculate the camera pose relative to real-world objects. In computer vision, most of the available tracking techniques can be divided into two classes, feature-based and model-based [44]. The rationale underlying feature-based methods is to find a correspondence between 2D image features and their 3D world frame coordinates. The camera pose can then be found from projecting the 3D coordinates of the feature into the observed 2D image coordinates and minimizing the distance to their corresponding 2D features [56]. One of the earliest feature-based tracking techniques is marker tracking method that could be used to calculate camera pose in real-time from artificial markers. The popular ARToolKit library [32], an open-source framework, allows developers to define their own marker designs and train the tracking system to recognize them in the environment. Other studies explored tracking from efficient line finding methods [52] or non-square visual markers [21, 39]. Since marker-based tracking requires that markers be placed in the environment in advance and remain sufficiently visible during a tracked task, research in optical tracking has attempted to replace printed markers with natural features [17, 23] (such as visually unique parts of an engine), making marker-less tracking possible.

Model-based tracking methods [17, 43] can capitalize on the natural features existing in the environment and thus extend the range of the tracking area using natural features which are relatively invariant to illuminations. For example, edges are the most frequently used features as they are relatively easy to find and robust to changes in lighting. However, model-based methods also usually require the cumbersome process of modeling, especially when creating detailed models for a large cluttered environment.

For some AR applications, computer vision alone cannot provide a robust tracking solution and so hybrid methods have been developed which combine several sensing technologies. A *hybrid tracking* method can be a combination among inertial and computer vision technologies. Vision-based tracking has low jitter and no drift, but it is slow and outliers can occur. Inertial tracking is fast and robust and can be used for motion prediction when rapid changes occur. Klein and Drummond [35] presented a hybrid visual tracking system, based on a CAD model of edge and a serving system. It uses rate gyroscopes to track the rapid camera rotations. Bleser et al. [18] presented a hybrid approach combining structure from motion, SLAM (simultaneous localization and mapping) and model-based tracking. A CAD model was first used for initialization and then 3D structure was recovered during the tracking allowing the camera to move away and explore unknown parts of the scene. Hybrid methods are also widely used in the advanced assembly support systems such as ARTESAS or COGNITO.

2.2.2.2 Displaying

The display technologies mainly focus on three types: see-through head mounted displays, projection-based displays and handheld displays. In the assembly support system, see-through HMDs are mostly employed to allow the user to see the real world with virtual objects superimposed on it under the user's field of view. Virtual objects are displayed in front of one eye (*monocular*) or both eyes of the user (*biocular* if the images presented to both eyes are the same, *binocular* if the images presented to both eyes form a stereo pair). See-through HMDs may be fundamentally divided into two categories; optical see-through (OST) and video see-through (VST) HMDs [59]. Optical see-through displays allow the user to see the real world with their natural eyes. The real and virtual worlds are merged using optical combiners, such as half-silvered mirrors or prisms. Video see-through displays use cameras to capture real world imagery, combine the real and virtual content digitally or through video mixing hardware, and present it on the same displays. The assembly support system in STARMATE [51] uses a pair of semi-transparent goggles (HMD) to allow the user to see the real scenes and display visual elements over them. In ARVIKA [2], the assembly support system uses a MicroOptical Clip-on display (640 × 480) to show the augmentation information. In COGNITO [3], the assembly support system uses a monocular optical see-through view-up HMD to provide the system feedback and user assistance information.

Optical see-through displays seem to be used more often than video see-though displays in the assembly support systems. Obviously, optical see-through displays have the advantage of presenting the real world at its full spatial resolution, with no temporal lag, full stereoscopy, and no mismatch between vergence (the angle between the lines of sight from each eye to a given real-world object) and accommodation (the distance at which the eyes must focus to perceive that real world object). However, they have some limitations. Other input devices such as cameras are required for interaction and registration. Furthermore, combining the virtual objects holographically through transparent mirrors and lenses creates disadvantages as it reduces brightness and contrast of both the images and the real-world perception. In addition, occlusion (or mediation) of real objects is difficult because their light is always combined with the virtual image.

In contrast, video see-through displays have the advantage of being the cheapest and easiest to implement. Since reality is digitized, it is easier to mediate or remove objects from reality. This includes removing or replacing fiducial markers or placeholders with that of virtual objects. Brightness and contrast of virtual objects are matched easily with the real environment. Furthermore, the digitized images allow tracking of head movement for better registration. It also becomes possible to match perception delays of the real and virtual. VST displays can handle occlusion problems more easily compared to OST displays due to various image processing techniques.

Recently, head mounted projective displays (HMPDs) [30] have been used as an alternative to HMDs. They typically use a pair of miniature projectors mounted on the head that project images onto retro-reflective material in the real environment which is then reflected into the user's eyes. The main advantages of HMPDs compared to HMDs are that they can support a large field of view (up to around 90°), that they allow easier corrections of optical distortions, and that they provide the ability to project undistorted images onto curved surfaces. However, light from an HMPD needs to pass through the display optics several times, which can cause a reduction in image brightness. Furthermore, Normal spatial AR systems typically assume that all virtual material is intended to lie on the projected surface, limiting the type of geometry that can be presented.

2.2.2.3 Content Authoring

Augmentation information for assembly instructions used in assembly support systems can be in various types (text information, 2D images, 3D objects, etc.). In the assembly support system at Boeing, text, wire frames and simple images were used for assembly instructions. In STARMATE [51], ARVIKA [2], ARTESAS [1], 2D and 3D augmentation are used for assembly instructions.

At the early period, content development was performed mainly at source-code level using programming languages such as C or C++. Currently, there are many solutions, frameworks, authoring tools dedicated to augmented reality, aimed at supporting the creation of AR contents. Some of the available AR authoring tools include the CREATE tool from Information in Place [4], the DART toolkit [5], and the MARS Authoring Tool [8]. Companies like Thinglab [9] assist in 3D scanning or digitising of objects. Recently, Kinect-Fusion [31] enables a user holding and moving a standard Kinect camera to rapidly create detailed 3D reconstructions of an indoor scene as well as individual objects by segmentation through direct interaction. It can be used as a low-cost object scanner, and the reconstructed 3D objects can be imported to CAD or other 3D modeling applications, or they can even be materialized by a 3D printer. Optical capture systems, capture suits, and other tracking devices available at companies such as Inition [6] can be tools for creating AR content beyond "simple" annotations.

Another approach of AR content authoring in the manufacturing industry

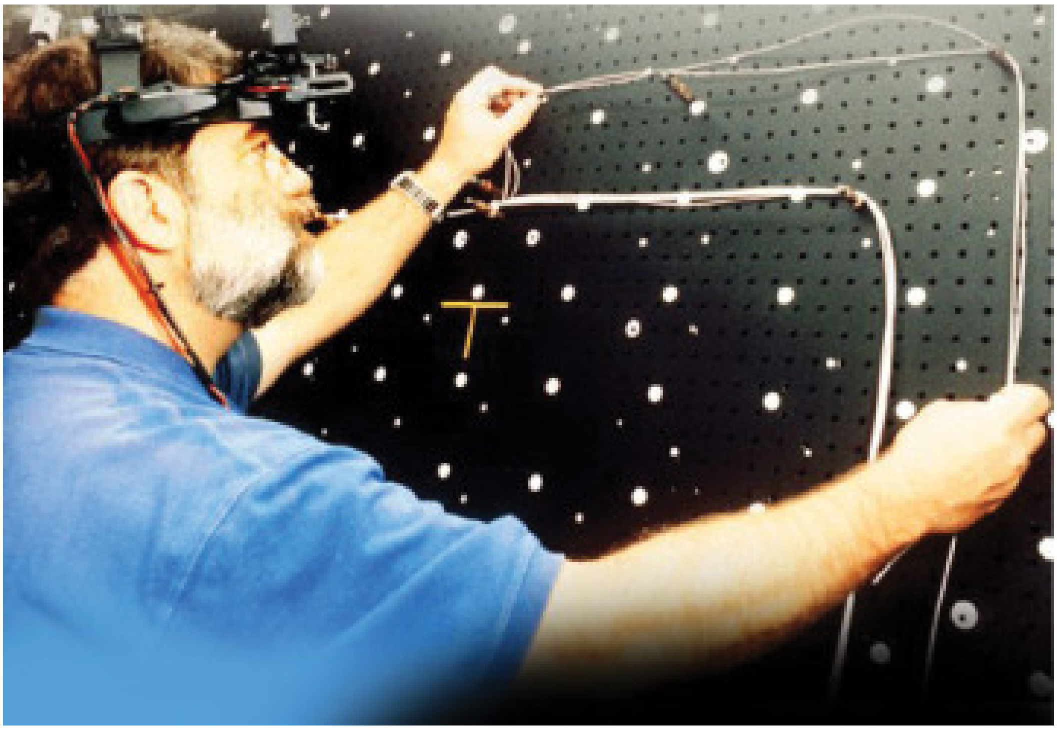


Figure 2.17: A small head-mounted video camera to display a diagram and text on the workpiece [11] .

is reuse of existing product data. Product data is stored in CAD / PDM / PLM systems. These systems include all relevant product data (3D geometry, product structures, simulation results, part fabrication plans, assembly plans etc.). The information can be retrieved from CAD / PDM / PLM systems, which can be then converted to some forms suitable for AR displays. [49] proposed methodology how the AR instructions are created from the product's 3D model.

2.3 Visualization Techniques for Assembly Support Systems

There has also been a rich body of work on visualization techniques for assembly support systems using AR. Caudell and Mizell [11] proposed one of the first implementation of a classic AR assembly system by combining head position sensing and real world registration with the HMD, such that a computer-produced diagram, containing pertinent information, can be superimposed and stabilized on a specific position on a real-world object (Figure 2.17).

Reiners [45] and his colleagues demonstrated a prototype AR system that uses passive retro-reflective markers illuminated by IR sources to augment a mechanic's natural view with text, labels, arrows, and animated sequences designed to facilitate task comprehension, location, and execution. Translational and rotational animations of visual graphics in the world coordinates allow for those graphics to convey additional meaning in an intuitive way such as the insertion of one part into another, or twisting of a tool, etc. Zhang et al. [58] proposed a method to implement the RFID technology in the application of assembly guidance in an augmented reality environment, aiming at providing just-in-time information rendering and intuitive information navigation for the assembly operator. Henderson and Feiner [26] presented the first AR system to aid users in the psychomotor phase of procedural tasks. The system provides dynamic and prescriptive instructions in response to the user's on-going activities.

A basic design decision in building an AR system is how to accomplish the combining of real and virtual. Two basic choices for AR visualization are available: optical and video technologies. A see-through HMD is one device used to combine real and virtual. Standard closed-view HMDs do not allow any direct view of the real world. In contrast, a see-through HMD lets the user see the real world, with virtual objects superimposed by optical or video technologies.

Optical see-through HMDs work by placing optical combiners in front of the user's eyes. These combiners are partially transmissive, so that the user can look directly through them to see the real world. The combiners are also partially reflective, so that the user sees virtual images bounced off the combiners from head-mounted monitors. This approach is similar in nature to Head-Up Displays (HUDs) commonly used in military aircraft, except that the combiners are attached to the head. Figure 2.18 shows a conceptual diagram of an optical see-through HMD. Figure 2.19 shows an optical see-through HMD made by Seiko Epson Corporation.

In contrast, video see-through HMDs work by combining a closed-view HMD with one or two head-mounted video cameras. The video cameras provide the user's view of the real world. Video from these cameras is combined with the graphic images created by the scene generator, blending the real and virtual. The result is sent to the monitors in front of the user's eyes in the closed-view HMD. Figure 2.20 shows a conceptual diagram of a video see-through HMD. Figure 2.21 shows an actual video see-through HMD, Wrap1200AR made by Vuzix.

AR systems can also be built using monitor-based configurations, instead of see-through HMDs. Figure 2.22 shows how a monitor-based system might be built. In this case, one or two video cameras view the environment. The cameras may be static or mobile. In the mobile case, the cameras might move around by being attached to a robot, with their locations tracked. The video of the real world and the graphic images generated by a scene generator are combined, just as in the video see-through HMD case, and displayed in a monitor in front of the user. The user does not wear the display device. Figure 2.23 and Figure 2.24 shows an actual monitor-based AR system and

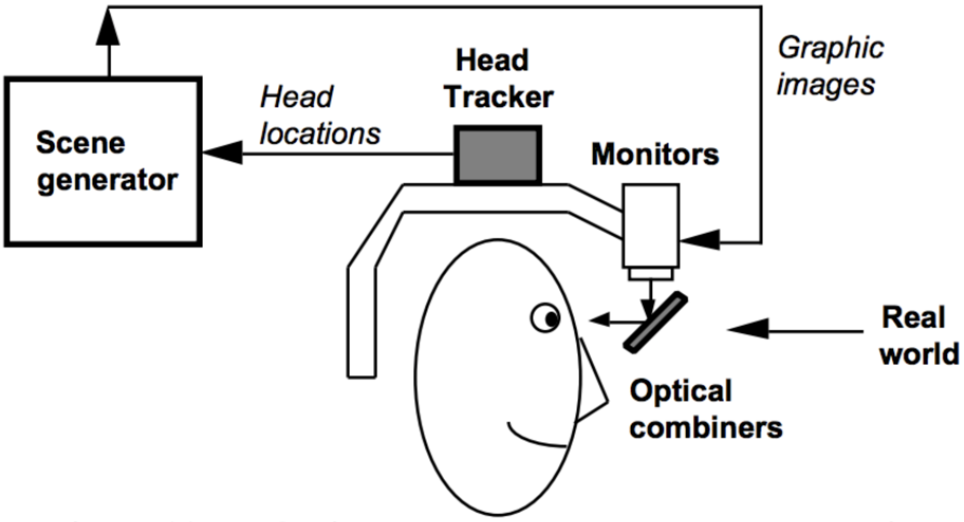


Figure 2.18: Optical see-through HMD conceptual diagram [13].

an AR system on a smart phone.

In this study we chose video see-through HMD approach due to following advantages over optical blending:

Real and virtual view delays can be matched: Video offers an approach for reducing or avoiding problems caused by temporal mismatches between the real and virtual images. Optical see-through HMDs offer an almost instantaneous view of the real world but a delayed view of the virtual. This temporal mismatch can cause problems. With video approaches, it is possible to delay the video of the real world to match the delay from the virtual image stream.

Additional registration strategies: In optical see-through, the only information the system has about the user's head location comes from the head tracker. Video blending provides another source of information: the digitized image of the real scene. This digitized image means that video approaches can employ additional registration strategies unavailable to optical approaches.

Easier to match the brightness of real and virtual objects: Ideally, the brightness of the real and virtual objects should be appropriately matched. Unfortunately, in the worst case scenario, this means the system must match a very large range of brightness levels. The eye is a logarithmic detector, where the brightest light that it can handle is about eleven orders of magnitude greater than the smallest, including both dark adapted and light-adapted eyes. In any one adaptation state, the eye can cover about six orders of magnitude. Most display devices cannot come close to this level of contrast. This is a particular problem with optical technologies, because the user has a direct view of the real world. If the real environment is too bright, it will wash out the virtual image. If the real environment is too dark, the virtual image will wash out the real world. Contrast problems are not as severe with video,

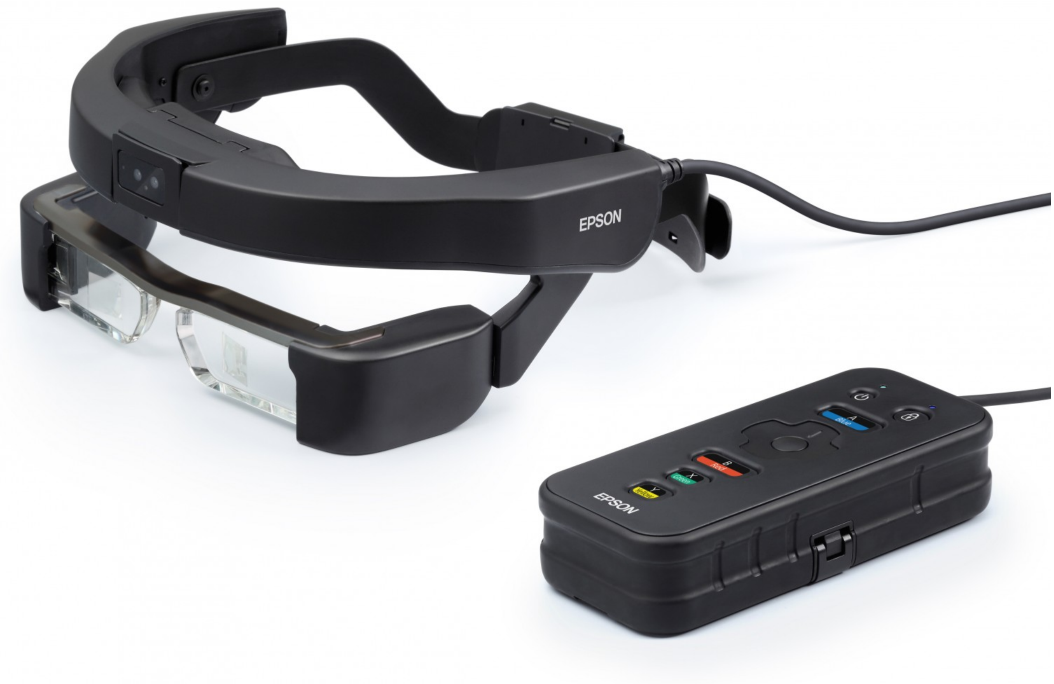


Figure 2.19: Epson - Moverio Pro BT-2000.

because the video cameras themselves have limited dynamic response, and the view of both the real and virtual is generated by the monitor, so everything must be clipped or compressed into the monitor's dynamic range.

AR information needs to be presented to the user in such a way that ambiguities are minimized as to what the information is referring to. Most studies on AR visualization methods cope with two typical sources of such ambiguities; misinterpretation of depth orders and registration errors. For the former problem, it is well known that AR information with solid rendering appears to be front most regardless of its intended depth. To better convey spatial relationships to the real objects, AR information is often rendered with a cut away box or in a semi-transparent manner [24][19]. A combination of wireframe and semi-transparent rendering is proven to help discern depth ordering [37]. For the latter problem, in the presence of registration error, expanded boundary regions based on estimated registration errors have been proposed to disambiguate the target object of concern [20][22]. Robertson et al. report additional visual context can also ameliorate the negative effects of registration error [46]. In this study we propose some new approaches to suppress the impact of moderate registration accuracy to effectiveness of assembly tasks in object assembly.

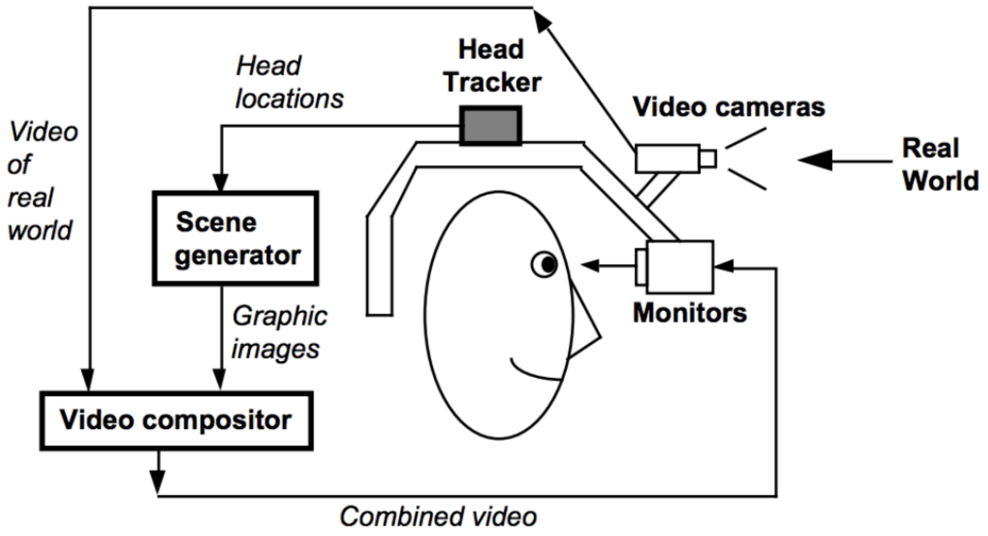


Figure 2.20: Video see-through HMD conceptual diagram [13].

2.3.1 World-Stabilized and Screen-Stabilized Visualization Techniques

Most of visualization techniques introduced in the systems above are world-stabilized visualization techniques. World-stabilized visualization (Figure 2.25) and head (screen)-stabilized information display (Figure 2.26) definition were first introduced in a work of Billinghurst et al. [15] and repeatedly used in the literature (e.g. [42]). In world-stabilized visualization techniques, information is fixed to the real world and its apparent position on screen varies as the user moves his or her head. This requires the user's viewpoint position and orientation to be tracked. World-stabilized information presentation enables annotation of the real world with context dependent visual and audio data, creating information enriched environments. This increases the intuitiveness of the real world tasks [15].

Despite the advantages of world-stabilized visualization techniques, some assembly support systems only use head-stabilized (screen-stabilized) information display. Baird and Barfield [14] presented a system with screen-fixed instructions on untracked monocular OST and opaque HMDs to support a computer motherboard assembly task. In the screen-stabilized visualization techniques, information is fixed to the user's screen and it does not change as the user moves his or her viewpoint. Therefore, guidance information is always available to users and he or she can refer to it very quickly. However, the poses of the virtual 3D guidance information are not normally updated in real-time to match to those of their referring real objects. Thus the user needs to mentally rotate the guidance information to the corresponding real object.

Figure 2.21: Vuzix - Wrap1200AR.

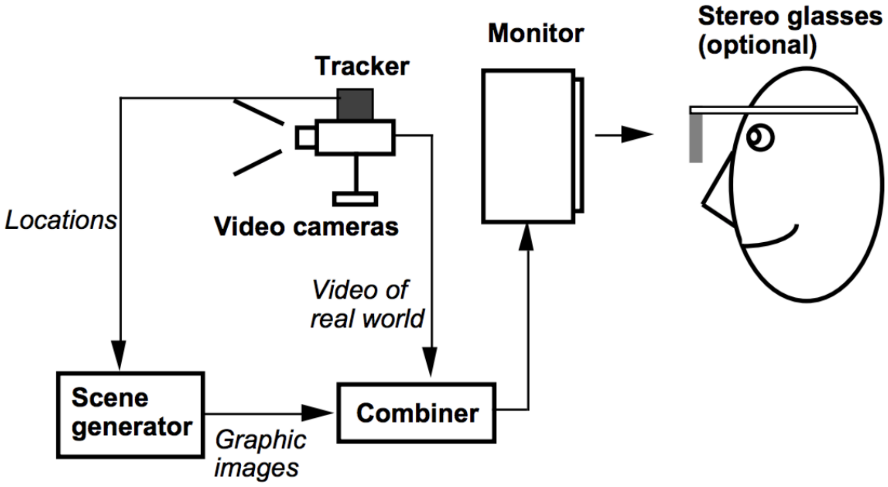


Figure 2.22: Monitor-based AR conceptual diagram [13].

2.4 Survey Summary

A number of manufacturing assembly support systems using AR from "proof-of-concept" applications to the state-of-the-art on-going projects have been reviewed in this chapter. Although a lot of new solutions and techniques for building up an innovative assembly support system have been proposed and improved over the years, there are still a lot of limitations specifically in object tracking and recognition techniques that need to be overcome. In one of the state-of-the-art projects for building up an innovative assembly support system, COGNITO, the user's motions are detected and tracked by using an on-body multi-sensor network. However, multiple sensors attached on the user's body limit the freedom of the user's movement as well as increasing the mental workload.

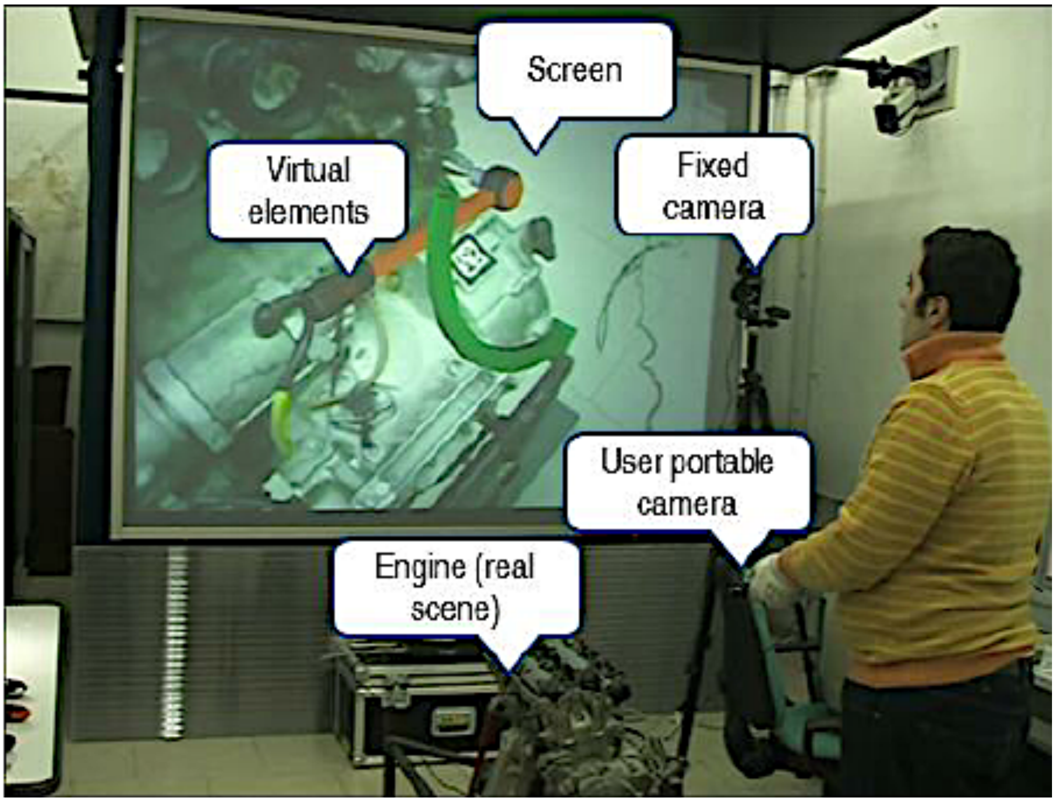


Figure 2.23: An actual monitor-based AR system.

Focusing on the assembled objects themselves, a context-aware assembly support system that can keep track of the objects' status in real-time, and automatically recognize error and completion states at each assembly step as well as display guidance information corresponding to the recognized states flexibly, is desirable and expected to meet the needs for good training, improving labor efficiency and accuracy in work. In the next Chapter we will introduce a such assembly support system using augmented reality that we proposed previously in my master' thesis. We also use it as the test-bed system for evaluations in this thesis.

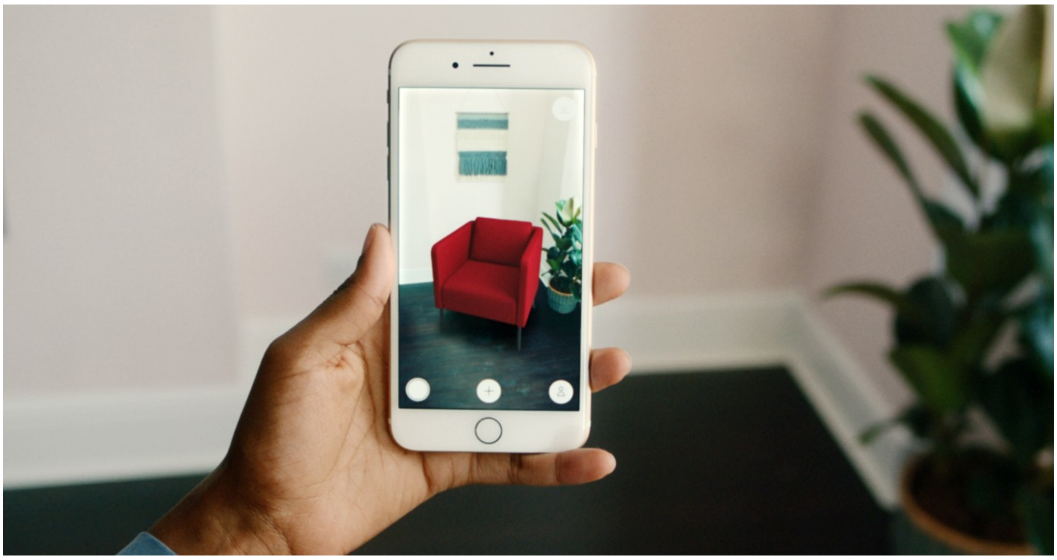


Figure 2.24: AR on a smart phone.

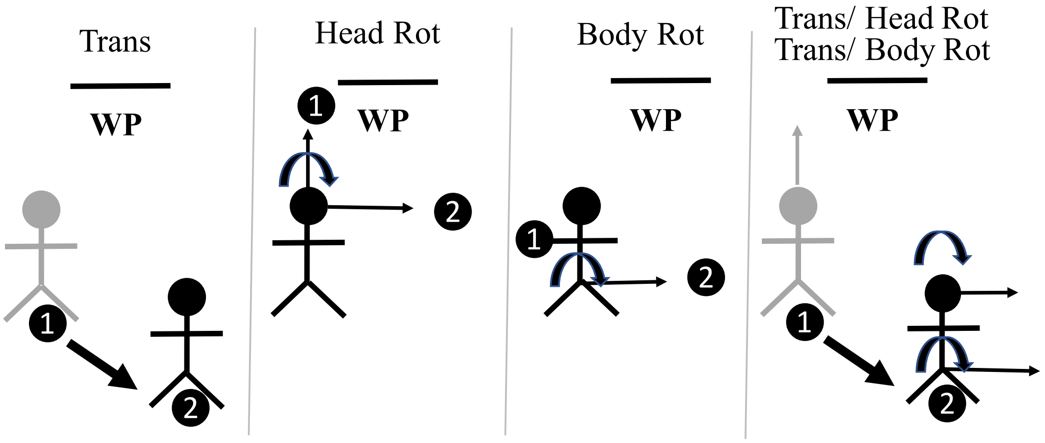


Figure 2.25: World-stabilized AR working planes remain fixed during user movement such as translation and rotation [42].

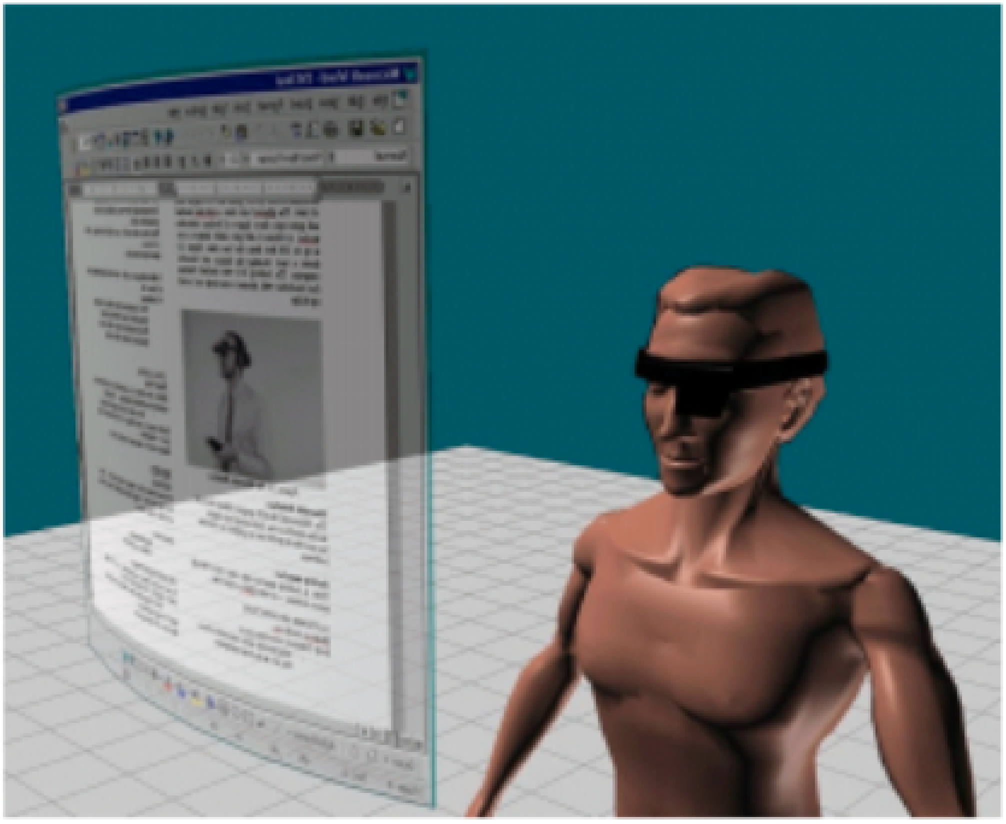


Figure 2.26: Head(sreen)-stabilized visualization [15].

Chapter 3

The Test-Bed System

3.1 System Design

In this chapter we will describe the test-bed system which we used for evaluations in this thesis. The hardware setup of the test-bed system is shown in Figure 3.1. The user builds a building block structure from the table up, layer by layer, while the system highlights the next layer to build on the virtual representation of the real model being assembled. The system uses depth information captured in real-time by a depth sensor, the Microsoft Kinect, for acquiring and tracking. A video see-through head mounted display (Vuzix Wrap 920AR) is used to display assembly guidance information to the user. A depth camera-based system using an algorithm called Lattice-First [38] is used for real-time reconstruction and tracking of LEGO block models. It is fast and effective, providing users with the ability to incrementally construct a block-based physical model using their bare hands while the system updates the model's virtual representation in real-time. ARToolkit library and multi-marker tracking are used to calculate the real camera position (equally with the user's head position) and the orientation relative to the markers in real-time. The system combines marker-based pose estimation of user's head with the depth-based pose estimation of the physical LEGO block models above, guidance information and error detection information can be aligned and displayed to the user under his or her field of view. The software runs on a desktop computer with an Intel core 2 Duo E7400 2.80GHZx2 processor with an NVIDIA ®GeForce ®GT230 GPU.

The software architecture of the test-bed system is shown in Figure 3.2. It consists of three parts as below:

The input part: Depth info captured from a depth sensor and a video stream captured from a camera mounted on a HMD are the main inputs of the system. Another input is the information of the target model. This information can be prepared manually or by automatic tools. It is read into the system at run time for 3D comparison at every frame.

The process part: There are four main modules: 3D construction and model tracking module, assembly guidance and error detection module, multi-marker tracking module and displaying module. The 3D construction and model tracking module receives depth information (point cloud) of physical models being assembled from a depth sensor. It constructs the 3D virtual model of the physical model by considering the point cloud in a predefined

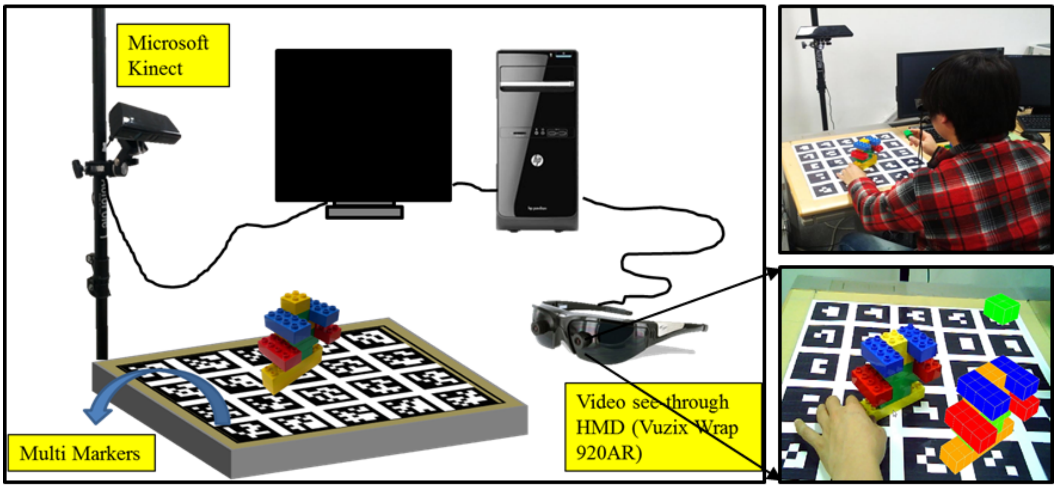


Figure 3.1: The prototype system for guided assembly of building block (LEGO) structures.

3D voxel grid space. By estimating the occupied voxels and vacant voxels in the 3D voxel space every frame as well as using the estimation result from previous frames, this module helps the system recognize which blocks were added or which blocks were removed in the physical model. Then, the system can update the virtual model. In addition, by calculating the surface normal from depth information as well as updating the model information through frames (block grid information, block's vertices, the surface normal, etc.) this module helps the system keep track of the relative position and the orientation of the physical model with the depth sensor. The assembly guidance and error detection module reads information of the target model into the system and reconstructs its 3D virtual model at run time. Then, it compares this virtual model with the 3D virtual model of the physical model to determine correct parts, incorrect parts and parts that need to be filled in the physical model being assembled. At the same time, working in a different coordinate system and context, the multi-marker tracking module receives a video stream captured from a camera mounted on a HMD, recognizes markers and calculates the camera pose relative to the markers in real-time. The display module uses both pose estimations (the marker-based pose estimation and the depth-based pose estimation) to align guidance information with the physical model in the user's field of view.

The output part: A video stream with virtual guidance information rendered in each video frame is sent to the HMD in real-time and displayed in the user's view.

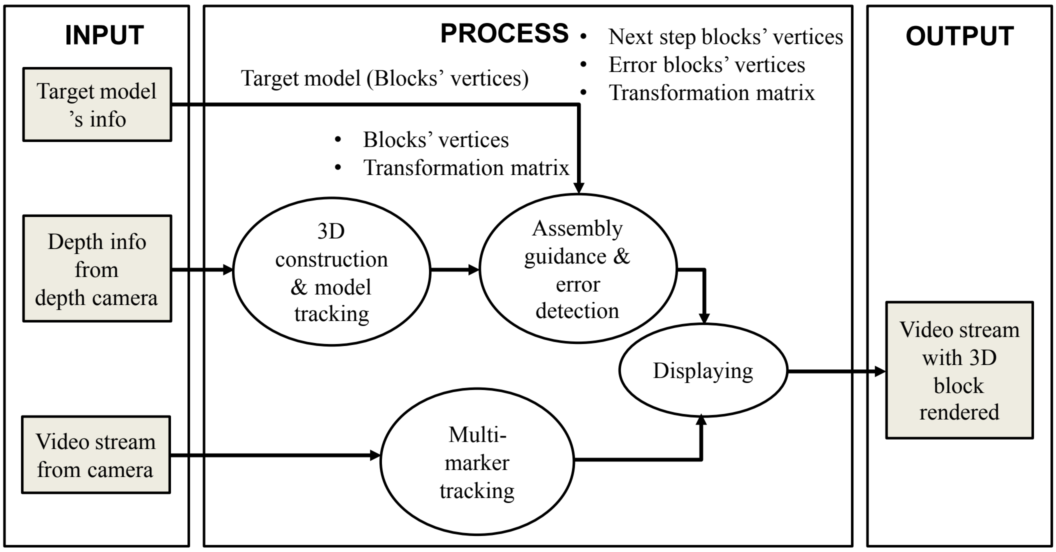

INPUT
Target model 's info
Depth info from depth camera
Video stream from camera
PROCESS
Target model (Blocks' vertices)
• Next step blocks' vertices
• Error blocks' vertices
• Transformation matrix
• Blocks' vertices
• Transformation matrix
3D construction & model tracking
Assembly guidance & error detection
Displaying
Multi-marker tracking
OUTPUT
Video stream with 3D block rendered

Figure 3.2: The software architecture of the proposed system.

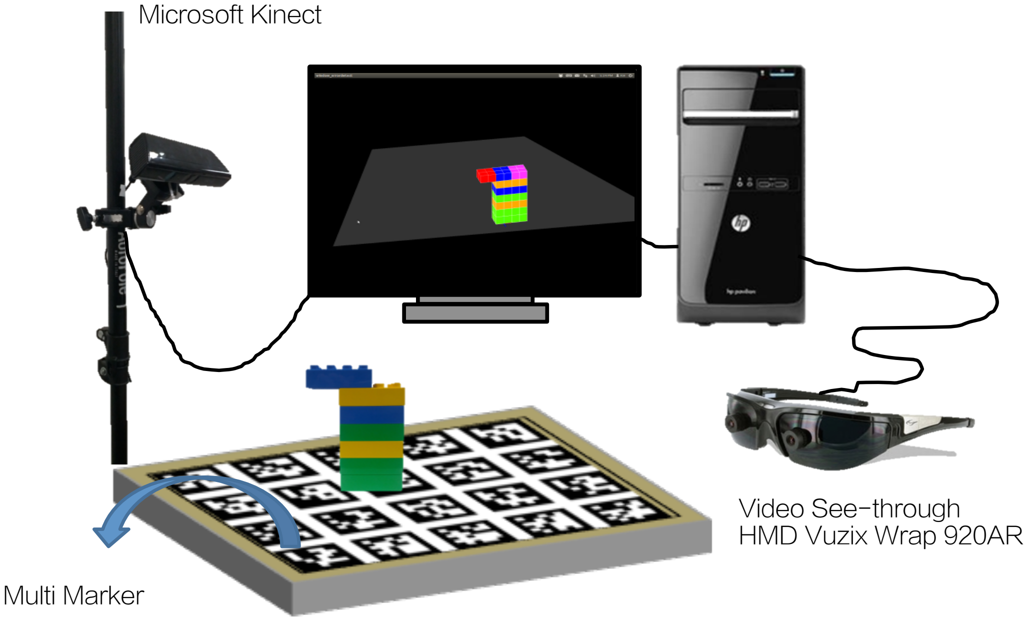


Figure 3.3: Hardware setup for the proposed system.

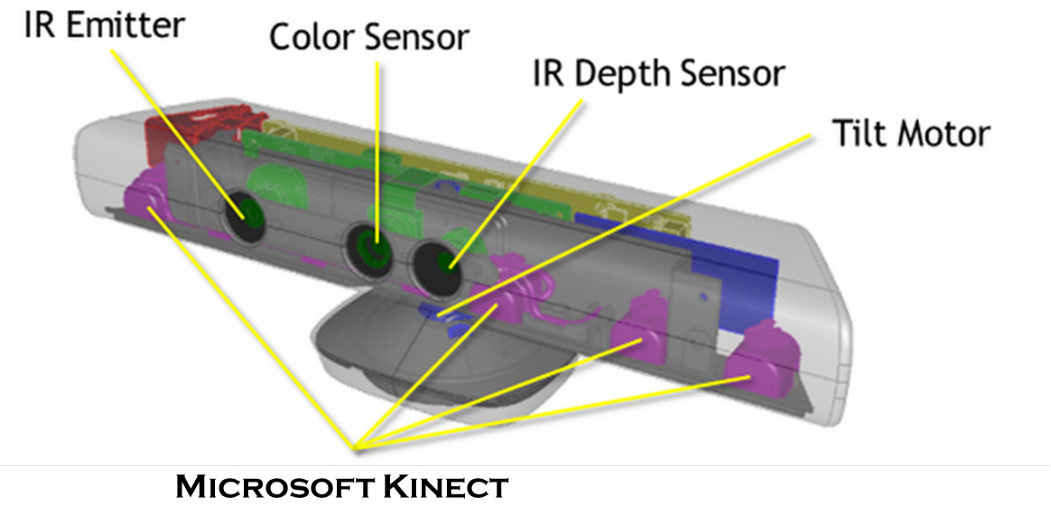


Figure 3.4: Kinect hardware [7].

Kinect	Specifications
Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	27°
Frame rate (depth and color stream)	30 frames per second (FPS)
Accelerometer characteristics	A 2G/4G/8G accelerometer configured for the 2G range, with a 1° accuracy upper limit.

Table 3.1: Kinect hardware specifications [7].

3.2 Hardware Setup

The Kinect sensor by Microsoft Corp. was introduced to the market in November 2010 as an input device for the Xbox 360 gaming console. The basic principle behind the Kinect depth sensor is emission of an IR pattern and the simultaneous image capture of the IR image with a CMOS camera that is fitted with an IR-pass filter. The image processor of the Kinect uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image. The Kinect hardware images and main specifications are shown in Figure 3.4 [7] and Table 3.1 [7].

The Wrap 920AR is a product of Vuzix [10], which is based upon the Vuzix Wrap 920 video eyewear and a Wrap VGA adapter that enables PC compatibility and connectivity. Embedded into the face of the eyewear are two specially modified USB video cameras are connected to a PC as two discrete USB webcams. Using the same display system as the popular Wrap 920 video eyewear, the Wrap 920AR provides the visual equivalent of a 67-

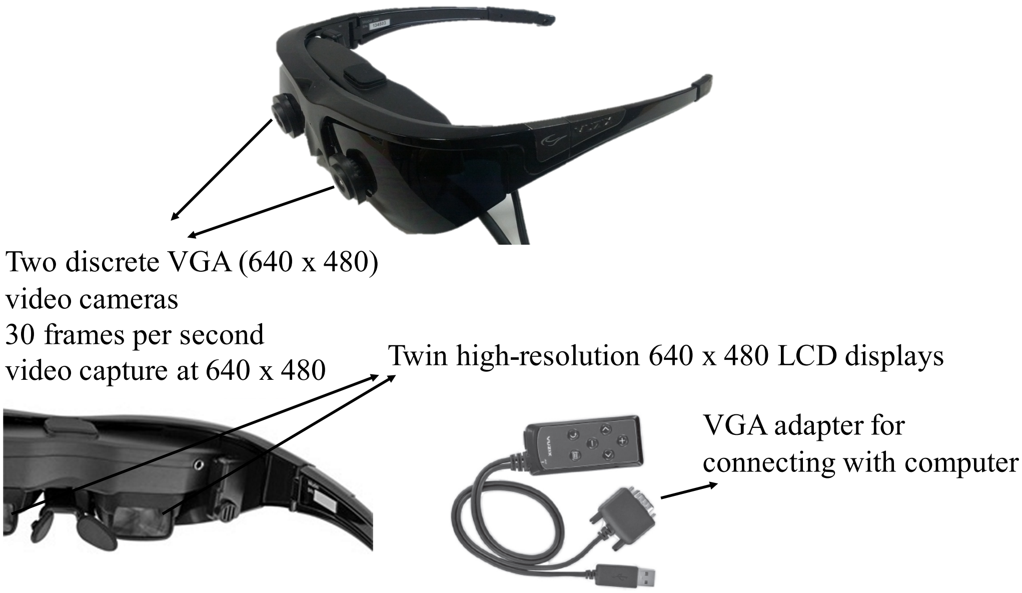

Two discrete VGA (640 x 480)
video cameras
30 frames per second
video capture at 640 x 480
Twin high-resolution 640 x 480 LCD displays
VGA adapter for
connecting with computer

Figure 3.5: Vuzix Wrap 920AR [10].

inch display, at a distance of 3 meters, displaying crystal clear 2D or 3D video. For use as an augmented reality device, it is connected to a Windows based personal computer (desktop, netbook or laptop) through the included Wrap VGA adapter. The twin camera system of the Wrap 920AR provides two discrete video sources, each provided to a USB connected PC as a standard USB video camera device. Each camera captures 640 × 480 VGA video at 30 frames per second. The main specifications of the Wrap 920AR is shown in Figure 3.5 [10] and Table 3.2 [10].

3.3 Interaction between Users and the System

The interaction between the user and the system is shown in Figure 3.6. The user wears a video see-through Vuxiz 920AR HMD, sits down at a table and assembles toy block (Duplo) structures on the tabletop. The Kinect sensor is placed opposite to the user. It is suspended about half a meter on a stand above the tabletop. Depth information is captured by the Kinect sensor and a video stream is captured by the camera on the HMD in real-time. The system processes these input data, constructs a 3D virtual model, keeps track of the physical model and detects error blocks at every assembly step. Finally, the system in different visualization modes we will propose and investigate in the following chapters.

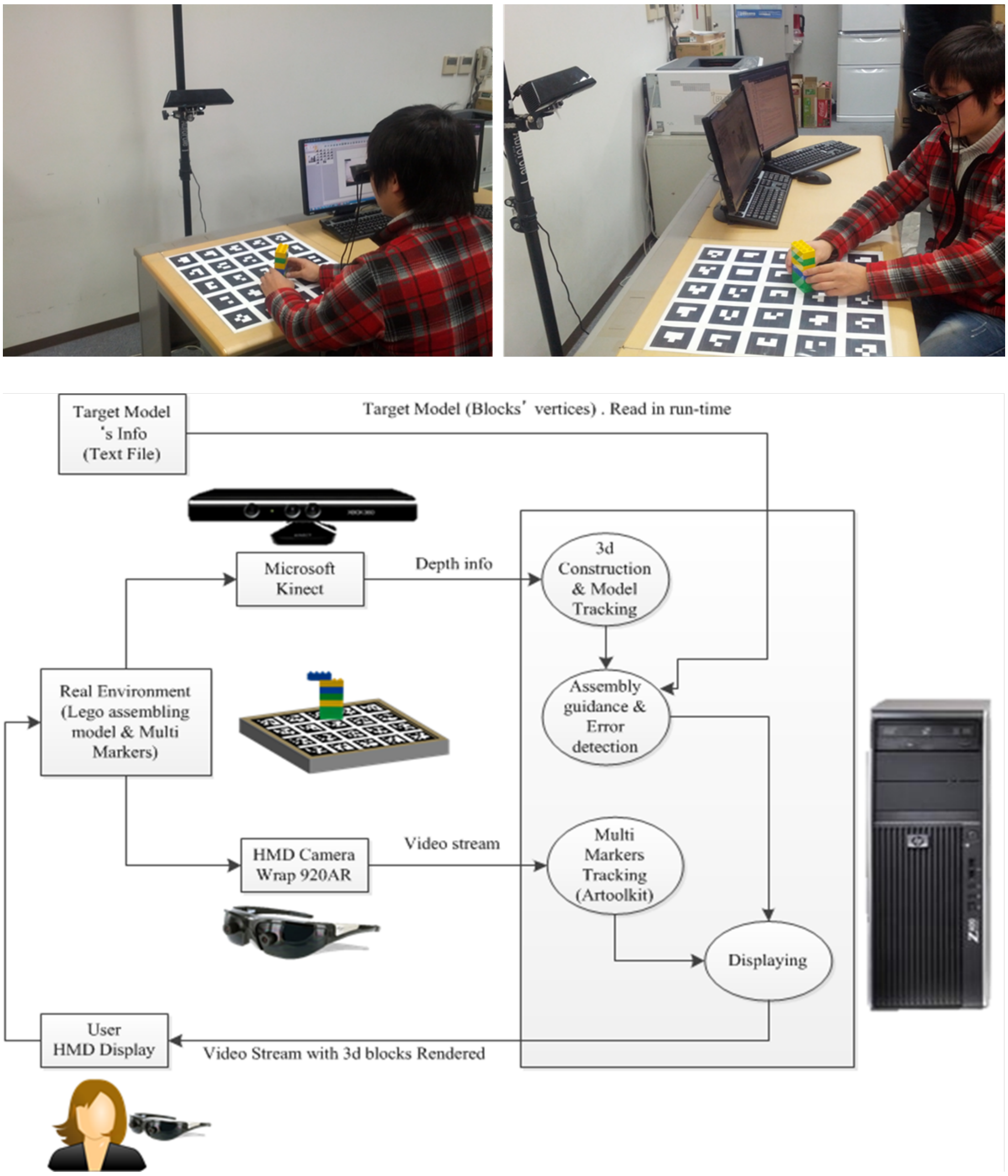

Target Model 's Info (Text File)
Target Model (Blocks' vertices) . Read in run-time
Microsoft Kinect
Depth info
3d Construction & Model Tracking
Assembly guidance & Error detection
Real Environment (Lego assembling model & Multi Markers)
HMD Camera Wrap 920AR
Video stream
Multi Markers Tracking (Artoolkit)
Displaying
User HMD Display
Video Stream with 3d blocks Rendered

Figure 3.6: Interaction between the user and the system.

Vuzix Wrap 920AR	Specifications
Display resolution	Twin high-resolution 640 × 480 LCD displays
Screen size	Equivalent to a 67-inch screen as viewed from ten feet (approximately 3 m)
Refresh rate	60 Hz progressive scan update rate
Video distortion	Ultra-low video distortion
Field of view	31-degree diagonal field of view
Color depth	24-bit true color (16 million colors)
Focus	Independent +2 to -5 diopter focus adjustment
Cameras	Two discrete VGA (640 × 480) video cameras
Camera frame rate	30 frames per second video capture at 640 × 480
Driver	USB video camera – no proprietary drivers required
PC connection	Connects to VGA port
Adapter	Includes VGA to DVI video port adapter
Video card	Works with all Windows dual monitor compatible video card, all brands

Table 3.2: Vuzix Wrap 920AR specifications [10].

3.4 Interactive 3D Model Reconstruction and Tracking

The Interactive 3D Model Reconstruction and Tracking Module is one of the most important modules of the system. In this module, depth info captured in real time from the depth sensor is used to construct and update the 3D virtual model of the physical model. The implementation of this module's functions is based on an algorithm called Lattice-First which was introduced by Andrew Miller, University of Central Florida [38]. The algorithm assumes that the toy block structures always remain in the tabletop horizontally during translation and rotation. The pose estimation problem was therefore constrained to three degrees of freedom (two degrees of freedom in translation and one degree of freedom in rotation) rather than six. The building blocks are arranged in a 3D point lattice where the smallest building block unit is the basis (Figure 3.7) [38]. The lattice can be written as N_1W_x, N_2W_y, N_3W_z where N_1, N_2, N_3 are points from the depth camera and W_x,W_y, W_z are the dimensions of the smallest building block unit. The block unit width is the same in the X and Z axes ($W_x = W_z$). The Y axis is perpendicular to the tabletop.Under these assumptions, points on the surfaces of the blocks lie on orthogonal planes intersecting the lattice and parallel to XY, YZ, or XZ. Duplo blocks are used which have unit dimensions (W_x, W_y, W_z) = (16mm, 19.2mm, 16mm). The algorithm takes advantage of the orthogonal and grid-like properties of building

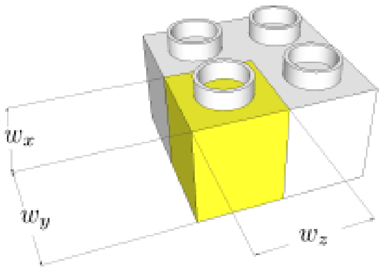


Figure 3.7: Size of the basic Duplo block.

block structures in order to achieve robustness to interference and occlusions from the user's hands, and to support a dynamic model in which pieces can be added and removed. The algorithm is fast and effective, providing users with the ability to incrementally construct a block-based physical model while the system maintains the model's virtual representation.

The algorithm first finds the transformation from physical coordinates to model coordinates, in which the blocks comprising the tracked object align with the coordinate system. This transformation is represented as the product of several matrices:

$$\boldsymbol{P}^{\boldsymbol{model}} = (\boldsymbol{C})(\boldsymbol{T})(\boldsymbol{R})\boldsymbol{P} \tag{3.1}$$

where P is a 4 × N matrix, N is the total number of points from the depth camera, R is a rotation matrix, T is a translation matrix, and C is a discrete correction (translation by a multiple of $W_x = W_z$ or rotation by a multiple of 90^0) that aligns the current frame to a previous model estimate. After finding this transformation, the remaining goal is to update the voxel grid of the virtual model, estimating which voxels are occupied and which are vacant. The steps of the algorithm (Figure 3.8) are summarized as follows:

1. Take a new depth image from the sensor, computing point positions P and surface normals n̂.

2. Use the surface normals n̂ to determine the lattice orientation R and the oriented point samples.

3. Determine the lattice translation (i.e., T) and aligned point samples.

4. Bin the point samples to determine occupancy estimates and vacancy estimates.

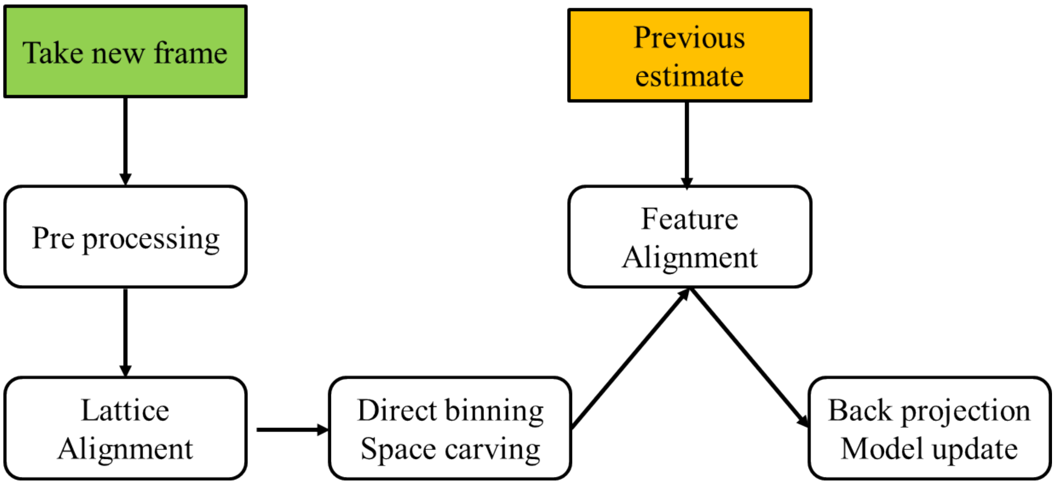


Figure 3.8: The steps of the algorithm.

5. Use space carving to augment the vacancy estimates.
6. Align the previous estimate to the current estimate using feature matching in a finite search space (i.e., find C).
7. Back project the union of the previous estimate and current estimate into the depth image to validate updates to the model.

This part will illustrates how the algorithm works. The previous estimation of the assembling model was resumed as shown in Figure 3.9. At the previous estimation, the blocks 1, 2 are assumed that they have not been seen by the Kinect so they were not rendered on the screen. Only a part of the model is rendered. Next, the current frame is assumed as shown in Figure 3.10. The bocks 1, 2 now have been seen by the Kinect.

At the preprocessing step, the camera is assumed that it has been extrinsically calibrated to the table surface so that the XZ plane of the measurements is coplanar with the table. This extrinsic calibration is performed by manually clicking four corner points in a depth image of the empty table surface. These four points are also used to define a 3D volume of interest, a prism extending upward from a quadrilateral on the table as indicated by the four clicked points. The bounds of this volume are used to segment the foreground (i.e., the user's hands and the block structure) from the background, without requiring the background to remain static.

The depth info captured from the depth sensor of the Kinect is arranged in a two-dimensional grid as a depth image (Figure 3.11). The depth image is smoothed using a uniform kernel, and the surface normals (orthogonal vector with the surface at observed point) are computed using the method described in [29].

The next step of the algorithm is to determine the orientation of the lattice that fits to the physical model by finding the dominant orientation of the

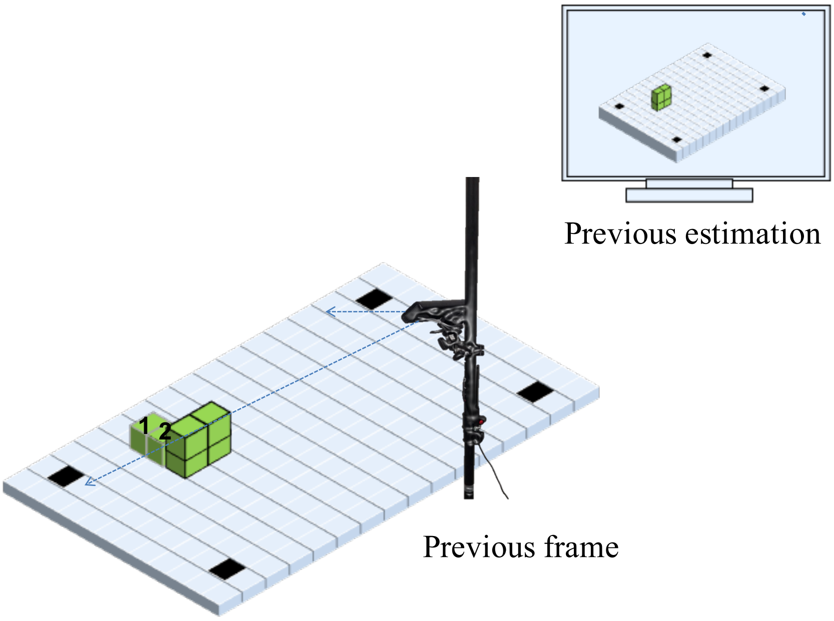


Figure 3.9: The previous frame estimation.

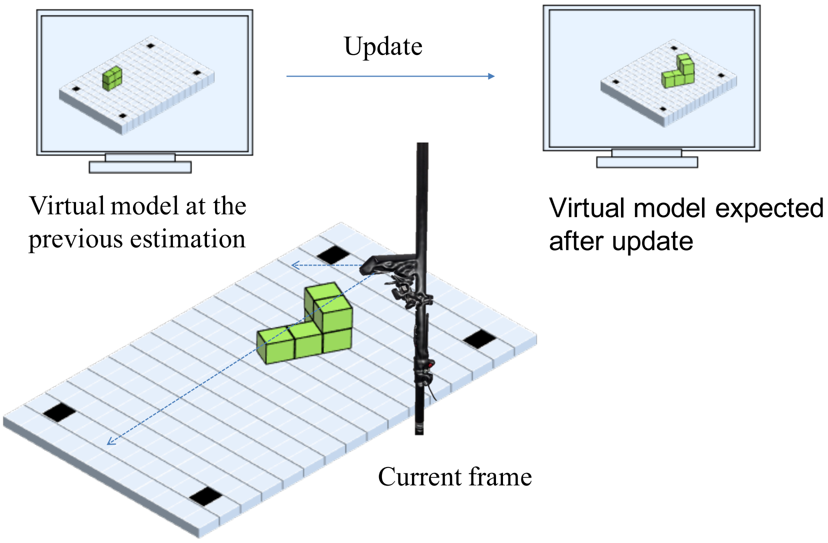

Update
Virtual model at the previous estimation
Virtual model expected after update
Current frame

Figure 3.10: The current frame.

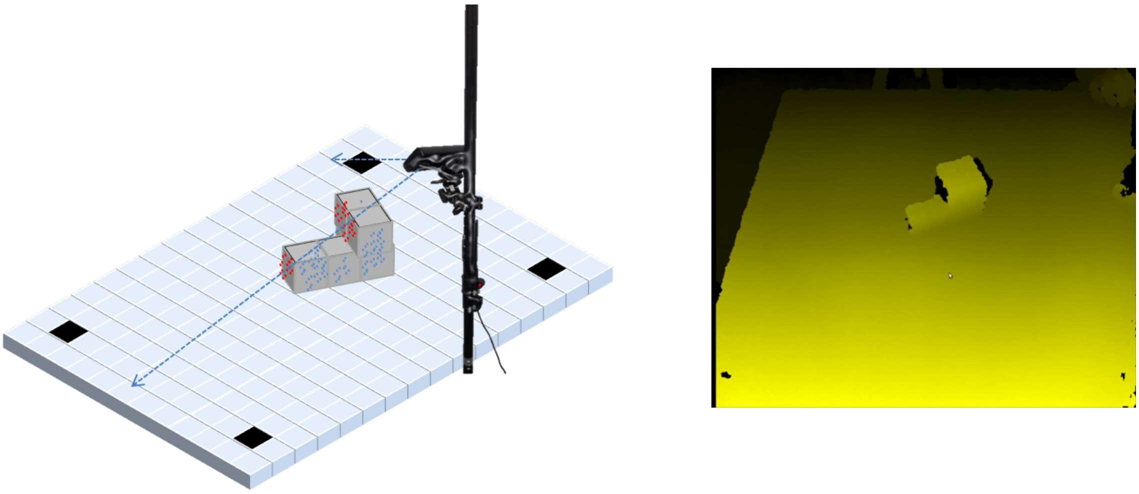


Figure 3.11: The depth information (point cloud). The surface points can be labeled as aligning with the X and Z axis (red or blue).

surface normals of the model. The normal vectors that are not parallel to the table surface are discarded (Figure 3.12). Other normal vectors will form four orthogonal clusters corresponding to the possible surface orientations of the blocks and the dominant orientation of the surface normals is estimated. The angle between the dominant orientation of the surface normals and the X or the Z axes is found and this value is used to calculate the orientation matrix of the physical model with the X or the Z axes.

The next step in the algorithm is calculating the translation of the lattice with the physical model. Since the blocks are assumed to be in contact with the table surface horizontally during translation and rotation, the Y component of translation is zero by assumption. Only translation of the lattice on XZ plane is considered. The distance from each point in the physical model's point cloud to the nearest lattice plane is calculated and the suitable translation is estimated. As in Figure 3.13, the distance from point cloud to X axis, and Z axis (T_x, T_z) are calculated based on the surface normals. From T_x, T_z the suitable translation is calculated so that each point of the point cloud can be mapped to the nearest lattice plane.

At this timing, even though the lattice is aligned to the point cloud, the shape of the virtual model hasn't been formed. The point cloud is considered in a 3D voxel grid space and the occupied voxels, that the points of point cloud belong with, are estimated to make the shape of the 3D virtual model. The observed points are binned to the nearest voxel surface and tallied. Each observed point corresponds to a surface between two grid cells and can be considered an evidence of the occupancy of one voxel and the vacancy of another. The tally for each voxel is compared to a threshold $T_v = 30$ [38] to obtain binary values for occupancy and vacancy of each voxel (Figure 3.14).

The direct binning step only produces estimates on, or adjacent to the

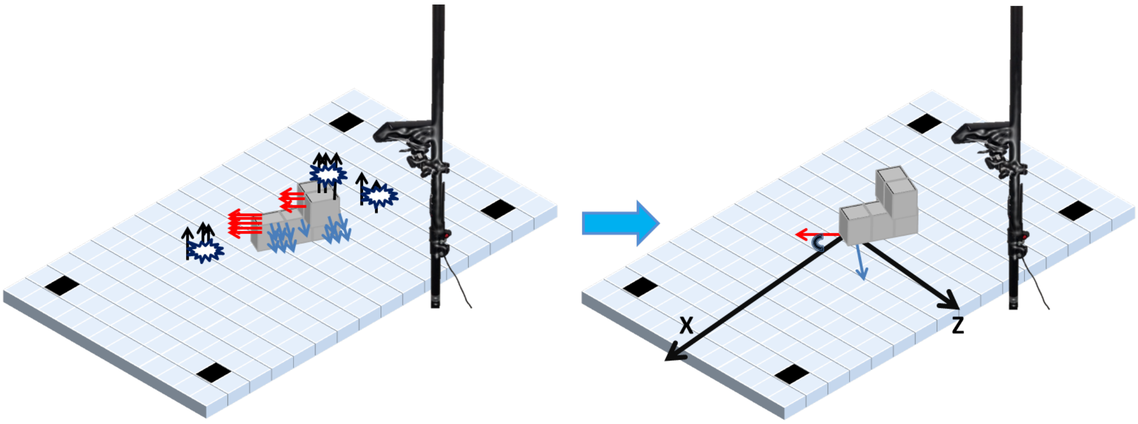


Figure 3.12: Calculate surface normal vectors. Discard the normal vectors that are not parallel to the table surface. Find rotation angle of the model based on dominant normal vectors.

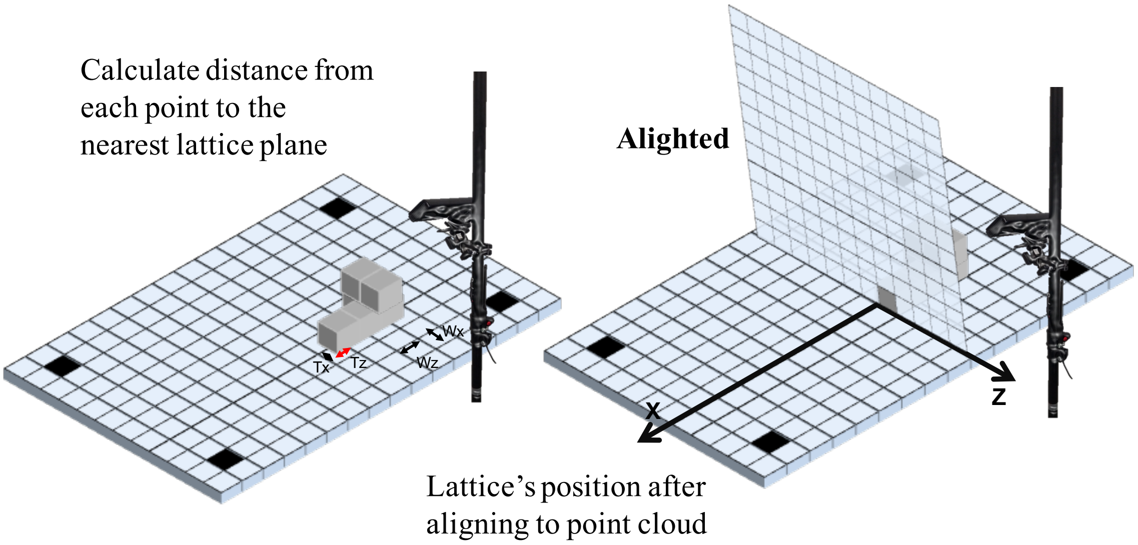

Calculate distance from each point to the nearest lattice plane
Tx
Tz
Wx
Wz
Alighted
X
Z
Lattice's position after aligning to point cloud

Figure 3.13: Calculate lattice translation.

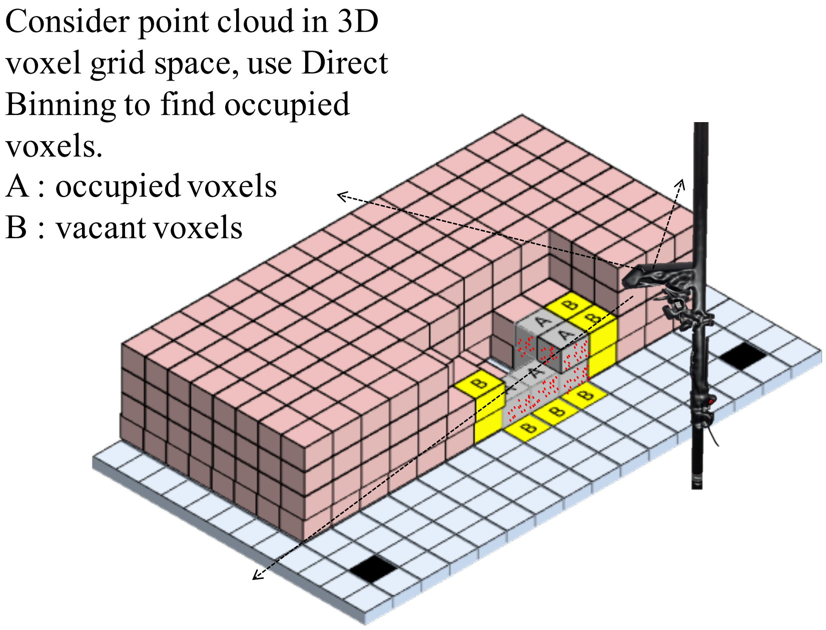

Consider point cloud in 3D voxel grid space, use Direct Binning to find occupied voxels.
A : occupied voxels
B : vacant voxels
A
A
A
B
B
B
B
B
B

Figure 3.14: Direct binning.

occupied blocks, when in fact many more voxels can be marked vacant using information from the depth image. A variant of space carving [36] is used. The adaptation of this technique is simpler: the center point of each voxel is projected onto the depth image and compared to the corresponding depth measurement. If the depth measurement is farther from the camera than the projected center point, then the voxel is marked vacant (Figure 3.15).

After finishing direct binning and space carving, the shape of the 3D virtual model of the current assembly toy block structure is presented and it lies exactly on the lattice. In order to align this 3D virtual model with the previous model (status of the model in the previous frame) to find changes, translations are performed easily by integer multiples of $W_x = W_z$, and rotations are performed by a multiple of 90^0. A cost function is used to reward the matching at each translation or rotation of the physical model in the current state. The cost value will be smallest at the best matching position of current model with the previous model (Figure 3.16).

To make the matching process faster, the algorithm does not evaluate the cost function for every possible translation and rotation, feature-based points that are XZ corner voxels (voxels marked occupied that are adjacent to two vacant or unmarked) are used to reduce the comparison to a subset of the possible alignment. The cost value among these possible alignment are evaluated and the best matching will give the smallest cost value. At the best matching, the system compares the current state of the physical model with the previous and it will know at where the physical model is changed and then it can update the 3D virtual model. As in Figure 3.17, the system has more

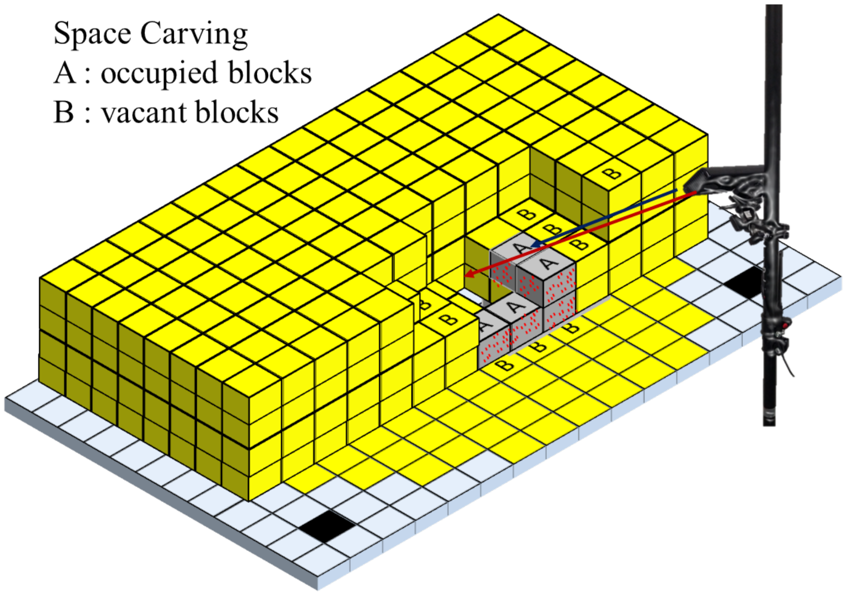
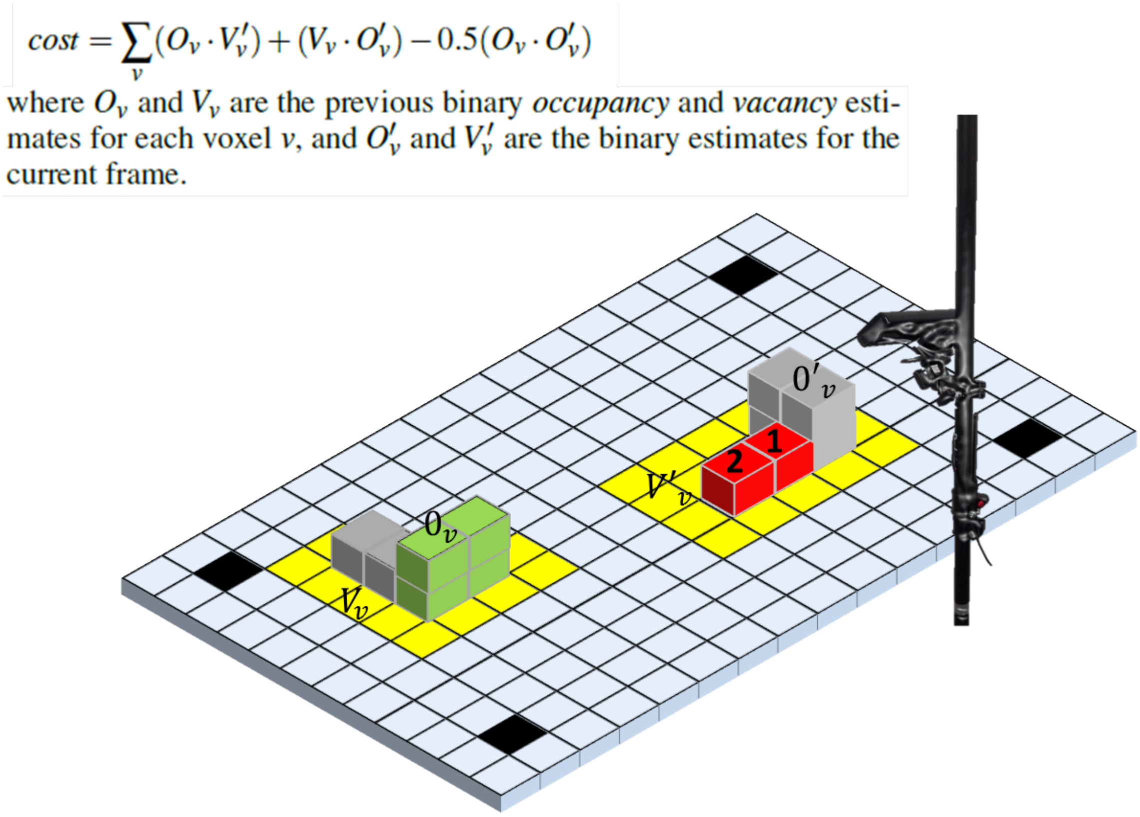


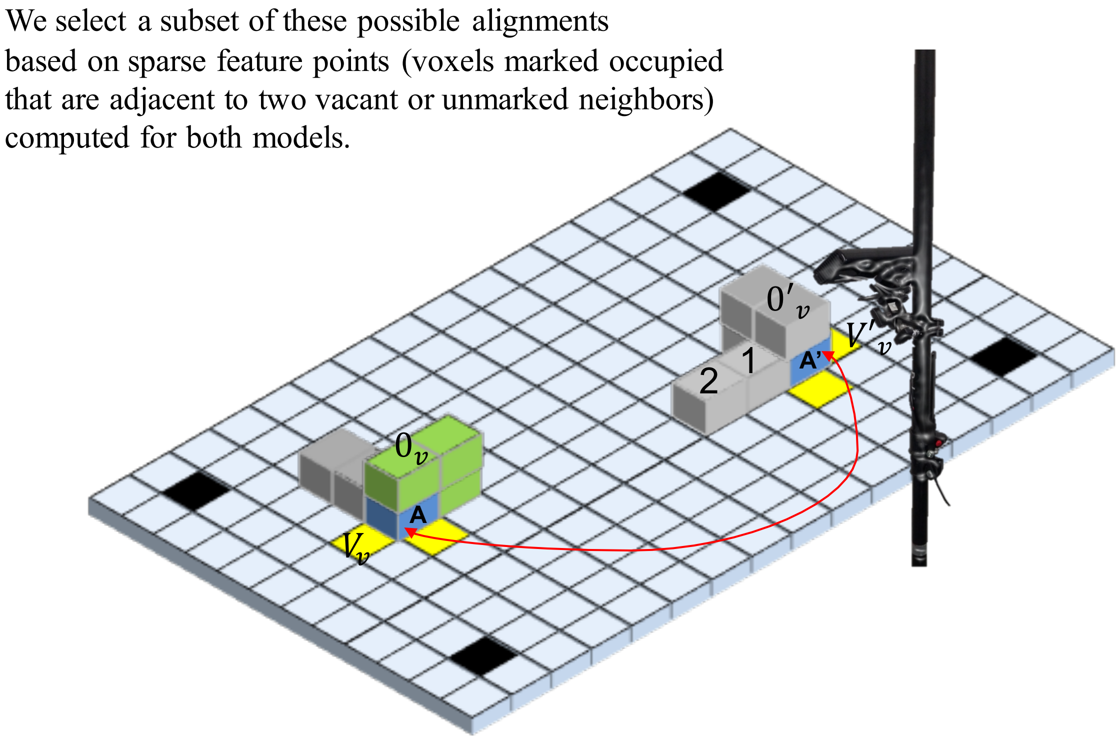
Figure 3.15: Space carving.



$$cost = \sum_{v} (O_v \cdot V'_v) + (V_v \cdot O'_v) - 0.5(O_v \cdot O'_v)$$

where O_v and V_v are the previous binary *occupancy* and *vacancy* estimates for each voxel v, and O'_v and V'_v are the binary estimates for the current frame.

Figure 3.16: Alignment based on a cost function.



We select a subset of these possible alignments based on sparse feature points (voxels marked occupied that are adjacent to two vacant or unmarked neighbors) computed for both models.

Figure 3.17: Use features of XZ corner voxels for alignment.

information of the blocks 1 and 2, that unmarked (unknown blocks) in the previous estimation, at the current frame. So now, the system can update the virtual model with voxel 1 and 2 rendered.

3.5 Assembly Guidance and Error Detection Module

The implementation of this module makes this system become different with existing assembly support systems. At this module, information of the target model is prepared and read into the system at run time. Its 3D virtual model is reconstructed and the system compares this virtual model with the 3D virtual model of the physical model being assembled in real time at every frame to find out not-filled parts as well as error parts in the physical model.

3.5.1 Assembly Guidance Mechanism

In this part, the assembly guidance mechanism will be explained in detail. the target model and the physical model being assembled are assumed that they have the shapes as shown in Figure 3.18. The system is expected to be able

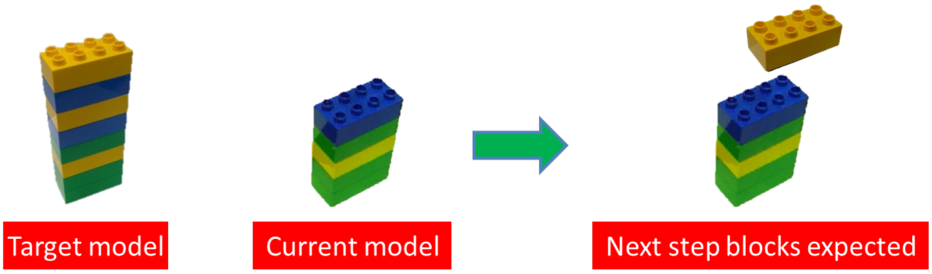


Figure 3.18: The assembly guidance scenario.

to recognize the assembly toy block structure's status and guide the user with the next step blocks as shown in the figure.

The main point in the assembly guidance mechanism is that besides conducting comparison between the current 3D virtual model of the physical model (constructed at the current frame) with the previous 3D virtual model (constructed at the previous frame), the system conducts another comparison between the current 3D virtual model with the 3D virtual model of the target model which is constructed at run time. The encoding of the target model's structure information in a specific way (will be explained in Section 3.5.3) so that the system can easily construct the 3D virtual model of the target model in a 3D voxel space at run time plays an essential role. The assembly guidance mechanism proposed is shown in detail in Figure 3.19.

At first, the system constructs the 3D virtual model of the target model and the 3D virtual model of the physical model being assembled. Next, the two 3D virtual models are considered in the same 3d voxel grid space. The system estimates the models based on estimating occupied voxels and vacant voxels on each model. The two models need to be aligned together before the system can compare them. A feature alignment algorithm which takes advantages of XZ corner voxels' features is used to align the two models. Then, the system compares two models base on occupied voxels to find out the parts that should be filled in the physical model. Finally, the system extracts the lowest layer of the parts that should be filled in the physical model to make the next step assembly guidance information.

3.5.2 Error Detection Mechanism

The idea of conducting the comparison between the current 3D virtual model with the 3D virtual model of the target model at run time is applied in the error detection mechanism.

The target model and the physical model being assembled by the user are assumed as shown in Figure 3.20. There are four error blocks in the assembling model if compared with the physical target model.

Humans easily recognize error blocks when comparing the two models, but it is not easy work for computer systems. The error detection mechanism proposed is illustrated in Figure 3.21. To detect error parts on the assembling physical model, in the comparison step of two 2D virtual models, blocks that exist in the physical model but do not exist in the target model are treated as error blocks. The system notifies the user these blocks by rendering highlighted red blinking wire frames in overlay mode or red blinking solid cubes in side-by-side mode.

3.5.3 Information Input Mechanism for the Target Model

As mentioned in Section 3.5.1, The encoding of the target model's structure information in a specific way so that the system can easily construct the 3D virtual model of the target model in a 3D voxel space at run time plays a very essential role in the assembly guidance mechanism and error detection mechaism.

First, the target model is separated layer by layer. At each layer, occupied voxels and vacant voxels of that layer are estimated. The occupied voxels are marked by 1, and the vacant voxels are marked by 0. After this step, the layers are presented by two-dimensional arrays with 0 and 1 values and

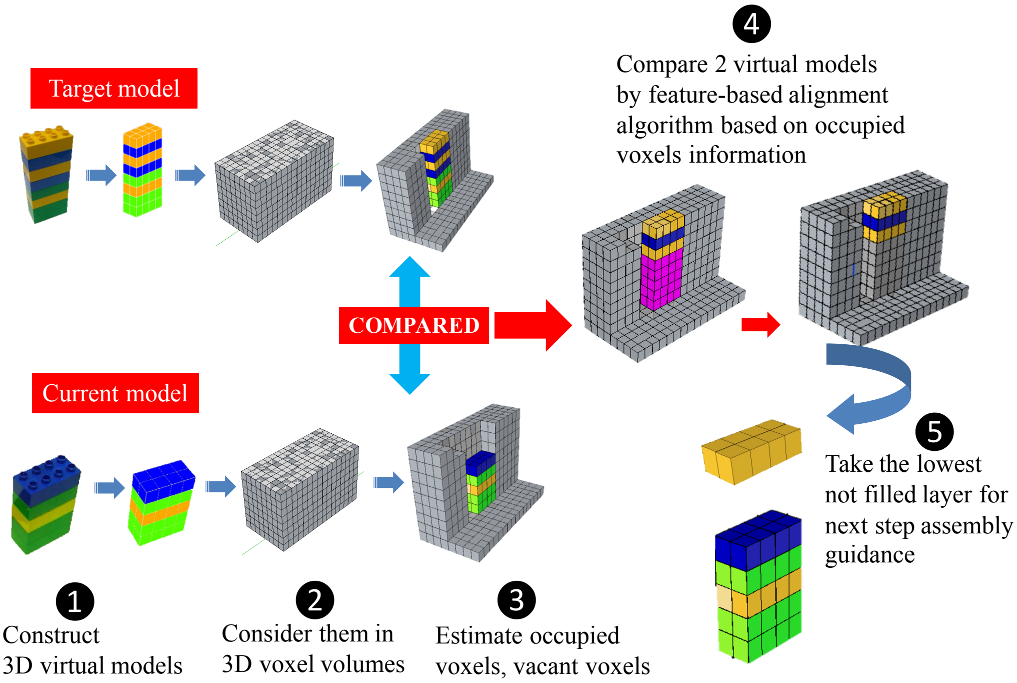

Target model
4
Compare 2 virtual models by feature-based alignment algorithm based on occupied voxels information
COMPARED
Current model
5
Take the lowest not filled layer for next step assembly guidance
1
Construct 3D virtual models
2
Consider them in 3D voxel volumes
3
Estimate occupied voxels, vacant voxels

Figure 3.19: Assembly guidance mechanism. Two models are compared in real-time based on occupied voxels information estimated in every frame. The lowest layer of parts that have not been filled in is displayed as the next assembly step.

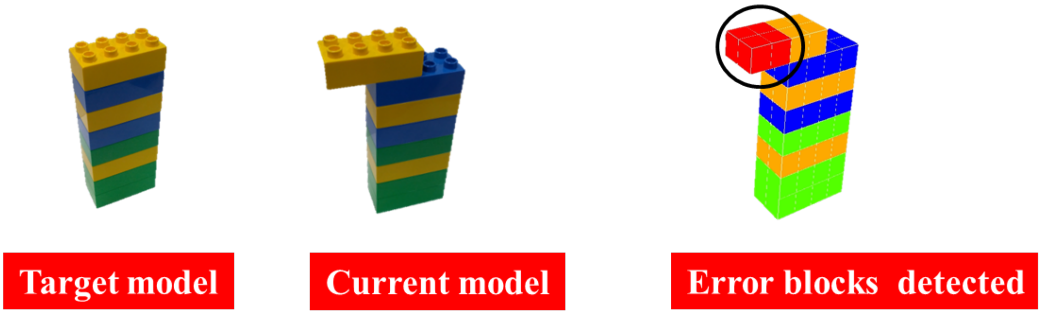


Figure 3.20: The error detection scenario.

the whole target model is presented by a three-dimensional array. The index of the array is equal to the order of the layers of the model from the table-top. The mechanism for inputting the target model's information is shown in Figure 3.22.

Another challenge is presenting the color information of the target model. The following approach is used. the colors of blocks are presented by different codes (Figure 3.23). The red color is encoded by 1, the green color by 2, the blue color by 3 and the yellow color by 4 in the same text file. The system converted these codes to corresponding colors and found the corresponding blocks with these colors based on the index of the layers and position of the blocks in each layer.

3.6 Multi-Marker Tracking Module

To track the user's viewpoint during the assembly process in real time so that the virtual images are exactly aligned with real world objects, ARToolkit and markers are used. The ARToolKit uses computer vision techniques to calculate the real camera position and orientation relative to the markers, allowing the programmer to overlay virtual objects onto these markers. ARToolKit tracking works as follows:

1. The camera captures video of the real world and sends it to the computer.
2. Software on the computer searches through each video frame for any square shapes.
3. If a square is found, the software uses some mathematics to calculate the position of the camera relative to it.
4. Once the position of the camera is known a computer graphics model is drawn from that same position.

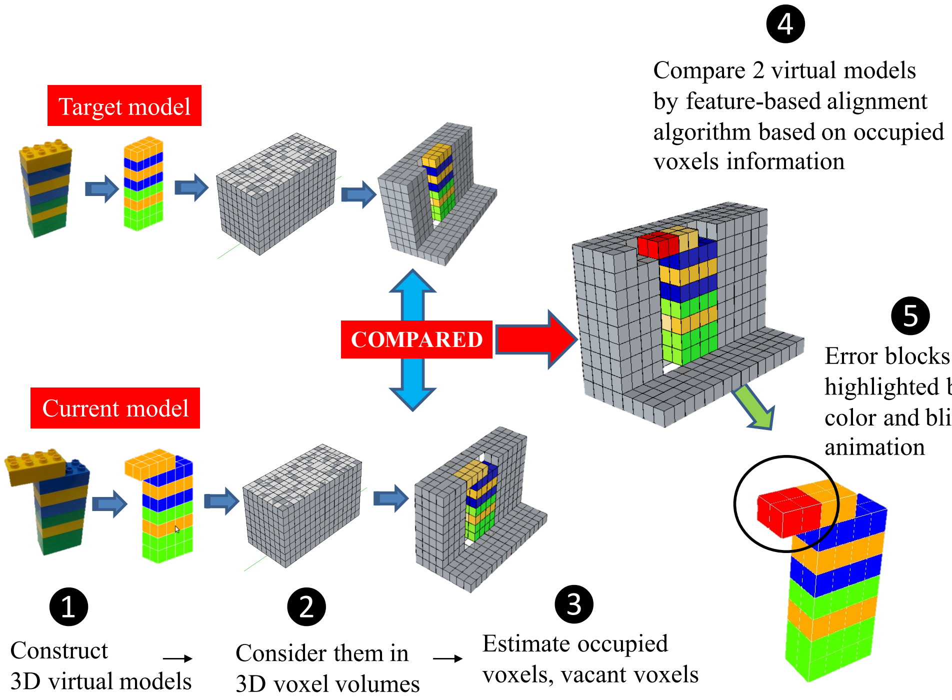

4
Compare 2 virtual models by feature-based alignment algorithm based on occupied voxels information
Target model
COMPARED
5
Error blocks are highlighted by red color and blink animation
Current model
1
Construct 3D virtual models
2
Consider them in 3D voxel volumes
3
Estimate occupied voxels, vacant voxels

Figure 3.21: Error detection mechanism. Blocks that exist in the physical model but do not exist in the target model are marked as error blocks.

5. This model is drawn on top of the video of the real world and so it appears stuck on the square marker.

6. The final output is send back in the display, so when the user looks through the display they see graphics overlaid on the real world.

Figure 3.24 below summarizes these steps. The fast, precise tracking provided by ARToolKit can help keep track the user's viewpoint in real time. An obvious problem when using a single marker is hand occlusion. Tracking will fail when the track of marker is partially covered by a user's hand or other objects (Figure 3.25). To solve the hand occlusion problem which is usual in assembly process, multiple markers are used instead of a single marker. A set of markers are prepared, whose positions are fixed on the tabletop. The size of each marker is 80mm, and a gap between markers is 20mm. The world coordinate frame is defined as a set of coordinate axes aligned with the table surface at the top left of the center marker in the markers set. Then, the position of all markers in the marker set relative with the coordinate origin are defined. The coordinate system of the markers is shown in Figure 3.27.

The size and coordinates of the markers are stored in a configuration file. The structure of the configuration file is shown in Figure 3.28. The multi-marker tracking principle is used: a set of markers are defined based on their

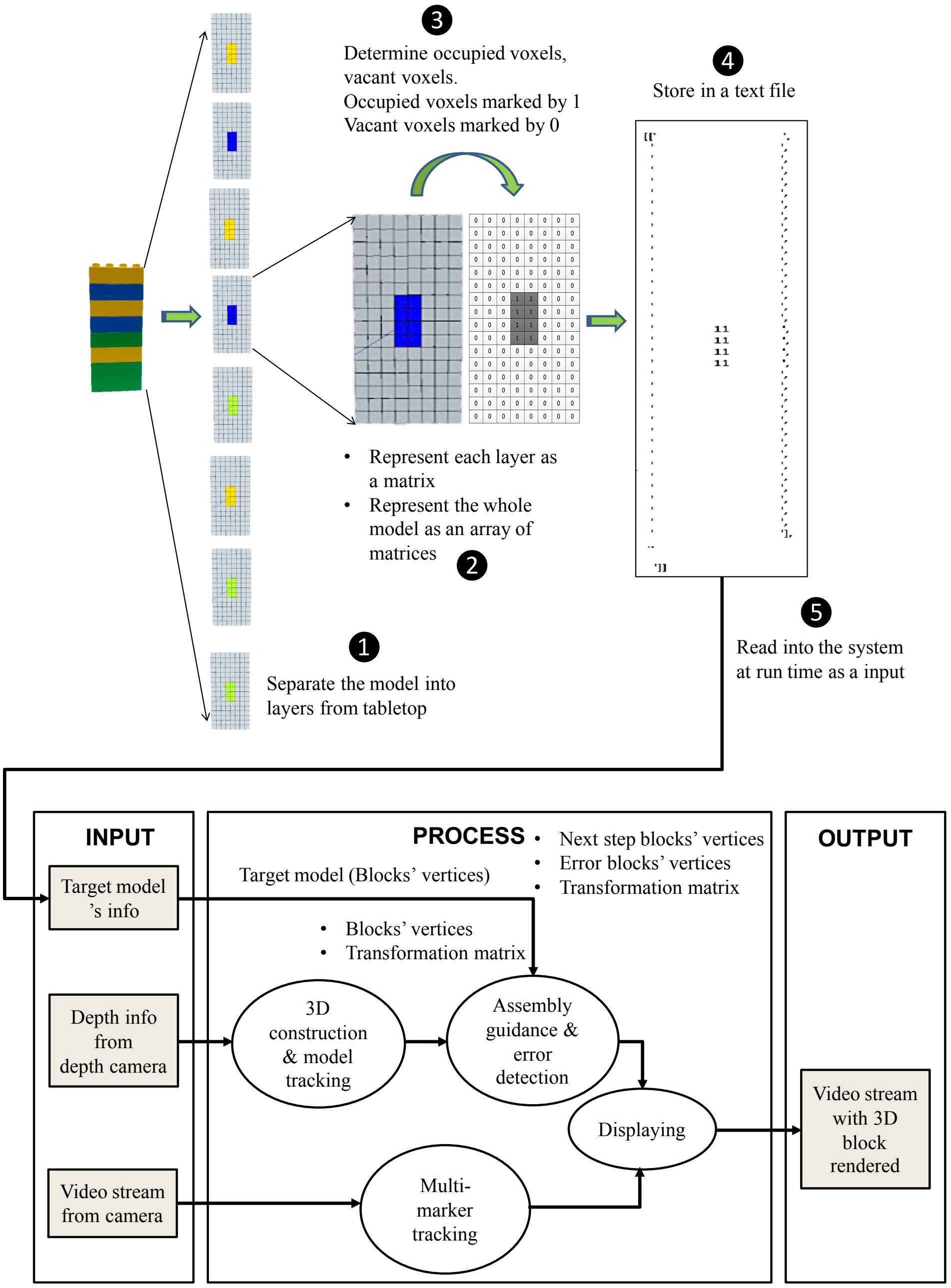

3
Determine occupied voxels, vacant voxels.
Occupied voxels marked by 1
Vacant voxels marked by 0
4
Store in a text file
• Represent each layer as a matrix
• Represent the whole model as an array of matrices
2
5
Read into the system at run time as a input
1
Separate the model into layers from tabletop
INPUT
Target model 's info
Depth info from depth camera
Video stream from camera
PROCESS
• Next step blocks' vertices
• Error blocks' vertices
• Transformation matrix
Target model (Blocks' vertices)
• Blocks' vertices
• Transformation matrix
3D construction & model tracking
Assembly guidance & error detection
Displaying
Multi-marker tracking
OUTPUT
Video stream with 3D block rendered

Figure 3.22: The information input mechanism for the target model.

relative positions. When at least one marker is visible the position of the other markers in the marker set can be easily computed.

3.7 Display Module

The display module uses marker-based pose estimation along with depth-based pose estimation to align guidance information with the physical model being assembled in the user's field of view.

Because tracking between a camera mounted on a HMD and markers and the tracking between a physical assembling model and a Kinect are conducted in the different coordinate (i.e., the marker coordinate system and the Kinect coordinate system), so to align the assembly instruction with the physical assembling model when user moves his head or move, rotate the physical model during the assembly process, the two different coordinate systems need to be considered in the same world coordinate system.

To convert the marker coordinate system to the world coordinate system, at multi-marker tracking module, the world coordinate system is defined as a set of coordinate axes aligned with the table surface at the top left of the center marker in the marker set (the marker 13^{th} among 25 markers of the makers set) with the z axis which is perpendicular to the marker. Then, it is easy to know the positions of all markers relative to the world coordinate origin. The conversion is shown in Figure 3.29 as below.

On the Kinect coordinate system, as mentioned in Section 3.4, the y axis is defined perpendicular to the tabletop (this definition takes the advantage of the fact that the size of the building blocks in x and z axes are the same (Duplo blocks : $W_x = W_z = 0.016$cm) , the translation on the table can be done just by a multiple of $W_x = W_z$). The Kinect coordinate system and the conversion to the world coordinate system is shown in Figure 3.30.

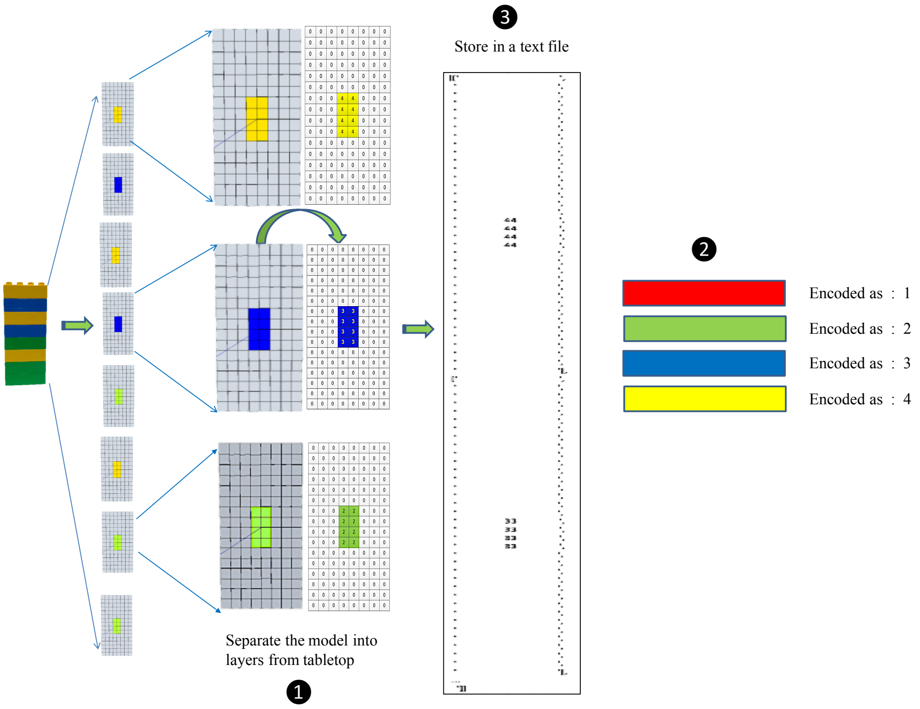


Figure 3.23: The algorithm for encoding the colors of the target model.

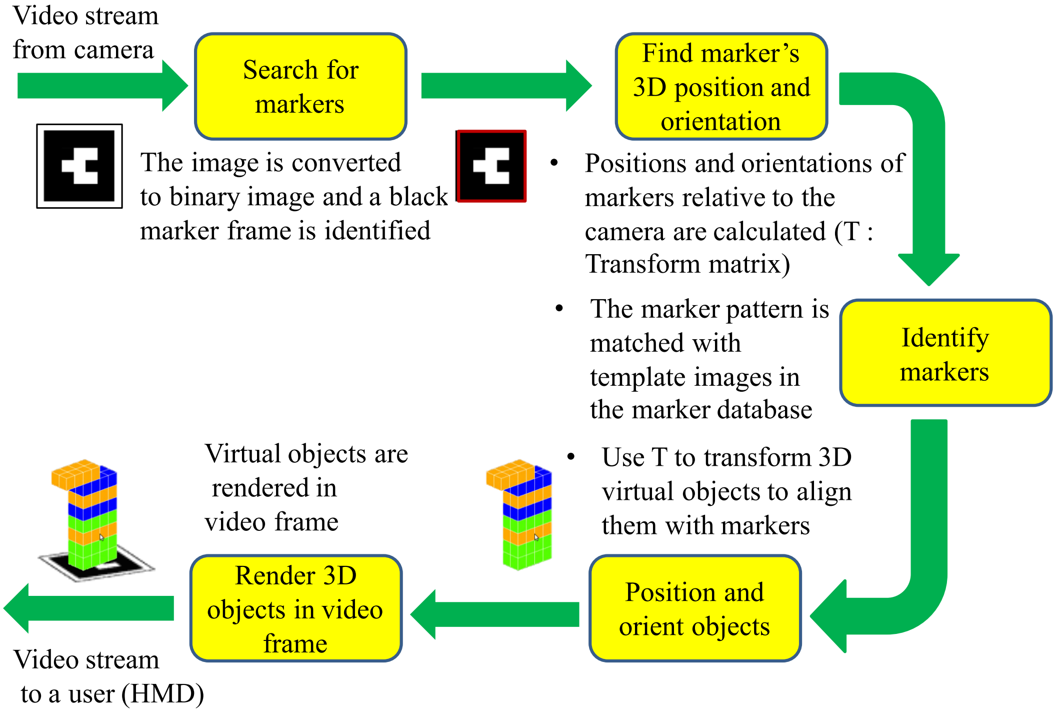

Video stream
from camera
Search for markers
The image is converted to binary image and a black marker frame is identified
Find marker's 3D position and orientation
• Positions and orientations of markers relative to the camera are calculated (T : Transform matrix)
• The marker pattern is matched with template images in the marker database
Identify markers
• Use T to transform 3D virtual objects to align them with markers
Position and orient objects
Virtual objects are rendered in video frame
Render 3D objects in video frame
Video stream
to a user (HMD)

Figure 3.24: Tracking steps in ARToolkit.

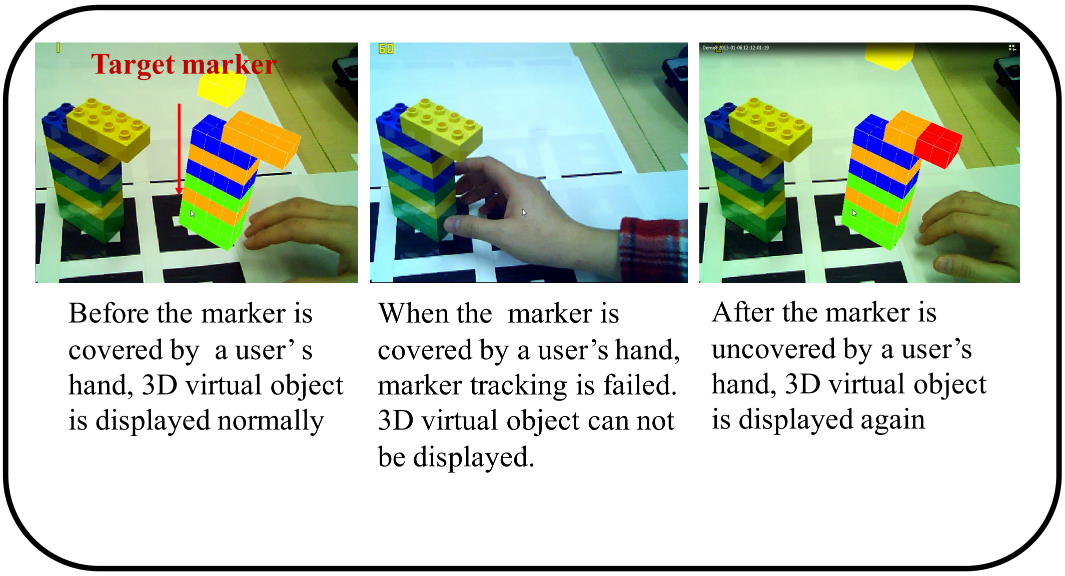

Target marker
Before the marker is covered by a user' s hand, 3D virtual object is displayed normally
When the marker is covered by a user's hand, marker tracking is failed. 3D virtual object can not be displayed.
After the marker is uncovered by a user's hand, 3D virtual object is displayed again

Figure 3.25: Hand occlusion problem when using a single marker.

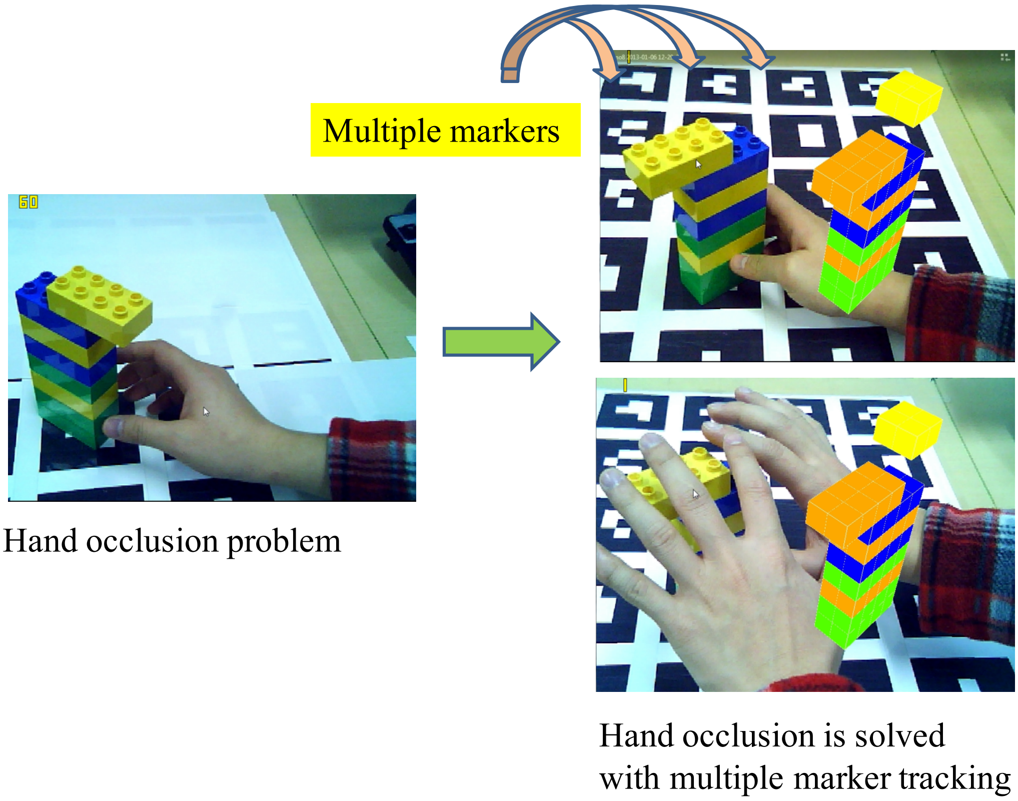


Figure 3.26: Solve the hand occlusion problem by using multi trackers.

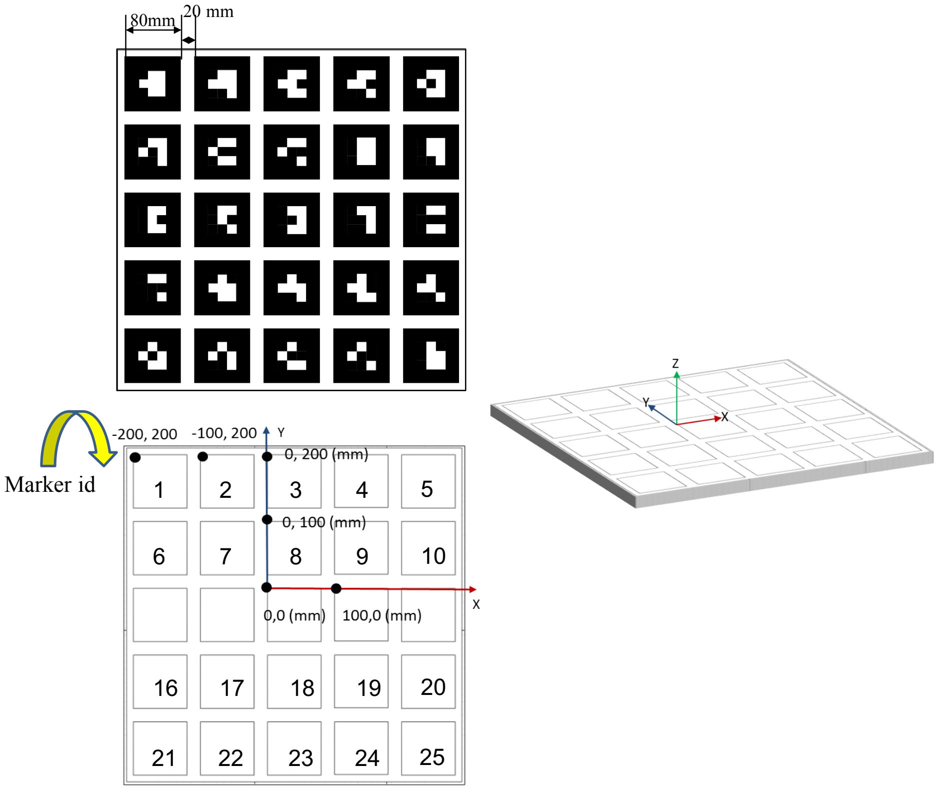
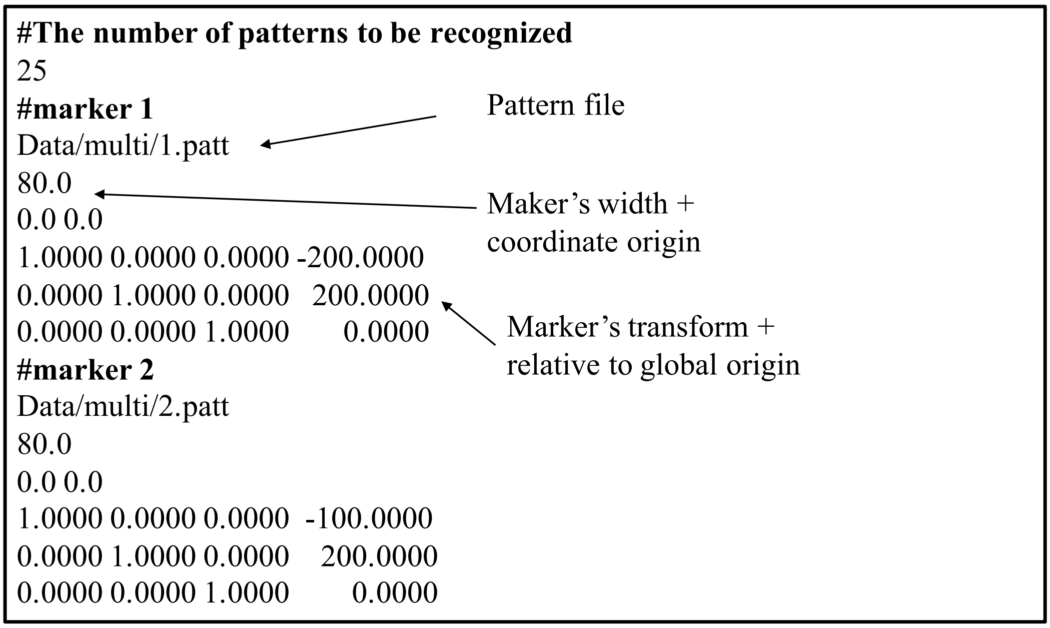


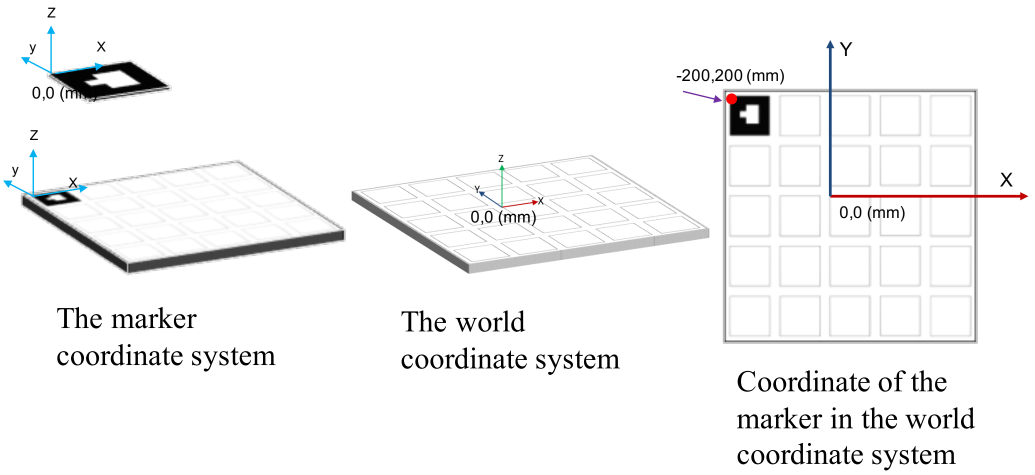
Figure 3.27: The coordinate system of the markers.



```
#The number of patterns to be recognized
25
#marker 1
Data/multi/1.patt
80.0
0.0 0.0
1.0000 0.0000 0.0000 -200.0000
0.0000 1.0000 0.0000   200.0000
0.0000 0.0000 1.0000       0.0000
#marker 2
Data/multi/2.patt
80.0
0.0 0.0
1.0000 0.0000 0.0000  -100.0000
0.0000 1.0000 0.0000    200.0000
0.0000 0.0000 1.0000        0.0000
```

Pattern file
Maker's width +
coordinate origin
Marker's transform +
relative to global origin

Figure 3.28: The configuration file of the markers.

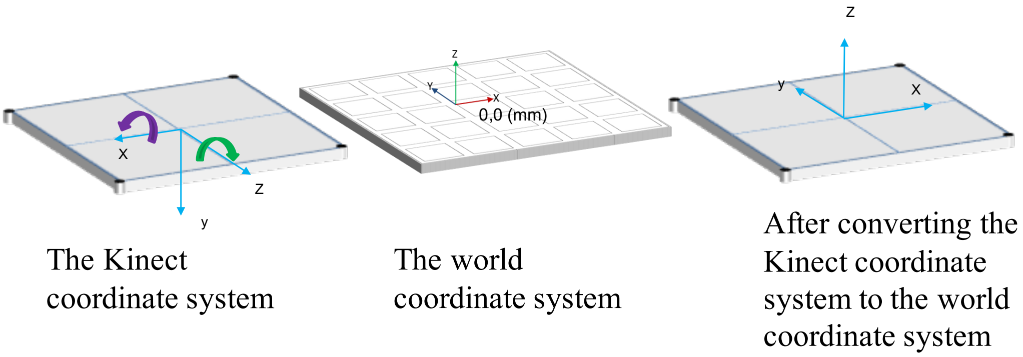


The marker coordinate system

The world coordinate system

Coordinate of the marker in the world coordinate system

Figure 3.29: Conversion from the marker coordinate system to the world coordinate system.



The Kinect coordinate system

The world coordinate system

After converting the Kinect coordinate system to the world coordinate system

Figure 3.30: Conversion from the Kinect coordinate system to the world coordinate system.

Chapter 4

The Effectiveness of AR-Based Context-Aware Visualization Techniques vs Traditional Paper Manual

4.1 Introduction

In this Chapter, we propose and compare the performance of the first two visualization modes for displaying assembly instructions and error detection information on HMD with a traditional assembly instruction style-paper manual in assembly tasks.

4.2 Full-wireframe and Side-by-side Visualization Mode Proposed

4.2.1 Design Concept

The first and most important purpose of designing the two visualization modes is to help users easier to understand assembly tasks through context-aware step-by-step guidance and conduct those assembly tasks correctly as much as possible in a poor registration context.

In a bottom-up assembly style that we consider in this study, users need to finish a lower assembly layer before they can go up to a next higher one. We believe that in this context, a combination of automatically recognizing the assembly status and suggesting users suitable next step guidances (step-by-step instructions) in the users' field of view is the best way to help user easier to understand assembly tasks even they are novice.

To help users assemble correctly as much as possible in a poor registration context, firstly we suppress misalignment between virtual guidace information and real assembly object by adding virtual reference information to the context. A relation between virtual guidance information and referecence information always be kept during the assembly process. When misalignment occurs, users can infer the correct position of the guidance information on the real object based on the reference information. Secondly, we implement

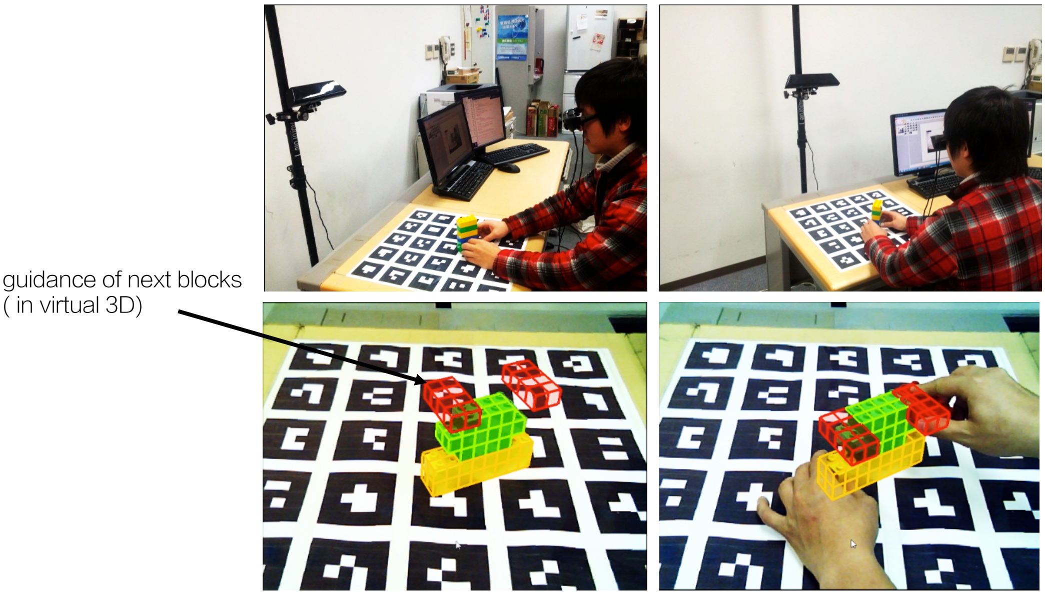


Figure 4.1: The full-wireframe overlay mode proposed.

an error detection mechanism that automatically detects assembly errors and suggests suitable guidance information corresponding to the current status to help users correct the errors quickly and finish assembly tasks correctly.

4.2.2 Proposed Techniques

Base on the design concept above, we propose two visualization modes, full-wireframe overlay mode and side-by-side mode. In the full-wireframe overlay mode (Figure 4.1), a full-writeframe that acts like reference information is added and overlaid on the real object. Assembly instructions (next step blocks) are displayed as animated color wire frames which drop down to places where the next step blocks should be attached. Blocks that were assembled incorrectly (the error blocks) are detected and the system notifies the user by highlighted red blinking wire frames.

In the side-by-side mode, a 3D virtual model of the assembly physical model is reconstructed in real time, and displayed side-by-side with the physical model on HMD within the user's field of view. Assembly instructions of the next step blocks are displayed as animated solid color cubes whose color is corresponding to that of the next step blocks. The error blocks are indicated by highlighted red blinking solid cubes. The operation of the side-by-side mode is shown in Figure 4.2.

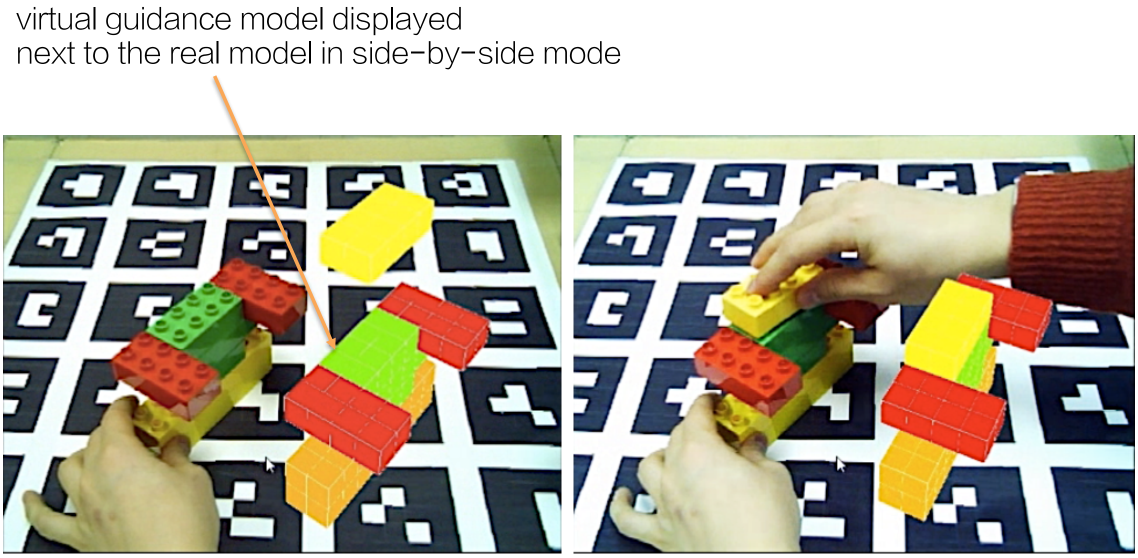

virtual guidance model displayed
next to the real model in side–by–side mode

Figure 4.2: The side-by-side mode proposed.

4.3 Evaluation

In this evaluation, we evaluate the two evaluation modes, full-wireframe overlay mode and side-by-side mode, with a traditional assembly instruction style-paper manual in Duplo block structure assembly tasks.

4.3.1 Hypotheses

The AR test-bed system displays guidance information in the form of visual feedback, such as animation, highlighting, and flashing, which should contribute beneficially to the user's overall experience. If errors are detected, such as placing parts in an incorrect location or orientation, the test-bed system can display appropriate guidance information to help the user correct them. However, users have to rotate the physical model in order to help the real-time error detection system function correctly, which takes time and slows the user down. Additionally, due to the limitations of the video see-through HMD, such as limited field-of-view, fixed focus distance, and biocular display, we anticipate that users of our system may find the system unfamiliar or encounter discomfort or fatigue.

Therefore, we make the following predictions:

- *H1: When compared to traditional media (printed manual), the test-bed system will improve accuracy and reduce the rate of assembly errors.*

- *H2: When compared to traditional instruction media (printed manual), visualization of guidance information supported by the system will be* *better in the following aspects: ease of understanding, ease of seeing, satisfaction level, and usefulness.*

- *H3: When compared to traditional media (printed manual), using the current test-bed system will not achieve better completion time of the assembly task.*
- *H4: When compared to traditional media (printed manual), using the current test-bed system will not support better stress level and familiarity.*

4.3.2 Experiment Design

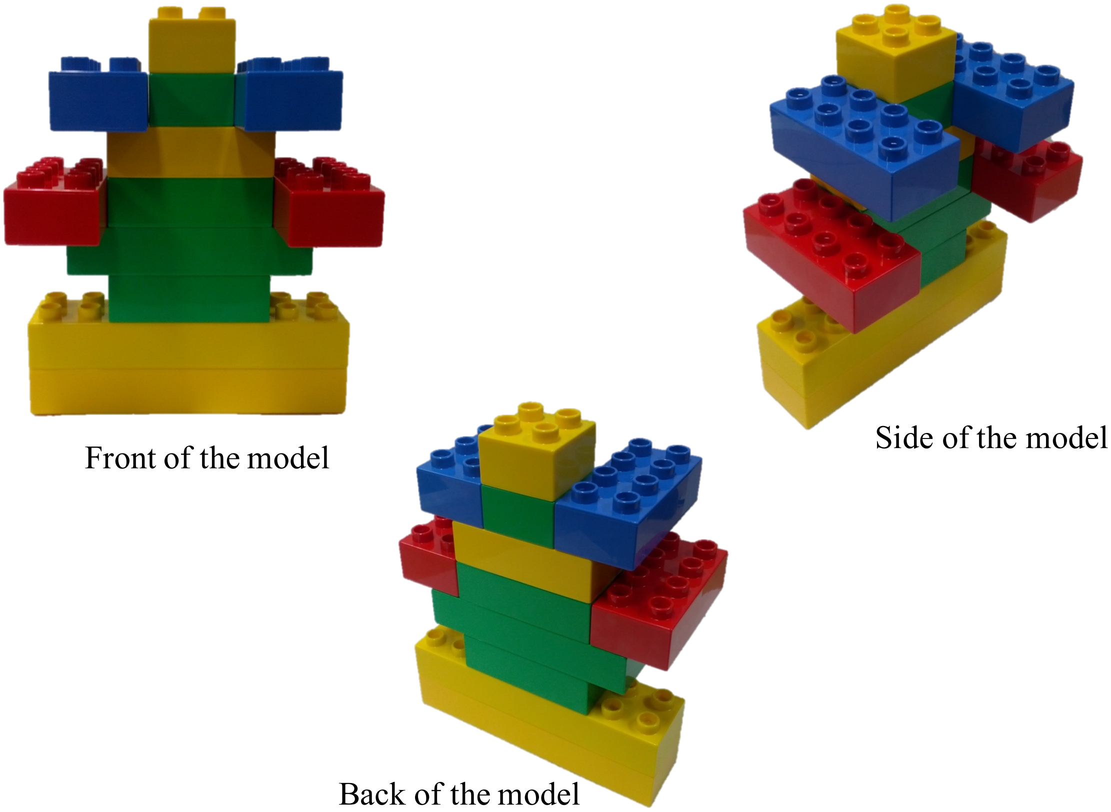
We used a three-way within-subjects experimental design, where the independent variable was the visualization mode for assembly instructions, and the dependent variables were time taken to complete the task, error rate, ease of use, ease of understanding, ease of seeing, stress level, familiarity, satisfaction, and usefulness. The independent variable ranged over three conditions: the control, a traditional printed instruction manual (Figure 4.3), the full-wireframe overlay mode and the side-by-side mode, variations of an augmented reality display. Each participant was subjected to all three conditions in a randomized order. For each condition, the participants were asked to assemble five building block (Duplo) models in randomized order (Figure 4.4).

4.3.2.1 Procedure

Before the experiment, each participant was given a tutorial on each of the conditions. After completing all fifteen assembly tasks, each participant was asked to fill out a questionnaire asking them for feedback about their experience using the questions shown in Table 4.1.

4.3.2.2 Metrics

- **Assembly task**: We use five Duplo structure models in this evaluation. Assembly of a structure model is considered as an assembly task.
- **Completion time**: Completion time of an assembly task is recorded by an observer when a participant starts an assembly task until he or she indicates that he or she has finished the assembly and the assembled model is ready for checking and evaluation. Completion time of an assembly task is measured in second unit. We will explore the mean of completion time in each visualization mode.
- **Errors**: A wrong position of any Duplo block when compared to the target model is counted as one error. In this evaluation, errors in each assembly task are counted when a participant indicates that he or she has finished the assembly and the assembled model is ready for checking



Front of the model

Side of the model

Back of the model

Figure 4.3: An example of a printed manual used to describe Model 1.

and evaluation. The whole assembled model is checked and errors are counted by the observer.

- **Scaling user preference**: We also explore the user preference for each mode based on the questionnaire mentioned above. The questionnaire consisted of 7-point ordinal scale responses, with 1 indicating the most negative response and 7 indicating the most positive response.

4.3.2.3 Subjects

Twelve people (12 male) from the author's laboratory participated in this study. The ages of participants were between 22 and 40 years. None of the participants had used any assembly support system using AR before. Three participants reported that they had previous experience assembling LEGO or Duplo block structures. Ten participants had experience with AR applications; four of these had experience with head mounted displays (HMD) in particular.

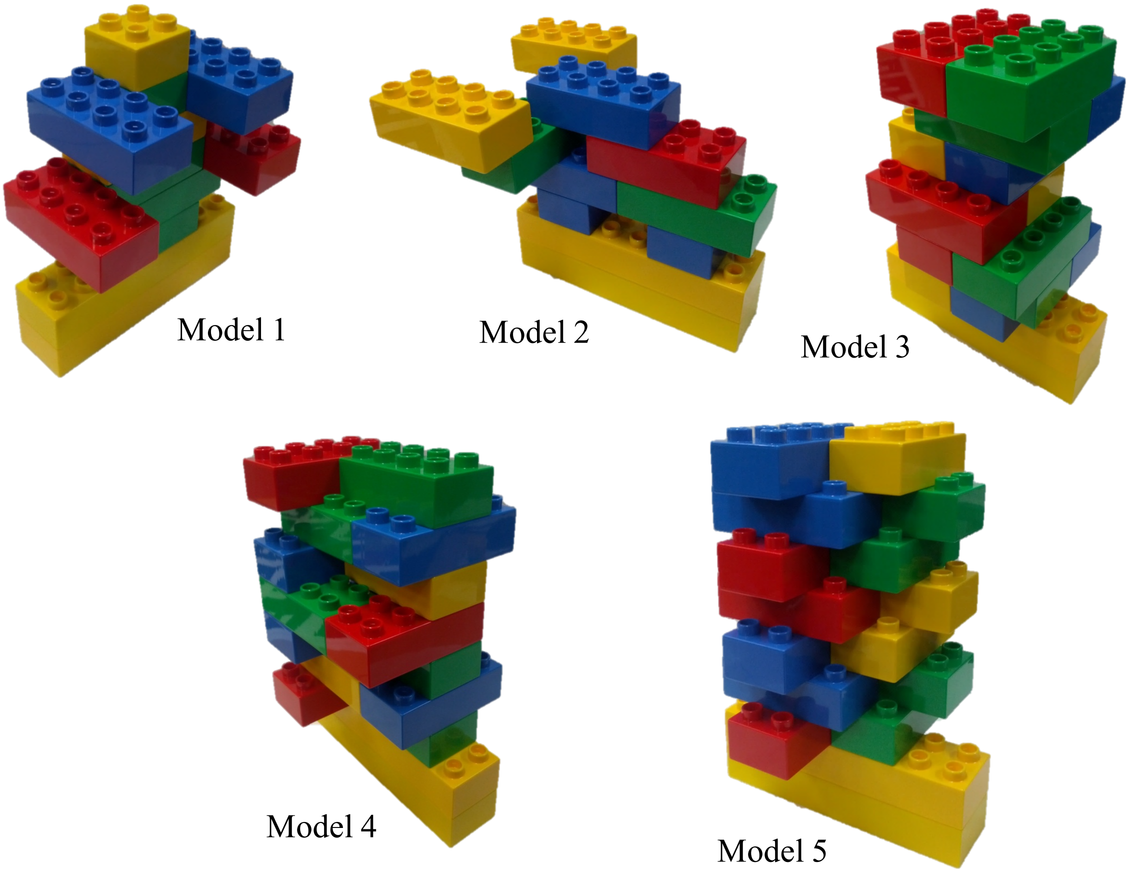


Figure 4.4: Models for the evaluation I.

4.3.3 Analysis of Quantitative Data

Figure 4.5 and Figure 4.6 indicate the overall completion time and average number of errors, respectively. Stars indicate significance levels as follows: * p<0.05, ** p<0.01, *** p<0.001. In this experiment, AR instructions did not appear to have an advantage in completion time compared with the traditional instruction media; the printed manual condition had the shortest completion time while the full-wireframe overlay mode had the longest. However, subjects in the printed manual condition occasionally made assembly errors without noticing, whereas in the AR conditions the errors were pointed out by the system and the subjects corrected them before finishing the assembly.

We conducted a repeated measure Analysis of Variance (ANOVA) test and found significant differences among the three conditions in both completion time ($F_{2,9}$ = 24.116, p < 0.0001) and mean number of errors ($F_{2,9}$ = 25.000, p < 0.0001). We conducted a post-hoc analysis using pairwise t-tests with the Holm's Bonferroni adjustment [28], and found that mean completion time was significantly shorter with a printed manual than with the side-by-side mode (t_{11} = -3.577, p < 0.025), yet shorter with the side-by-side mode than the

Table 4.1: Questionnaire for evaluating the effectiveness of conditions.

No	Question	Response Type
1	Were the assembly instructions information and error notification difficult to understand?	7-point ordinal scale (1:Difficult to understand; 7:Easy to understand)
2	Were the assembly instructions information and error notification difficult to see?	7-point ordinal scale (1:Difficult to see; 7:Easy to see)
3	Did you feel stress when using this assembly instructions media?	7-point ordinal scale (1:Feel very stressed; 7:Do not feel the stress)
4	Did you feel difficult to become familiar with the assembly instructions media?	7-point ordinal scale (1:Difficult to become familiar; 7:Easy to become familiar)
5	Did you feel satisfied with the assembly instructions media after using it?	7-point ordinal scale (1:Not satisfied at all; 7:Very satisfied)
6	Did you feel the assembly instructions media useful for the assembly tasks?	7-point ordinal scale (1:Not useful at all; 7:Very useful)

full-wireframe overlay mode (t_{11} = -3.645, $p < 0.05$). We also found that the error rate using the printed manual was significantly greater than in the other conditions (t_{11} = 5.000, $p < 0.025$) when compared with the full-wireframe overlay mode and (t_{11} =5.000, $p < 0.05$) when compared with the side-by-side mode.

4.3.4 Analysis of Questionnaire Data

Figure 4.7 shows that the side-by-side mode performed better than the printed manual specifically in the following aspects: ease of understanding, ease of seeing, satisfaction level and usefulness.

We used a non-parametric Friedman test to check for significant differences in qualitative metrics reported by participants for each condition. We found significant differences among effect of conditions in the aspect of ease of understanding (X^2 = 18.681, $p < 0.0001$). Using Wilcoxon signed rank tests with the Holm's Boneferonni correction, we found significant differences between the side-by-side mode and the paper manual (z = -3.063, $p < 0.0167$) and between the side-by-side mode and the full-wireframe overlay mode (z = -2.937, $p < 0.025$) but there was no significant difference between the full-wireframe overlay mode and the printed manual (z = -1.533, p = 0.125). The participants reported that they were easy to understand and easy to figure

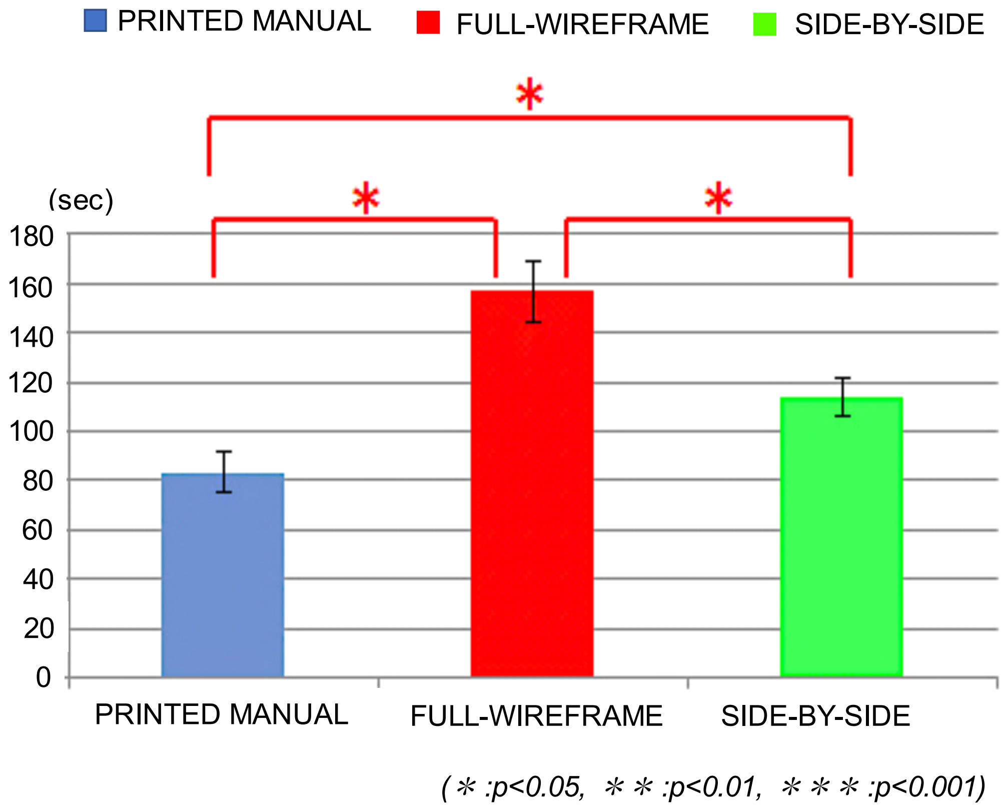


Figure 4.5: The mean of completion time of each condition in the evaluation I (the second unit). Error bars indicate 95% confidence intervals.

out what to do next when using the support of the system. We also found significant differences among effect of the conditions on the ease of seeing aspect (X^2 = 11.783, p < 0.01). Using Wilcoxon signed rank tests with the Holm's Boneferonni correction, we found significant differences between the side-by-side mode and the full-wireframe overlay mode (z = -2.941, p < 0.0167) and between the side-by-side mode and the printed manual (z = -2.672, p < 0.025) but no significant differences between the full-wireframe overlay mode and the printed manual (z = -0.534, p = 0.593). Statistical data also showed that effect of conditions in the aspect of stress level (X^2 = 6.617, p = 0.046), familiarity (X^2 = 4.638, p = 0.098), satisfaction level (X^2 = 3.957, p = 0.138) and usefulness (X^2 = 4.667, p = 0.097) were not found to be significant. However, the small p-value suggests that this difference may be significant with more participants.

4.4 Findings and Discussion

In this evaluation, the test-bed system was hypothesized to significantly improve accuracy and reduce errors of assembly tasks when compared to tradi-

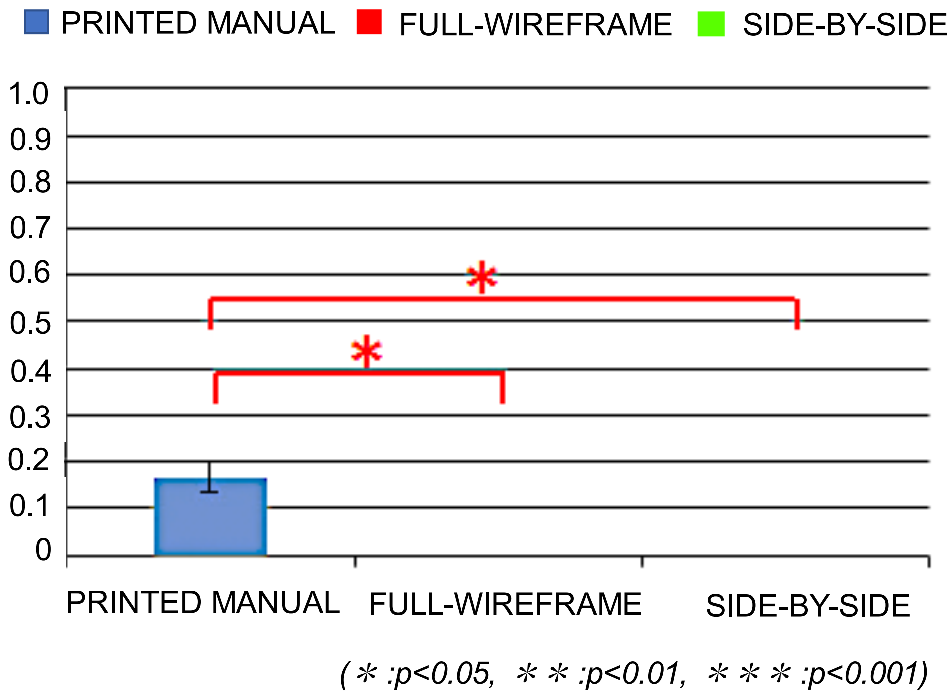


Figure 4.6: Mean number of errors per assembly task found on models after completing the assembly task. Error bars indicate 95% confidence intervals.

tional media (printed manual) (H1). The result of the experiment supported this hypothesis. With the traditional instruction media (printed manual), mistakes made by the subjects were realized when the subjects had passed several assembly steps and almost at or after the timing of completion assembly of the model. With automatic detecting the models status in real-time supported by the system, the subjects made almost no mistakes. Although there were still some mistakes which occurred by placing parts in a wrong location or incorrect orienting parts during assembly process, they were detected and notified by the system in real-time right at each assembly step and the subjects easily corrected them by following the appropriate guidance instructions of the system corresponding to the recognized states at that time.

In the aspects of ease of understanding, ease of seeing, satisfaction level and usefulness, our hypothesis stated that the guidance information modes (the full-wireframe overlay mode and the side-by-side mode) supported by the system are significantly better when compared with the traditional media (printed manual) (H2). We believed that displaying guidance information and notifying the user by visual feedback, such as animation, highlighting and flashing using AR display techniques help the subjects to see and understand guidance information. Visualization modes of the system display guidance information step-by-step visually. The result of the experiment supported this hypothesis.

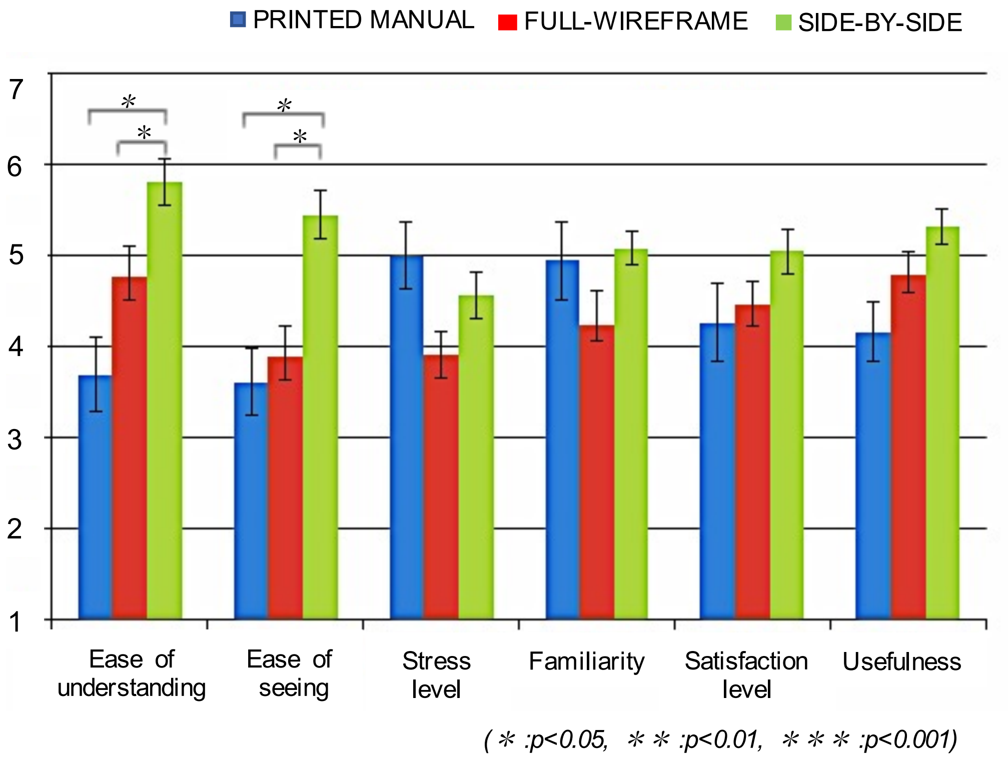


Figure 4.7: Conditions ranked on usefulness level in the evaluation I.

For completion time of the assembly tasks, the experiment's result also supported our hypothesis (H3) that using support of the test-bed system, subjects may not achieve better completion time of assembly tasks when compared with traditional media (printed manual). This result could be explained because the current test-bed system uses only one Kinect device, the subjects have to rotate the physical model to let the system construct the 3D virtual model of the model being assembled as well as recognize completion and error states on the physical model and this requires time. The system guides the user step-by-step instructions, therefore, with simple models, the subjects seem to spend more time to complete assembly tasks than using printed manual. As another reason for this result, the test-bed system uses a video see-through head mounted display (HMD) to display guidance instructions. Since the real world is digitized, and due to the lack of binocular and accommodation depth cues, the sense of distance to the model is not as good as sense of distance to the model in case of using traditional media with naked eye. This partially slowed down the subjects' assembly operations.

In the aspects of stress level and familiarity, we predicted that using the test-bed system may not support better stress level and familiarity (H4) because of disadvantages of video see-through displays include a low resolution of reality, a limited field-of-view, fixed focus distance and biocular display may cause user discomfort, eye strain and fatigue. The result of the experiment supported our prediction with the printed manual have stress level lower than the full-wireframe overlay mode but there was no statistically significant effect between the printed manual and the side-by-side mode.

4.5 Conclusion

In this Chapter, we proposed and evaluated the effectiveness of two visualization modes for an AR based context-aware assembly support system with traditional paper manual in object assembly. Our experiments showed that although subjects took longer to complete the assembly tasks using the proposed AR systems, the accuracy was dramatically improved when compared with the traditional instructions method (printed manual). Context-related visualization modes proposed were also significantly preferred the to traditional method in the following aspects: ease of understanding, ease of seeing, satisfaction level and usefulness.

Chapter 5

Overlay Visualization Techniques Improvement

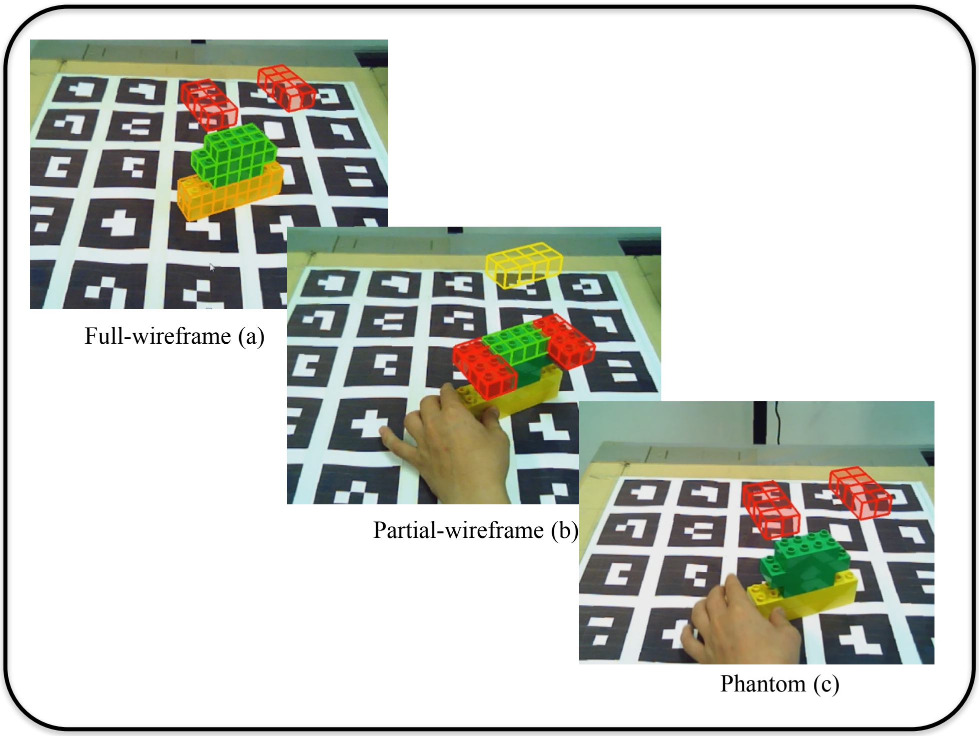
5.1 Introduction

In Chapter 5, we explore the effectiveness of a number of overlay visualization modes improved. Based on the subjects' feedbacks in the first evaluation, we propose a few new forms of the traditional AR visualization mode - the overlay mode. In the first evaluation, participants encountered two problems with the full-wireframe overlay mode. The first one is low visibility of the real object due to the overlaid full-wirefame virtual content. The second one is confusing visualization due to poor registration. These two problems may be mitigated by crafting the visualization techniques used for the overlay mode. Then, the improved overlay mode may be comparable or even superior to the side-by-side mode.

5.2 Proposed Variations of Overlay Mode

5.2.1 Design Concept

In the first evaluation, participants reported that the full-wireframe overlay mode made it easier to figure out the position of the next guide blocks under poor registration of the guidance information to the real models. However, display full virtual contents overlaid onto their real models in the full-wireframe overlay mode made them hard to see parts of the real models, so they spent more time to determine the right position of the next guide blocks on the real models. So, in the bottom-up assembly style considered in this study (one layer must be completed before starting the new layer above it), displaying only the top layer as wire frame blocks overlaid on the real counterpart blocks in addition to the next guide blocks will yield a better performance. Also in the first evaluation, some participants suggested not to render virtual blocks at all over the real counterpart model and only render the next guide blocks. It may help them to see the real blocks more easily specifically when rotating the models. However, they also agreed that they may have to pay more attention to determine the right position of the blocks being added because there will be no reference under poor registration situations.



Full-wireframe (a)

Partial-wireframe (b)

Phantom (c)

Figure 5.1: Overlay visualization modes proposed.

5.2.2 Proposed Techniques

To keep the advantage of the full-wireframe overlay mode as well as reducing the problems with it, we proposed a new visualization mode, the partial-wireframe mode (Figure 5.1b). The idea of this visualization mode is that instead of displaying the full virtual representation of real models being assembled, only a portion of the virtual representation immediately below the block being guided is displayed. This visualization mode helps users to see real models easier while it also helps them to figure out the position of the next blocks. This visualization is expected to have a better performance than the current full-wireframe overlay mode. We also proposed another visualization mode that we call the Phantom overlay mode (Figure 5.1c) based on feedback from study participants who felt that only the next guided block should be superimposed onto the real model.

5.3 Evaluation

We compare three variants of the overlay modes: full-wireframe, partial-wireframe and phantom in an assembly task experiment to determine the best overlay mode among them.

5.3.1 Hypotheses

We expect the partial-wireframe overlay mode to have the best performance because it combines the merits of the full-wireframe overlay mode and phantom overlay mode. The partial-wireframe overlay mode would be able to make it easier for users to see and determine the position of guided blocks on real models while still helping them to figure out the spatial correspondence between the guided blocks and the real models being assembled. In the phantom overlay mode, the design of only rendering the next guide blocks may have advantages in solving the low visibility problem of the real object due to the overlaid full-wirefame virtual content, the problem that experiment participants encountered in the first evaluation, but this design also has the weakness in a poor registration context due to lack of reference information.

Therefore, we have made the following hypotheses:

- *H1: The partial-wireframe will have the best performance as well as user preference among three overlay modes.*
- *H2: The phantom mode will have a better user preference specifically in easy to see aspect but the completion time of assembly tasks may not be better the partial-wireframe mode and the full-wireframe mode.*

5.3.2 Experiment Design

We used a three-way within-subjects experimental design, where the independent variable was the visualization mode for presenting assembly instructions, and the dependent variables were time taken to complete the task, ease of use, ease of understanding, ease of seeing, stress level, familiarity, satisfaction, and usefulness. The independent variable ranged over three conditions: the overlay mode from the first evaluation (the full-wireframe overlay mode), and two proposed variations, the partial-wireframe overlay mode and the Phantom overlay mode.

5.3.2.1 Procedure

We use two Duplo structure models in this evaluation, one with low complexity in structure and another one is high complexity. Each participant was subjected to all three conditions in a randomized order. For each condition, the participants were asked to assemble two building block (Duplo) models in randomized order (Figure 5.2). Before the experiment, each participant was given a tutorial on each of the conditions. The completion time in each assembly task was recorded by the observer. After completing all six assembly tasks, each participant was asked to fill out the same questionnaire from the first evaluation in Chapter 4 (Table 5.1), allowing them to give feedback about their experience with each condition.

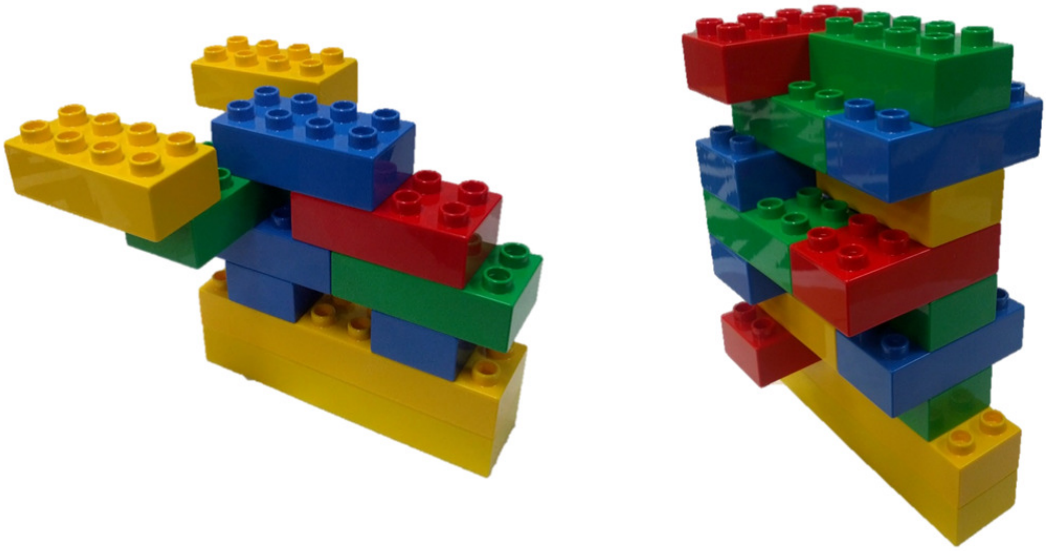


Figure 5.2: Models for the evaluation.

5.3.2.2 Metrics

- **Assembly task**: We use two Duplo structure models in this evaluation. Assembly of a structure model is considered as an assembly task.
- **Completion time**: Completion time of an assembly task is recorded by an observer when a participant starts an assembly task until he or she indicates that he or she has finished the assembly and the assembled model is ready for checking and evaluation. Completion time of an assembly task is measured in second unit. We will explore the mean of completion time in each visualization mode.
- **Scaling user preference**: We also explore the user preference for each mode based on the questionnaire mentioned above. The questionnaire consisted of 7-point ordinal scale responses, with 1 indicating the most negative response and 7 indicating the most positive response.

5.3.2.3 Subjects

Six people (3 males, 3 females) from different faculties of a co-author's university participated in this evaluation. The ages of participants were between 22 and 35 years. None of the participants had used any assembly support system using AR before. Five participants have no experience with AR and they reported that this was the first time they assembled LEGO, Duplo block structures. This evaluation was conducted on a test-bed with the same hardware setup in the first evaluation.

Table 5.1: Questionnaire for evaluating the effectiveness of conditions.

No	Question	Response Type
1	Were the assembly instructions information and error notification difficult to understand?	7-point ordinal scale (1:Difficult to understand; 7:Easy to understand)
2	Were the assembly instructions information and error notification difficult to see?	7-point ordinal scale (1:Difficult to see; 7:Easy to see)
3	Did you feel stress when using this assembly instructions media?	7-point ordinal scale (1:Feel very stressed; 7:Do not feel the stress)
4	Did you feel difficult to become familiar with the assembly instructions media?	7-point ordinal scale (1:Difficult to become familiar; 7:Easy to become familiar)
5	Did you feel satisfied with the assembly instructions media after using it?	7-point ordinal scale (1:Not satisfied at all; 7:Very satisfied)
6	Did you feel the assembly instructions media useful for the assembly tasks?	7-point ordinal scale (1:Not useful at all; 7:Very useful)

5.3.3 Analysis of Quantitative Data

Figure 5.3 illustrates the mean time of completion for each condition in the pilot study. The partial-wireframe overlay condition had the shortest completion time, while the Phantom overlay mode had the longest. We conducted a repeated measure ANOVA test and found differences in completion among the conditions ($F_{2,3}$ =7.737, p < 0.009). Using post-hoc pairwise t-tests, we found differences between the partial-wireframe overlay mode and the Phantom overlay mode (t_5 = -2.992, p = 0.03) and between the full-wireframe overlay mode and the Phantom overlay mode (t_5 = -2.710, p = 0.042). However, due to the Holm's Bonferroni adjustment, these differences were not statistically significant.

In this evaluation, mean number of errors per assembly task found on models after completing the assembly task were not considered because the automatic error detection function of the test-bed system helped users to find and fix all errors occurred during assembly process.

5.3.4 Analysis of Questionnaire Data

Figure 5.4 shows that the partial-wireframe overlay mode had a better performance than the Phantom overlay mode specifically in aspects: ease of understanding and usefulness.

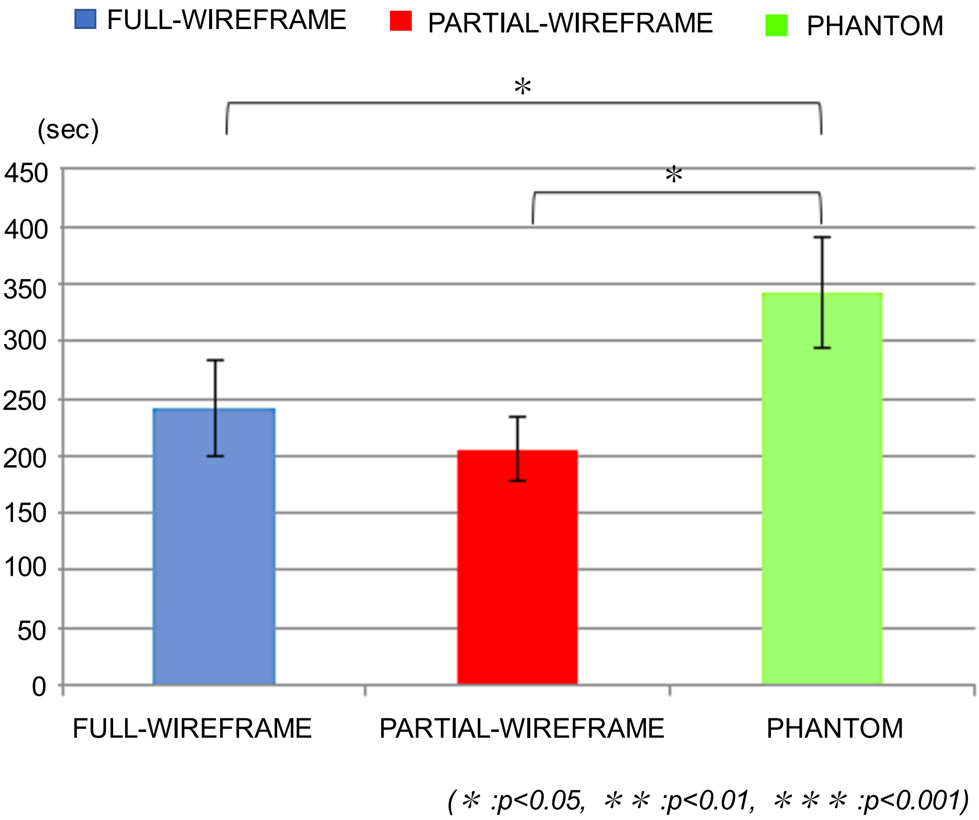


Figure 5.3: The mean of completion time of each condition in the evaluation II (the second unit). Error bars indicate 95% confidence intervals.

We used a non-parametric Friedman test to check for significant differences in qualitative metrics reported by participants for each condition. We found significant differences among conditions in the aspect of ease of understanding ($X^2 = 9.238$, p < 0.01) and in the aspect of usefulness ($X^2 = 9.238$, p < 0.01) but no significant differences on ease of seeing, stress level, familiarity and satisfaction level. We performed a post-hoc analysis with Wilcoxon signed rank tests with the Holm's Boneferonni correction on ease of understanding and usefulness aspects to uncover interesting patterns. For ease of understanding, we found differences between the partial-wireframe overlay mode and the Phantom overlay mode ($X^2 = 2.214$, p = 0.027) and between the full-wireframe overlay mode and the Phantom overlay mode ($X^2 = 2.032$, p = 0.042). For usefulness, we found differences between the partial-wireframe overlay mode and the Phantom overlay mode ($X^2 = 2.207$, p = 0.027) and between the full-wireframe overlay mode and the Phantom overlay mode ($X^2 = 2.041$, p = 0.041). Due to the Holm's Bonferroni adjustment, these differences were not found to be significant. However, small p-value suggests that these differences may be significant with more participants.

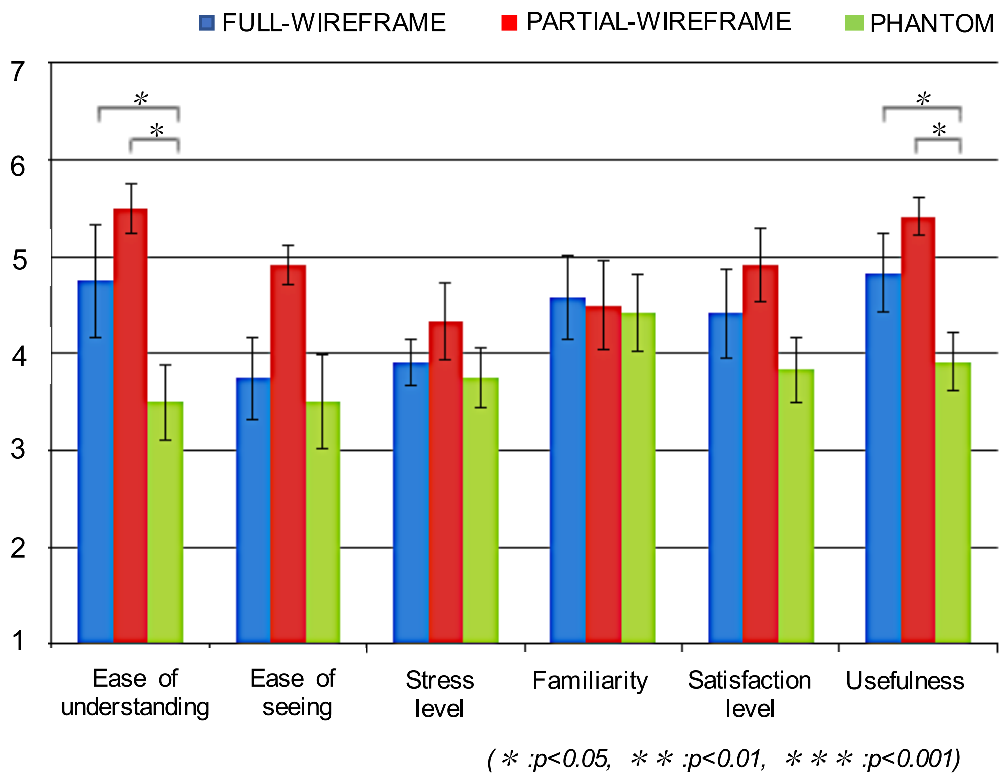


Figure 5.4: Conditions ranked on usefulness level in the evaluation II.

5.4 Finding and Discussion

In this evaluation, although we expected the partial-wireframe overlay mode to have the best performance among three visualization modes as we declared in the hypothesis H1, the statistical analysis of this evaluation did not support it. We did not find a significant difference in performance of the partial-wireframe among the three modes. The result can be explained due to a small sample size in both number of experiment models, complexity of them, as well as in number of experiment participants. However, with small p-value, we believe that the result will become significant with a bigger sample size. In the user preference aspects, five of the six participants preferred the partial-wireframe mode over the other two. They reported that the partial-wireframe overlay mode makes it easier for them to see and determine the position of guided blocks on real models while still helping them to figure out the spatial correspondence between the guided blocks and the real models being assembled.

In the hypothesis H2, we predicted that the phantom mode will have a better user preference specifically in easy to see aspect but the completion time of assembly task may not be better the partial-wireframe and full-wireframe modes, the experimental results supported it. The phantom overlay mode was reported to be relatively easy to see, but made it difficult to determine the correct location to place the next piece. The full-wireframe overlay mode got the lowest evaluation among three modes. The participants reported that because of misalignment and the potential to mistake between some parts of the real model and parts of virtual representation due to the same color, they spent more time to try to determine the position of next blocks in this mode.

5.5 Conclusion

In the current evaluation, we explored a better overlay visualization method and found that the partial-wireframe overlay method was the best among three overlay visualization modes proposed. As a follow-up study, we will conduct a comparative experiment between the side-by-side and the partial-wireframe overlay modes [33].

Chapter 6

Partial-wireframe and Side-by-side Visualization Modes

6.1 Introduction

In previous chapters, we found that each of the two visualization modes (Figure 6.1), partial-wireframe overlay mode and side-by-side mode, is better in task completion time and user preference than a naive overlay mode where a full wire-frame representation of the virtual model rendered overlaying the real model. In the partial-wireframe (Figure 6.1a), the assembly instructions (the next blocks to add) are displayed as 3D animated wireframe blocks, directly overlaying the physical model. The blocks are animated with a downward motion that suggests to snap the blocks in place. A partial-wireframe containing just the topmost layer of the virtual model is additionally rendered overlaying the actual model. The display of the partial-wireframe has the potential to reduce spatial ambiguity by overlaying instructions directly on the real model while it also helps users to figure out the correct position of next blocks in poor alignment situations. In the side-by-side mode (Figure 6.1b), the assembly instructions are displayed on top of a solid virtual model that is rendered to the side of the actual model, but in the same orientation. The primary objective of the evaluation in this chapter is to compare the two visualization modes proposed in order to provide design suggestions about how to best assist users in assembly tasks. A secondary goal of the evaluation is to determine the effectiveness of context-aware error detection. We modified the test-bed so that the error detection mechanism could be enabled or disabled, and used this as a second independent variable.

6.2 Evaluation

In this evaluation, we explore effectiveness of two modes: the side-by-side mode that is the best visualization mode proposed in the first evaluation and the partial-wireframe overlay mode, the best performance we found in the second evaluation. We also explore the effectiveness of error detection in combination with the two modes above.

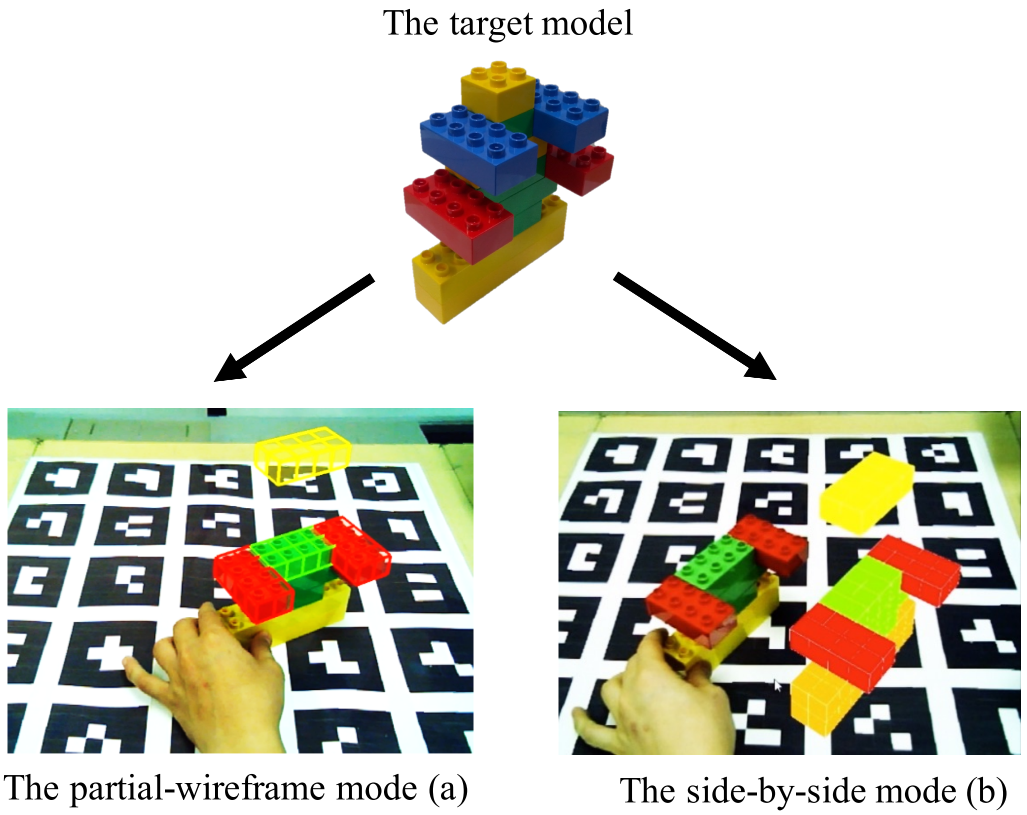


Figure 6.1: Compared visualization modes.

6.2.1 Hypotheses

With error detection enabled, the user needs to rotate the model and the system will automatically detect completion status as well as errors and notify the user in real-time. With error detection disabled, users do not need to rotate the model. Instead, they can use a left mouse button to forward assembly steps and right mouse button to backward assembly steps. However, users have to check and determine completion state as well as errors at each step by themselves. The partial-wireframe overlay mode can have problems with misalignment and users may need to pay more attention and time to determine the right position of blocks carefully. The side-by-side mode with a colorful, solid (not 80% transparent as in the partial-wireframe overlay mode) virtual representation displayed beside the real models not only avoids effects of misalignment but also makes it easier for users to see and determine the next guided blocks position. So, this mode can help users shorten the completion time, lower stress to finish the models and should expect the side-by-side mode to be useful in assembly tasks.

We therefore make the following predictions:

- *H1: The partial-wireframe overlay mode will not achieve better comple-*

tion time of assembly tasks when compared to the side-by-side mode.

- *H2: Visualization modes without error detection will achieve better completion time of assembly tasks but achieve higher error rates when compared to visualizations modes with support of error detection.*

- *H3: When compared to the partial-wireframe overlay mode, the side-by-side mode is significantly better in aspects: ease of understanding, ease of seeing, stress level, familiarity, satisfaction level and usefulness.*

6.2.2 Experiment Design

We used a 2x2 factorial within-subjects experimental design. The independent variables were the visualization mode (called VIS factor) for presenting assembly instructions (the side-by-side mode or the partial-wireframe overlay mode) and whether or not error detection was enabled (called ERR factor). The dependent variables were the time taken to complete the task, number of errors, ease of use, ease of understanding, ease of seeing, stress level, familiarity, satisfaction, and usefulness.

6.2.2.1 Procedure

We use two Duplo structure models in this evaluation. All twenty-four participants were subject to all four conditions in a randomized order. For each condition, the participants were asked to assemble two building block (Duplo) models in randomized order (Figure 6.2). We abbreviate the four conditions as follows: S_ON: the side-by-side mode with error detection, S_OFF: the side-by-side mode without error detection, PW_ON: the partial-wireframe overlay mode with error detection and PW_OFF: the partial-wireframe overlay mode without error detection. After finishing each assembly task, each participant was asked to fill out a questionnaire asking them for feedback about their experience with each condition using the questions shown in Table 6.1.

6.2.2.2 Metrics

In this evaluation, completion time, number of errors made during assembly, and number of errors present in the completed assembled model were recorded for each assembly task.

- **Assembly task**: We use two Duplo structure models in this evaluation. Assembly of a structure model is considered as an assembly task.

- **Completion time**: Completion time of an assembly task is recorded by an observer when a participant starts an assembly task until he or she indicates that he or she has finished the assembly and the assembled model is ready for checking and evaluation. Completion time of an

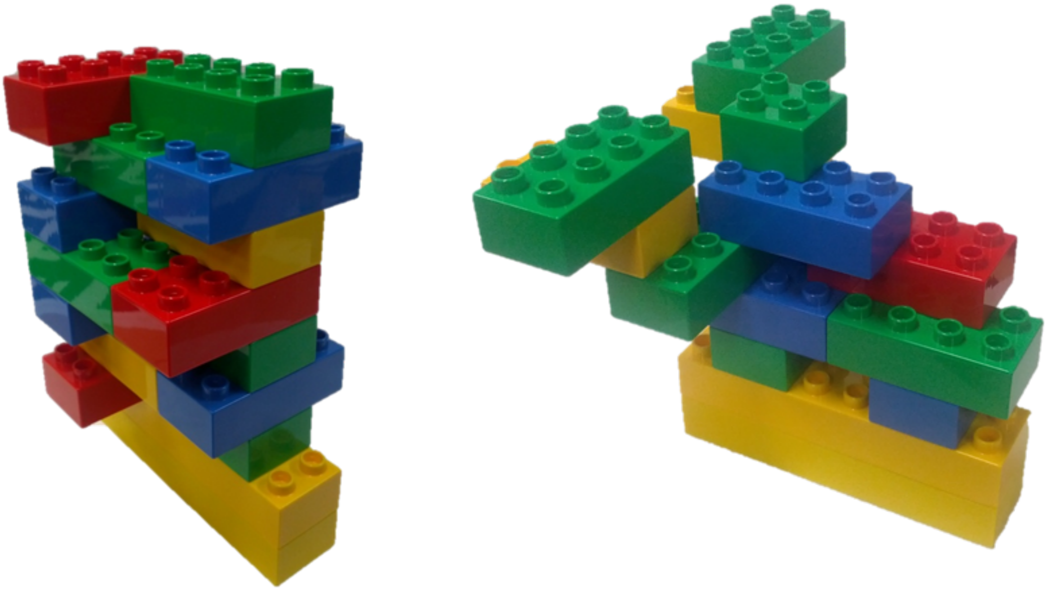


Figure 6.2: Assembly task models for the evaluation.

assembly task is measured in second unit. We will explore the mean of completion time in each visualization mode.

- **Errors during assembly process**: In aspect of number of errors made during assembly, a wrong position of any Duplo block is detected by the system and confirmed by the observer is counted as one error. Errors are counted when a participant starts an assembly task until he or she indicates that he or she has finished the assembly task.

- **Errors after assembly process**: In aspect of number of errors after assembly process, a wrong position of any Duplo block when compared to the target model is counted as one error. These errors are started to count when a participant indicates that he or she has finished the assembly and the assembled model is ready for checking and evaluation. The whole assembled model is checked and errors are counted by the observer.

- **Scaling user preference**: We also explore the user preference for each mode based on the questionnaire mentioned above. The questionnaire consisted of 7-point ordinal scale responses, with 1 indicating the most negative response and 7 indicating the most positive response.

6.2.2.3 Subjects

We recruited 24 people (12 males, 12 females) from many different departments of a co-author's university for participation in this evaluation. They

Table 6.1: Questionnaire for evaluating the effectiveness of conditions.

No	Question	Response Type
1	Were the assembly instructions information and error notification difficult to understand?	7-point Likert (1:Difficult to understand; 7:Easy to understand)
2	Were the assembly instructions information and error notification difficult to see?	7-point Likert (1:Difficult to see; 7:Easy to see)
3	Did you feel stress when using this assembly instructions media?	7-point Likert (1:Feel very stressed; 7:Do not feel the stress)
4	Did you feel difficult to become familiar with the assembly instructions media?	7-point Likert (1:Difficult to become familiar; 7:Easy to become familiar)
5	Did you feel satisfied with the assembly instructions media after using it?	7-point Likert (1:Not satisfied at all; 7:Very satisfied)
6	Did you feel the assembly instructions media useful for the assembly tasks?	7-point Likert (1:Not useful at all; 7:Very useful)

are from five countries in three continents. The ages of participants were between 22 and 35 years. Only 5 participants had prior experience with AR. None of the participants had used any AR assembly support system before. Sixteen participants reported that this is the first time they used LEGO or Duplo blocks.

6.2.3 Analysis of Quantitative Data

Figure 6.3, 6.4 and 6.5 illustrate respectively the mean time of completion, errors made during the task, and errors present in the completed model. Stars indicate significance levels as follows: * $p<0.05$, ** $p<0.01$, *** $p<0.001$. The side-by-side mode resulted in a shorter completion time than partial-wireframe mode, irrespective of the error detection mechanism. Among the conditions, S_OFF had the shortest time of completion while PW_OFF had the longest. Subjects using the partial-wireframe overlay mode made more errors than with the side-by-side mode. Without error detection, participants rarely noticed errors they made during the assembly process; however, with error detection enabled, participants recognized and fixed all errors by themselves during the assembly process.

We conducted a repeated measure ANOVA test among these conditions. In mean completion time, we found significance for VIS factor ($F_{1,23} = 27.311$,

Table 6.2: Significant results from analysis of questionnaire data.

	Friedman	S_ON > PW_ON	S_ON > PW_OFF	S_OFF > PW_ON	S_OFF > PW_OFF
Ease of understanding	$X^2 = 43.590$, $p < 0.0001$	z = 4.311, p < 0.0083	z = 3.999, p < 0.01	z = 3.519, p < 0.0125	z = 3.646, p < 0.0167
Ease of seeing	$X^2 = 45.069$, $p < 0.0001$	z = 3.921, p < 0.0083	z = 4.141, p < 0.01	z = 3.928, p < 0.0125	z = 3.714, p < 0.0167
Stress level	$X^2 = 48.433$, $p < 0.0001$	z = 4.120, p < 0.0083	z = 4.026, p < 0.01	z = 3.652, p < 0.0125	z = 3.839, p < 0.0167
Familiarity	$X^2 = 54.041$, $p < 0.0001$	z = 4.301, p < 0.0083	z = 4.144, p < 0.01	z = 3.912, p < 0.0125	z = 3.829, p < 0.0167
Satisfaction	$X^2 = 50.653$, $p < 0.0001$	z = 4.298, p < 0.0083	z = 4.242, p < 0.01	z = 3.611, p < 0.0125	z = 4.048, p < 0.0167
Usefulness	$X^2 = 45.646$, $p < 0.0001$	z = 4.300, p < 0.0083	z = 4.156, p < 0.01	z = 3.385, p < 0.0125	z = 3.758, p < 0.0167

p $<=$ 0.0001, $\eta_p^2 = 0.543$, OP $= 0.999$) and ERR factor ($F_{1,23} = 7.03$, p $<=$ 0.05, $\eta_p^2 = 0.234$, OP $= 0.719$) but did not find significance for the interaction between VIS and ERR (VISxERR) ($F_{1,23} = 0.102$, p $<=$ 0.752, $\eta_p^2 = 0.004$, OP $= 0.061$). In error rate during assembly, we fould significance for VIS factor ($F_{1,23} = 9.364$, p $<=$ 0.006, $\eta_p^2 = 0.289$, OP $= 0.834$), but did not find significance for ERR factor ($F_{1,23} = 0.276$, p $<=$ 0.605, $\eta_p^2 = 0.012$, OP $= 0.080$) and VISxERR ($F_{1,23} = 0.284$, p $<=$ 0.599, $\eta_p^2 = 0.012$, OP $= 0.080$). In error rate after completion, significance were found for VIS factor and ERR factor ($F_{1,23} = 4.713$, p $<=$ 0.05, $\eta_p^2 = 0.170$, OP $= 0.548$) and ($F_{1,23} = 14.57$, p $<=$ 0.001, $\eta_p^2 = 0.388$, OP $= 0.955$) as well as interaction between them ($F_{1,23} = 4.173$, p $<=$ 0.05, $\eta_p^2 = 0.0170$, OP $= 0.548$). We ran post-hoc analyses using pairwise t-tests with the Holm's Bonferroni adjustment to isolate the significant differences [27]. We found that the mean completion time for S_OFF was significantly faster than S_ON ($t_{23} = 1.426$, p < 0.05), and that each of these modes were significantly faster than either PW_ON (S_ON: t_{23} = -3.415, p < 0.0125; S_OFF: t_{23} = -5.176, p < 0.0083) or PW_OFF (S_ON: t_{23} = -2.460, p < 0.025; S_OFF: t_{23} = -4.773, p < 0.01). We found significant differences in the error rate during assembly between the S_ON and the PW_OFF conditions (t_{23} = -2.932, p < 0.0083) and between the PW_ON and the PW_OFF conditions (t_{23} = -2.916, p < 0.01). For error rate after completion we found significant differences between S_ON and the PW_OFF (t_{23} = -3.317, p < 0.0083) and between the PW_ON and the PW_OFF (t_{23} = -3.137, p < 0.01). Due to the Holm's Bonferroni adjustment, the differences between S_ON and the S_OFF (t_{23} = -2.015, p = 0.056) and between the PW_ON and the S_OFF (t_{23} = -2.015, p = 0.056) were not found to be significant; however, the small p_value suggests that these differences may be significant with more participants.

6.2.4 Analysis of Questionnaire Data

Figure 6.6 shows the ranking that participants provided for each condition in each aspect: ease of understanding, ease of seeing, stress level, familiarity, satisfaction level and usefulness.

Based on the rankings elicited from the participants, the side-by-side mode had better performance than partial-wireframe overlay mode in every qualitative aspect (Table 6.2). We used a non-parametric Friedman test to check for significant differences in each qualitative metric among the conditions,

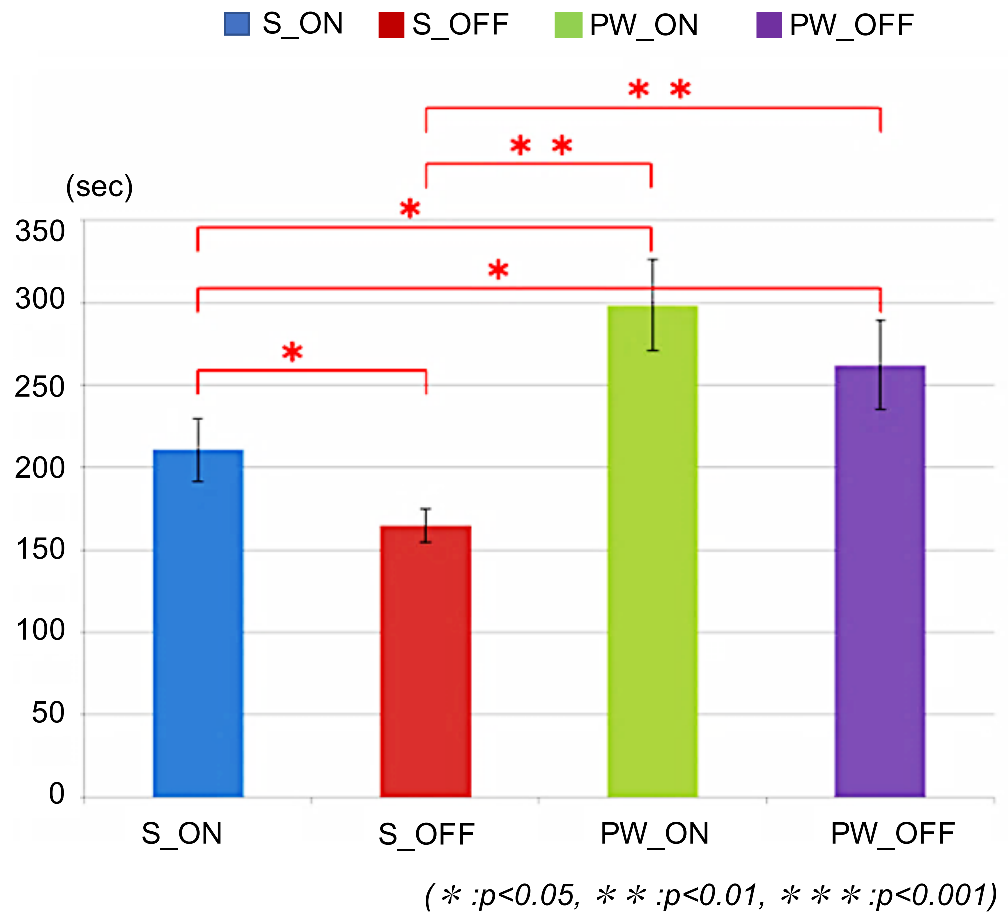


Figure 6.3: The mean of completion time of each condition in the evaluation (the second unit). Error bars indicate 95% confidence intervals.

followed by Wilcoxon signed rank tests.

6.3 Findings and Discussion

In this evaluation, we found that the side-by-side mode had the shorter completion time than the partial-wireframe overlay mode regardless of the presence of error detection. This finding supported our hypothesis (H1). Participants reported that because of misalignment and the potential to mistake some parts of the real model for parts of the virtual representation of the same color, they spent more time trying to determine the position of next blocks as they did in the side-by-side mode. It may still be the case that AR has the potential to reduce spatial ambiguity by overlaying instructions directly on the real model, however it seems that this is highly sensitive to misalignment, latency, or conflicting depth cues. At least for our test-bed, having a spatial separation between the virtual model and the real model led to significantly

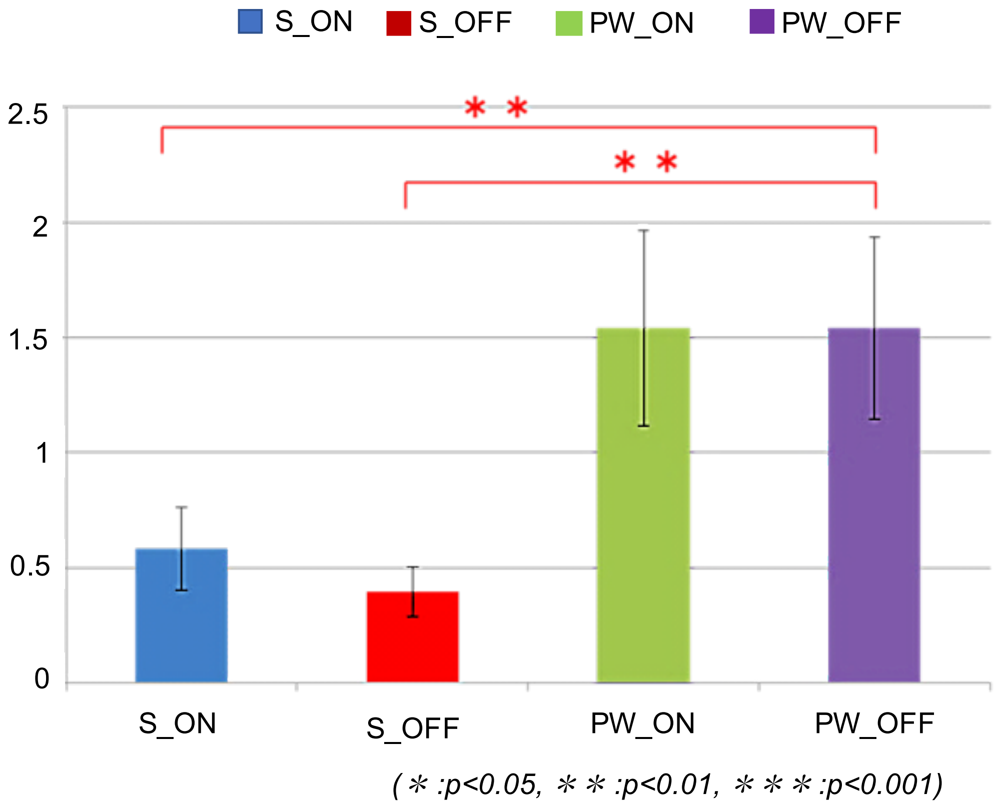


Figure 6.4: Mean number of errors per assembly task of each condition during the assembly process. Error bars indicate 95% confidence intervals.

better performance in every aspect (H3).

The result of this evaluation also showed that S_OFF (the side-by-side mode without error detection) had shorter completion time than S_ON (the side-by-side mode with error detection) and this difference was significant. On the other hand, S_OFF had higher error rates than S_ON after assembly finished (p=0.056). Although the difference was not found to be significant due to the Holm's Bonferroni adjustment, with the small p-values we believe it may be significant with more participants and more complex models. Hypothesis (H2) was therefore only partially supported.

6.4 Conclusion

In this Chapter, we evaluated the effectiveness of an AR-based context-aware assembly support system with different AR visualization modes proposed for object assembly, the overlay mode that display guidance information directly overlaid on the physical model, and the side-by-side mode in which guidance information is rendered on a virtual model that is shown adjacent to the real model. We found, somewhat surprisingly, that the visualization mode that

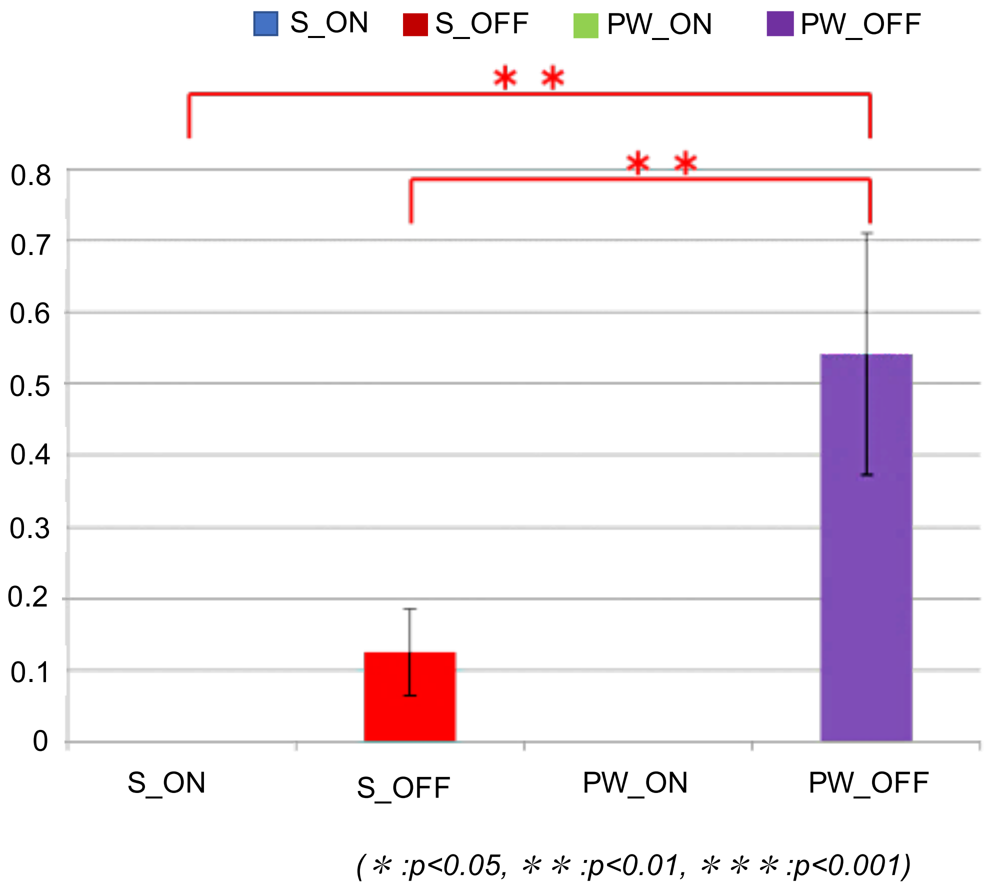


Figure 6.5: Mean number of errors per assembly task found on models after completing the assembly task. Error bars indicate 95% confidence intervals.

renders guidance information on a virtual model separate from the physical model (the side-by-side mode) was preferable in every aspect to displaying such information directly overlaying the physical model. This result is also held irrespective of whether we enabled a context-aware error detection mechanism.

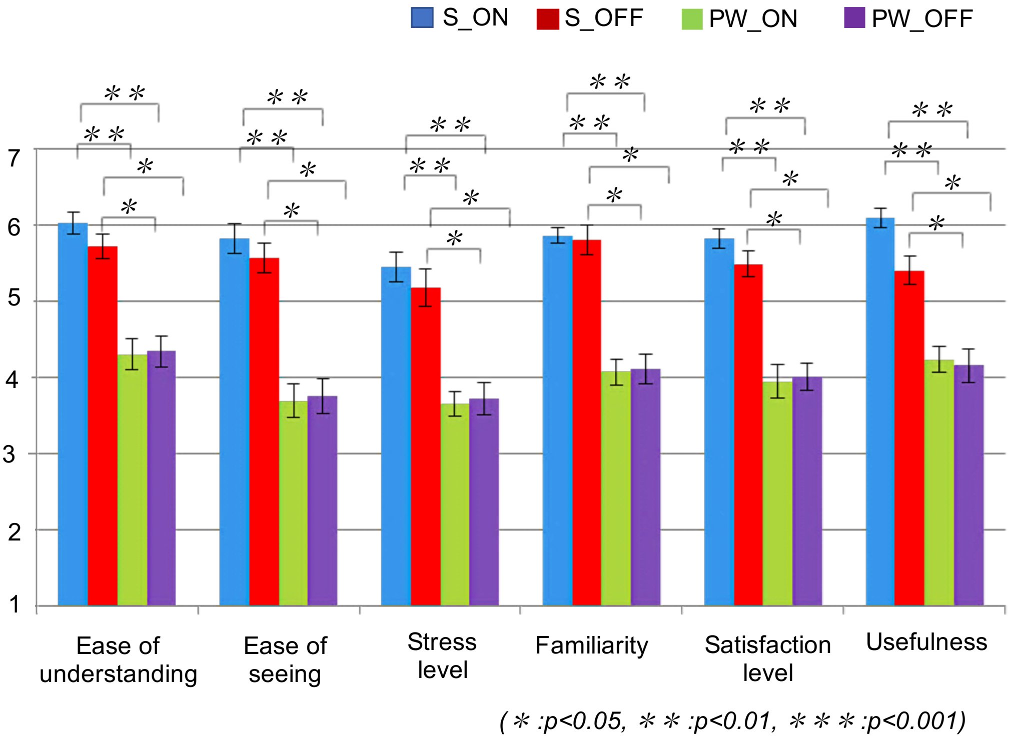


Figure 6.6: Conditions ranked on usefulness level in the evaluation.

Chapter 7
Hybrid Object- and Screen-Stabilized Visualization Modes

7.1 Introduction

In previous chapters, we found that the side-by-side mode (Figure 7.1) outperforms traditional direct overlay under moderate registration accuracy with marker-based head tracking and RGBD camera-based object tracking.

However, the side-by-side mode is not always available. In the context of large assembly models assembled with narrow field of view head mounted displays (HMDs), like those commonly used in assembly support systems, virtual guidance information is often out of the viewport of the HMD screen and assembly subjects need to move their head out of the workspace to find the guidance. This increases head and eye movement, cost of attention switching, and difficulties for spatial perception and thus it likely impacts the completion time of assembly tasks. Large assembly models in this context are models that do not fit within the field of view of the HMD at a normal reaching distance in assembly.

Most of visualization techniques introduced in the assembly support systems mentioned above are object-stabilized and it is the case of the side-by-side mode. In object-stabilized visualization techniques, information is affixed to the assembly objects themselves and its apparent position on screen changes as the user moves his or her head (Figure 7.2). This requires the user's viewpoint position and orientation to be tracked. Object stabilized information presentation also enables annotation of the real world with context dependent visual and audio data, creating information enriched environments. This can increase the intuitiveness of the real world tasks [15].

Despite the advantages of object stabilized visualization techniques, some assembly support systems only use head stabilized (screen stabilized) information display (Figure 7.3). Baird and Barfield [14] presented a system with screen fixed instructions on untracked monocular OST and opaque HMDs to support a computer motherboard assembly task. In the screen stabilized visualization techniques, information is fixed to the user's screen and it does not change as the user moves his or her viewpoint. Therefore, guidance information is always available to users and he or she can refer to it very quickly.

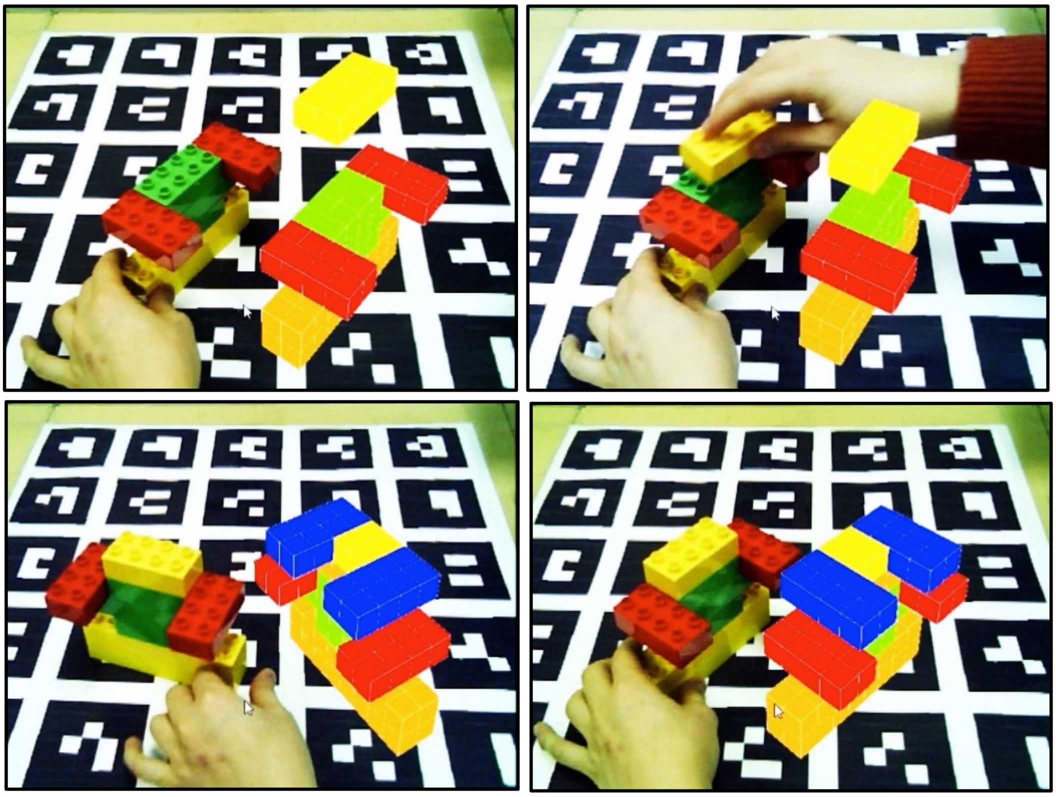


Figure 7.1: Side-by-side mode in our previous work. In each screen shot, the left object is real and the right object is virtual guidance with the next piece to attach.

However, the poses of virtual 3D guidance information are not normally updated in real-time to match to those of their real object counterparts. Thus the user needs to mentally rotate guidance information to the corresponding real object.

We take advantage of both object- and head (screen)-stabilized information display and propose hybrid object and screen stabilized visualization techniques as a solution for the problems above. We also conduct an evaluation between hybrid visualization modes proposed with the side-by-side mode, the best performing mode in our prior series of experiments. Our experimental results indicate that one of the two hybrid visualization modes yields better than the side-by-side mode in both performance and user preferences aspects.

7.2 Hybrid Visualization Modes Proposed

7.2.1 Design Concept

Considering the pros and cons of each visualization technique, in this study we propose hybrid visualization techniques by combining both object and screen

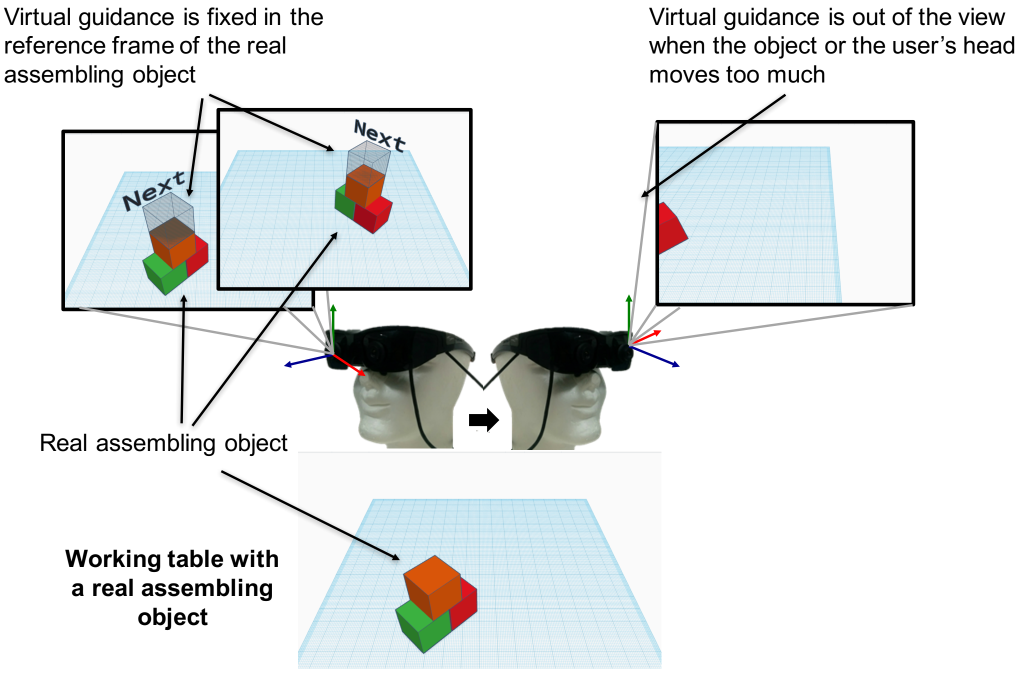

Virtual guidance is fixed in the reference frame of the real assembling object
Virtual guidance is out of the view when the object or the user's head moves too much
Next
Next
Real assembling object
Working table with a real assembling object

Figure 7.2: Object stabilized visualization

stabilized visualizations to take advantage of the two (Figure 7.4)..

As discussed in the previous section, object stabilized visualization can be advantageous for intuitive and coherent information display by referring to real objects. However, the guidance information is not available when the user looks away from the objects to which it refers. On the other hand, screen stabilized visualization is advantageous for immediate access to the guidance information regardless of the user's head orientation at the expense of the necessity of its mental rotation to match to the corresponding object.

To incorporate the advantages of both approaches, our idea is to fix the guidance information in the screen coordinates, and to update its orientation in real-time to match to that of the referring objects. In other words, orientation of the guidance information is object stabilized, but its position is screen stabilized. This way, we hypothesized that the guidance information would be always immediately accessible and understandable without the need for additional mental rotation.

7.2.2 Proposed Techniques

With the above design kept in mind, we propose two hybrid object and screen stabilized visualization modes (Figure 7.5). In the first mode (called hybrid fixed mode, Figure 7.5A), the virtual guidance information is displayed at a corner of the screen (indicated in red). Its center of gravity (COG) is always aligned with the center of the information area, while its 3D position in world

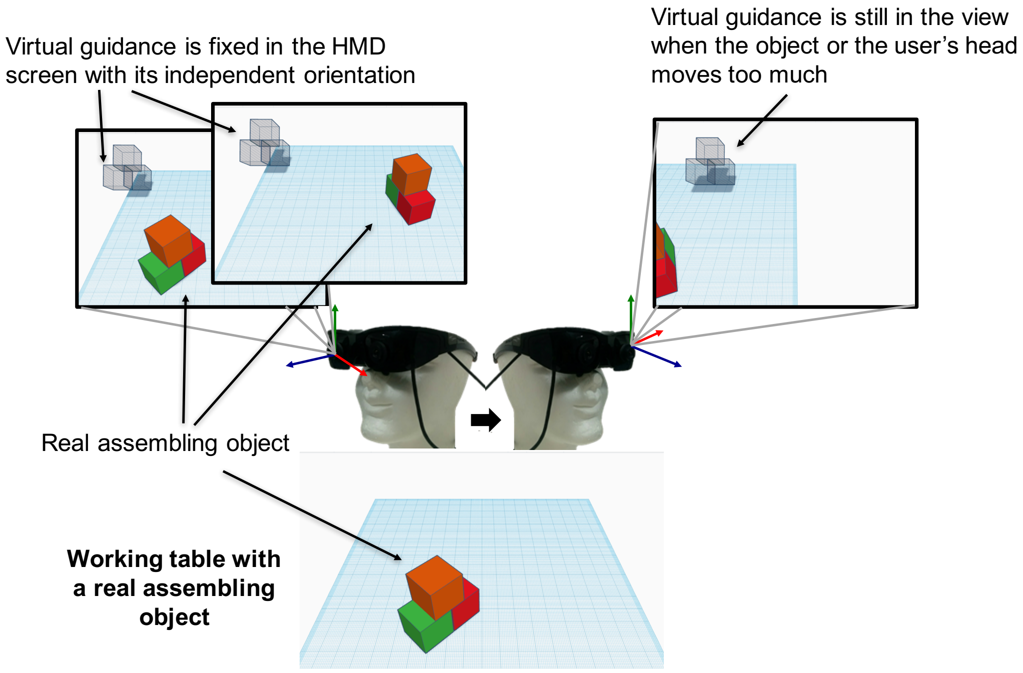

Virtual guidance is fixed in the HMD screen with its independent orientation
Virtual guidance is still in the view when the object or the user's head moves too much
Real assembling object
Working table with a real assembling object

Figure 7.3: Screen stabilized visualization

coordinates is determined by the intersection of the ray from the camera to the COG and the desktop surface (Figure 7.6). Thus the virtual guidance moves along the desktop surface depending on viewing direction. Its orientation is also updated in real-time to match to that of the corresponding real target object. Its size in space is shrunk to fit into the maximum screen area available. If there is no intersection between the ray and the desktop surface, its position and orientation remain the same.

In the second mode (called hybrid dynamic mode, Figure 7.5B), the system first detects the largest free rectangle area on the screen for every frame. It then automatically moves the virtual guidance information to that area with a smooth transition animation to minimize occlusion between the guidance information and the real objects. The 3D position of the virtual guidance is determined based on the intersection of the ray from the camera to the COG and the desktop surface.

7.2.3 Pilot Study to Determine Best Screen Position

To determine the best position and size of the guidance information area, we conducted a pilot study using the test bed system described in Section 4.1. We found that displaying guidance information on the top left area with a size of 50% by 50% of the entire screen yielded the best user preference among 12 participants, which would show the virtual guidance in approximately 60% in size of its real counterpart in our experiment. Four positions on the HMD

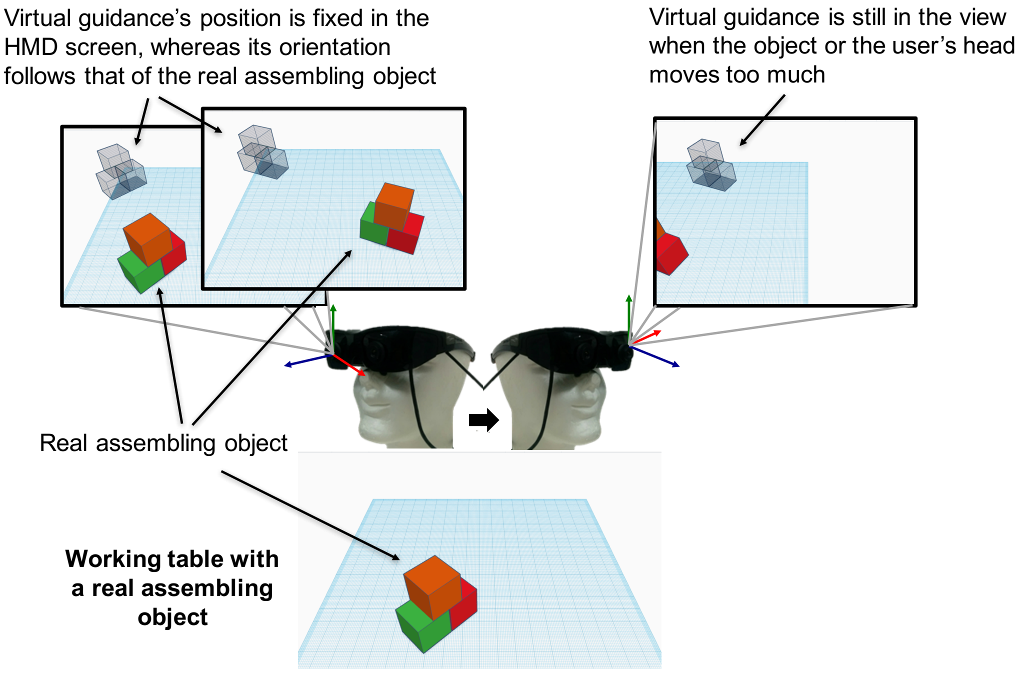

Virtual guidance's position is fixed in the HMD screen, whereas its orientation follows that of the real assembling object
Virtual guidance is still in the view when the object or the user's head moves too much
Real assembling object
Working table with a real assembling object

Figure 7.4: Hybrid object and screen stabilized visualization

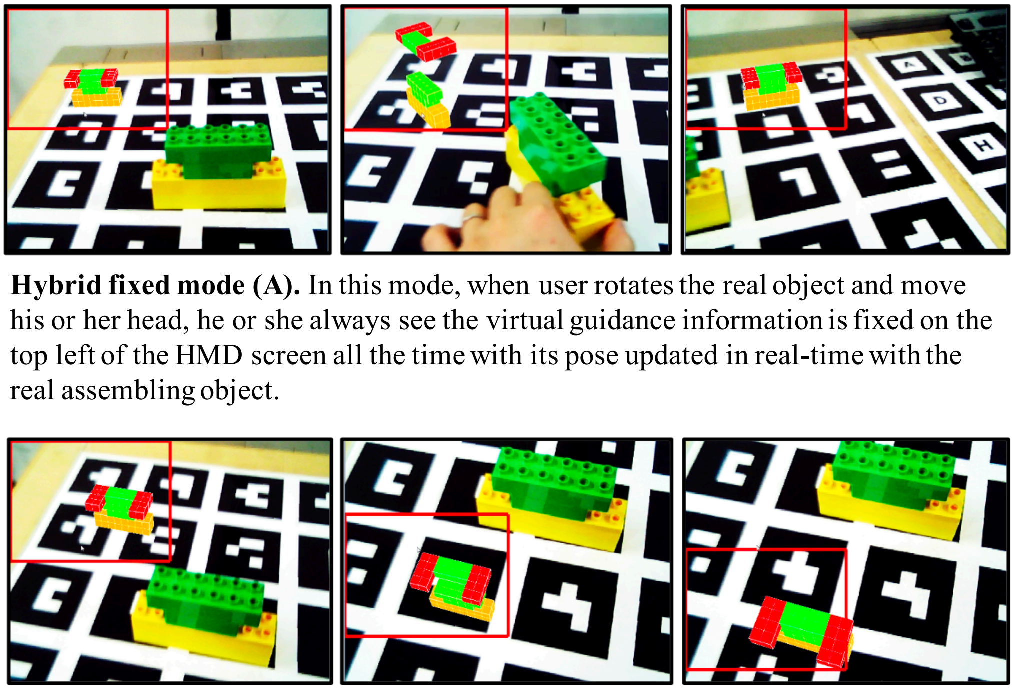
screen (top left (TL), top right (TR), bottom left (BL), and bottom right (BR)) and three levels of size of guidance information (40%, 50%, 60% of the real counterpart) were compared (Figure 7.7 and Figure 7.8). Other size options of guidance information (10%, 20%, 30% or 70%, 80%, 90%, etc.) were too small or too large when displaying on a narrow field-of-view HMD screen so we did not consider them in the scope of this pilot study. At each level of position, we respectively showed each level of size of the virtual guidance information and participants were asked to manipulate, translate, and rotate the model in the workspace. Then they were asked to fill out a questionnaire for feedback about the ease with which they could see the guidance information. The questionnaire consisted of a 7-point ordinal scale response, with 1 and 7 indicating the most negative and the most positive responses, respectively.

7.3 Evaluation

In the evaluation, we evaluate the two hybrid object- and screen-stabilized visualization modes, hybrid fixed mode and hybrid dynamic mode, in comparison to the side-by-side mode.

7.3.1 Hypotheses

In the hybrid fixed mode, users can see both the virtual guidance information and the real object being assembled in the HMD's view at the same time even



Hybrid fixed mode (A). In this mode, when user rotates the real object and move his or her head, he or she always see the virtual guidance information is fixed on the top left of the HMD screen all the time with its pose updated in real-time with the real assembling object.

Hybrid dynamic mode (B). In this mode, when the user moves his or her head down, the virtual guidance information is automatically moved to the largest free space recognized on the HMD screen to avoid occlusion between the guidance information and the real assembling object.

Figure 7.5: Proposed visualization modes.

if the real object is large. Automatic scaling of the virtual guidance helps to see the entire model or to confirm the detail on demand.

In hybrid dynamic mode, although the system automatically moves the guidance information to avoid occlusion with the real assembling object, the user may lose focus and feel stress when they have to chase the guidance information frequently. This may also increase completion time of assembly tasks. desiTherefore, we have made the following hypotheses:

- *H1: Hybrid fixed mode will achieve the best task completion time among the visualization modes considered.*

- *H2: Hybrid fixed mode will achieve the best user preference in aspects: ease of understanding, ease of seeing, stress levels, familiarity, satisfaction and usefulness.*

7.3.2 Experiment Design

We use four Duplo structure models in this evaluation, two with low complexity in structure and another two are high complexity. We used a one

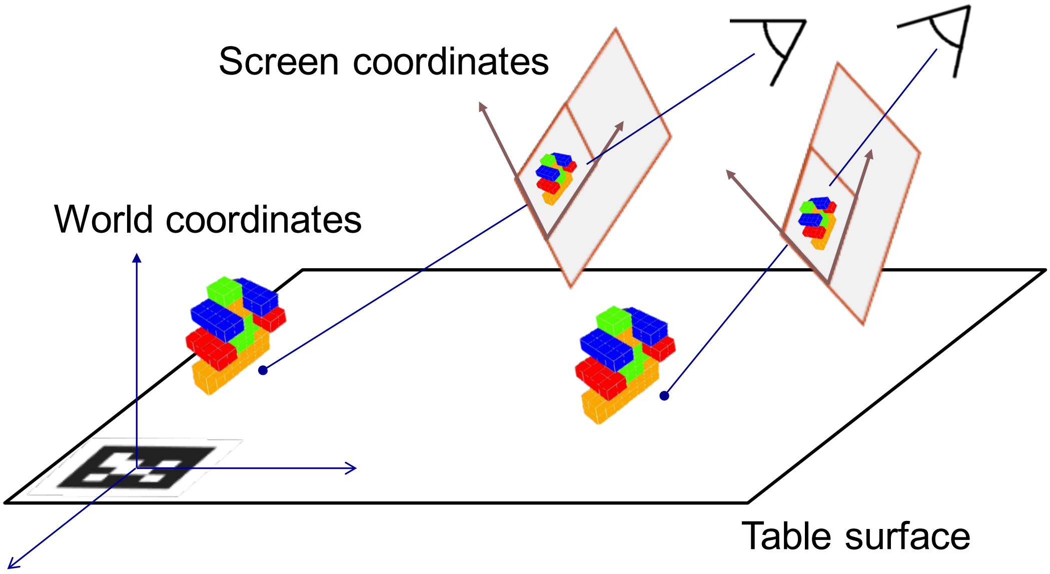


Figure 7.6: Intersection of the ray from the camera to the COG and the desktop surface.

way within-subjects experimental design, where independent variable was the visualization mode for presenting assembly instructions, and dependent variables were time taken to complete the tasks, and subjective scores on ease of understanding, ease of seeing, stress level, familiarity, satisfaction, and usefulness. The independent variable ranged over three levels: hybrid fixed mode, hybrid dynamic mode and side-by-side mode. Each participant was subjected to all three levels in a randomized order. For each level, the participants were asked to assemble four Duplo models in a randomized order (Figure 7.9).

7.3.2.1 Procedure

Before the experiment, each participant was given a tutorial on each of the levels. When a participant finishes an assembly task, he or she was asked to fill out a questionnaire shown in Table 7.1, allowing them to give feedback about their experience with each condition.

7.3.2.2 Metrics

- **Assembly task**: We use two Duplo structure models in this evaluation. Assembly of a structure model is considered as an assembly task.
- **Completion time**: Completion time of an assembly task is recorded by an observer when a participant starts an assembly task until he or she indicates that he or she has finished the assembly and the assembled model is ready for checking and evaluation. Completion time of an

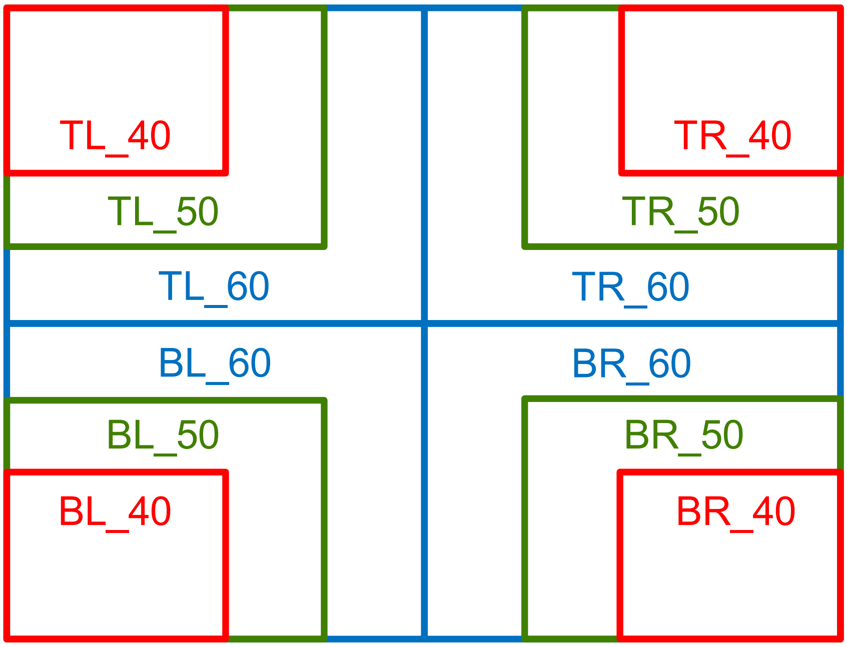


Figure 7.7: Compared position and size of the guidance information area in the pilot study.

assembly task is measured in second unit. We will explore the mean of completion time in each visualization mode.

- **Scaling user preference**: We also explore the user preference for each mode based on the questionnaire mentioned above. The questionnaire consisted of 7-point ordinal scale responses, with 1 indicating the most negative response and 7 indicating the most positive response.

7.3.2.3 Subjects

We recruited 24 people (12 males, 12 females) from many different departments of Osaka University for participation in this evaluation. The ages of participants were between 22 and 30 years. Only 5 participants had prior experience with AR.

7.3.3 Analysis on Quantitative Data

Figure 7.10 illustrates the mean task completion time for each level in the evaluation. A star indicates a significant level of $p < 0.05$. In average, hybrid fixed mode had the shortest completion time, while hybrid dynamic mode had the longest. We conducted a repeated measure ANOVA test and post-hoc pairwise t-tests and found differences between hybrid fixed mode and hybrid dynamic mode (t_{23} = -2.80, p = 0.010). We found a near marginally significant difference between hybrid fixed mode and Side-by-side mode (t_{23} =

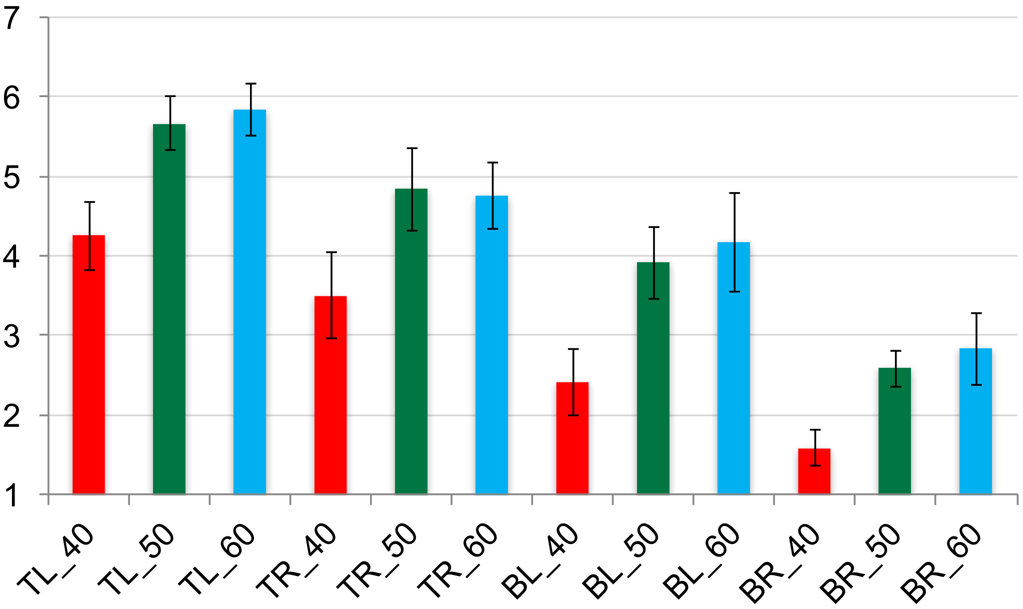


Figure 7.8: Conditions rated on ease with which participants could see them in the pilot study.

-1.70, $p = 0.10$) but no significant difference between Side-by-side mode and hybrid dynamic mode (t_{23} = -0.13, $p = 0.90$).

We also conducted a repeated measure ANOVA test and post-hoc pairwise t-tests on the task completion time data of the big models only (models 3 and 4 in Figure 7.9), and found significant differences between hybrid fixed mode and hybrid dynamic mode (t_{23} = - 2.67, $p = 0.014$). We found a marginally significant difference between hybrid fixed mode and Side-by-side mode (t_{23} = - 1.8794, $p = 0.07$) but no significant difference between Side-by-side mode and hybrid dynamic mode (t_{23} = 0.34, $p = 0.734$).

7.3.4 Analysis on Questionnaire Data

Figure 7.11 shows the participants' ratings in the questionnaire for each condition in each aspect; ease of understanding, ease of seeing, stress level, familiarity, satisfaction level and usefulness. Based on the ratings elicited from the participants, the hybrid fixed mode was the most preferred among three visualization modes in every qualitative aspect (Table 7.2). We used a non-parametric Friedman test to check for significant differences in each qualitative metric among the conditions, followed by Wilcoxon signed rank tests. One, two and three stars indicate significant differences of $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively.

Table 7.1: Questionnaire for evaluating the effectiveness of conditions.

No	Question	Response Type
1	Were the assembly instructions information and error notification difficult to understand?	7-point ordinal scale (1:Difficult to understand; 7:Easy to understand)
2	Were the assembly instructions information and error notification difficult to see?	7-point ordinal scale (1:Difficult to see; 7:Easy to see)
3	Did you feel stress when using this assembly instructions media?	7-point ordinal scale (1:Feel very stressed; 7:Do not feel the stress)
4	Did you feel difficult to become familiar with the assembly instructions media?	7-point ordinal scale (1:Difficult to become familiar; 7:Easy to become familiar)
5	Did you feel satisfied with the assembly instructions media after using it?	7-point ordinal scale (1:Not satisfied at all; 7:Very satisfied)
6	Did you feel the assembly instructions media useful for the assembly tasks?	7-point ordinal scale (1:Not useful at all; 7:Very useful)

Table 7.2: Significant results from analysis of questionnaire data.

	Friedman	FIXED > SIDE-BY-SIDE	FIXED > DYNAMIC	SIDE-BY-SIDE > DYNAMIC
Ease of understanding	$X^2 = 9.24$, $p < 0.01$	$z = 2.42$, $p < 0.05$	$z = 3.13$, $p < 0.01$	$z = 0.42$, $p < 0.68$
Ease of seeing	$X^2 = 16.87$, $p < 0.001$	$z = 2.84$, $p < 0.005$	$z = 3.82$, $p < 0.0005$	$z = 0.93$, $p < 0.35$
Stress level	$X^2 = 9.70$, $p < 0.01$	$z = 2.24$, $p < 0.05$	$z = 3.30$, $p < 0.001$	$z = 0.56$, $p < 0.58$
Familiarity	$X^2 = 11.04$, $p < 0.01$	$z = 2.10$, $p < 0.05$	$z = 3.62$, $p < 0.0005$	$z = 0.50$, $p < 0.62$
Satisfaction	$X^2 = 12.70$, $p < 0.01$	$z = 2.80$, $p < 0.01$	$z = 3.26$, $p < 0.005$	$z = 0.40$, $p < 0.69$
Usefulness	$X^2 = 16.34$, $p < 0.001$	$z = 2.85$, $p < 0.005$	$z = 3.90$, $p < 0.0001$	$z = 0.90$, $p < 0.37$

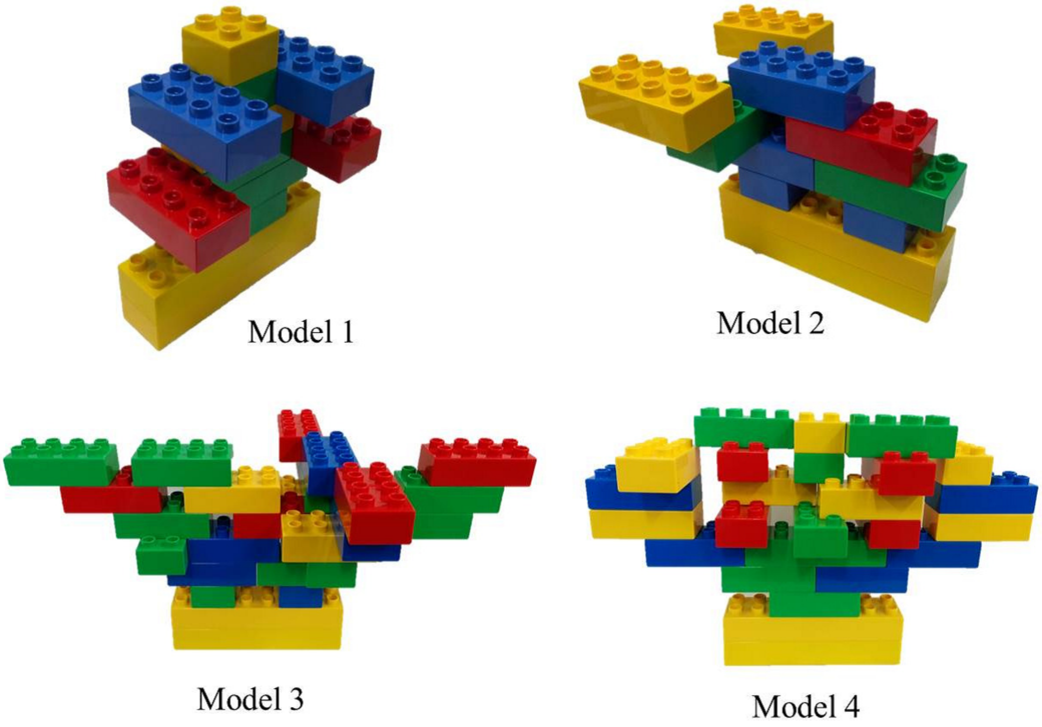


Figure 7.9: Models for the main evaluation.

7.4 Findings and Discussion

In the first hypothesis H1, we hypothesized that hybrid fixed mode would achieve the best task completion time among the visualization modes considered. The results of the experiment supported this hypothesis.

In hybrid fixed mode, users can see both virtual guidance information whose pose is updated in real-time, and the real counterpart objects being assembled in the same viewport on the HMD screen even when the real objects being assembled have a big size. The guidance information also automatically changes the size on demand when the user moves his or her head closer to or further away the work table. This helps the user reduce head and eye movements, cost of attention switching and spatial perception.

However, this mode still has occlusion problems between virtual guidance information and the real counterpart objects being assembled, the visualization of the guidance information becomes harder when models become bigger and more complex due to the limited screen space for the guidance information as well as narrow field of view and low resolution of the HMD screen. This might increase time to confirm parts' position as well as their relationship at each assembly step and thus overall completion time may also increase. The Hybrid fixed time was not clearly faster than the Side-by-side presumably due to this issue.

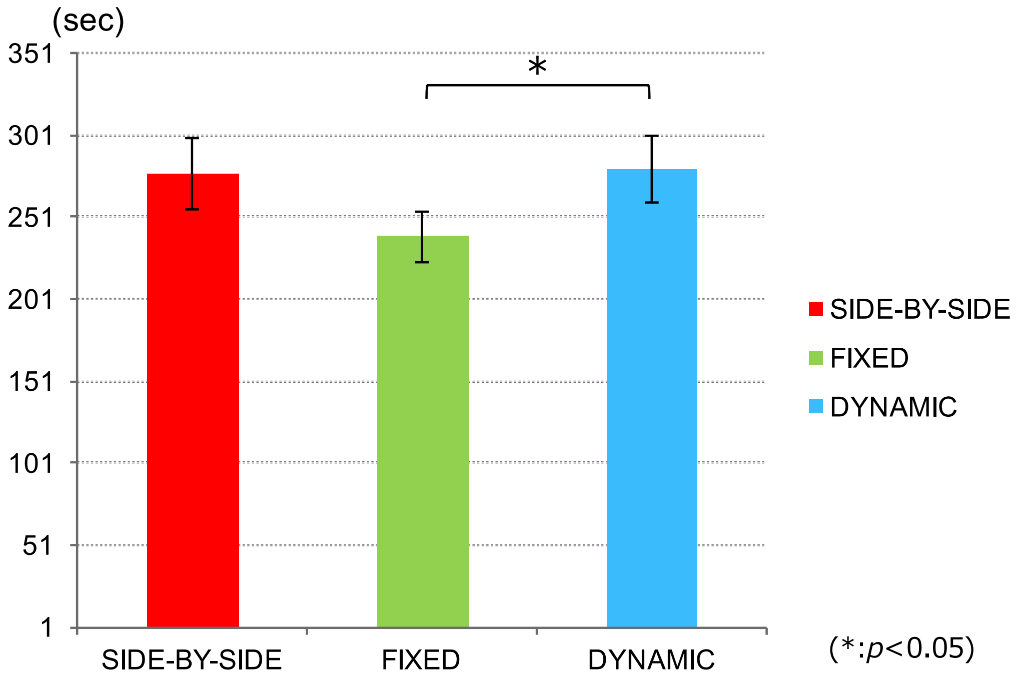


Figure 7.10: The mean task completion time (sec) of each level in the main evaluation. Error bars indicate 95% confidence intervals.

In hybrid dynamic mode, the users reported that although the system automatically detects and moves the guidance information to the position with the least occlusion between the guidance information and the real counterpart objects, the users were hard to control movement of the guidance information to a desirable position to refer. They had to chase the movement of the guidance information continuously making them lost of focus. They had to spend more time to find and confirm guidance information during assembly process and thus this mode had the worst task completion time among the visualization modes considered.

In the second hypothesis H2, we stated that hybrid fixed mode would achieve the best user preference in aspects; ease of understanding, ease of seeing, stress level, familiarity, satisfaction level and usefulness. The results of the experiment supported this hypothesis.

In hybrid fixed mode we believed that the ability to display virtual guidance information in a suitable size and rotation at a fixed position at a screen corner would help the users see and understand the relationship between it and the real counterpart objects with reduced perceived workload during assembly tasks. This mode can also make them feel more satisfaction and usefulness than other visualization modes in the experiments. The subjective data showed that this mode had much better user preference when compared to other visualization modes and the differences are statistically significant in

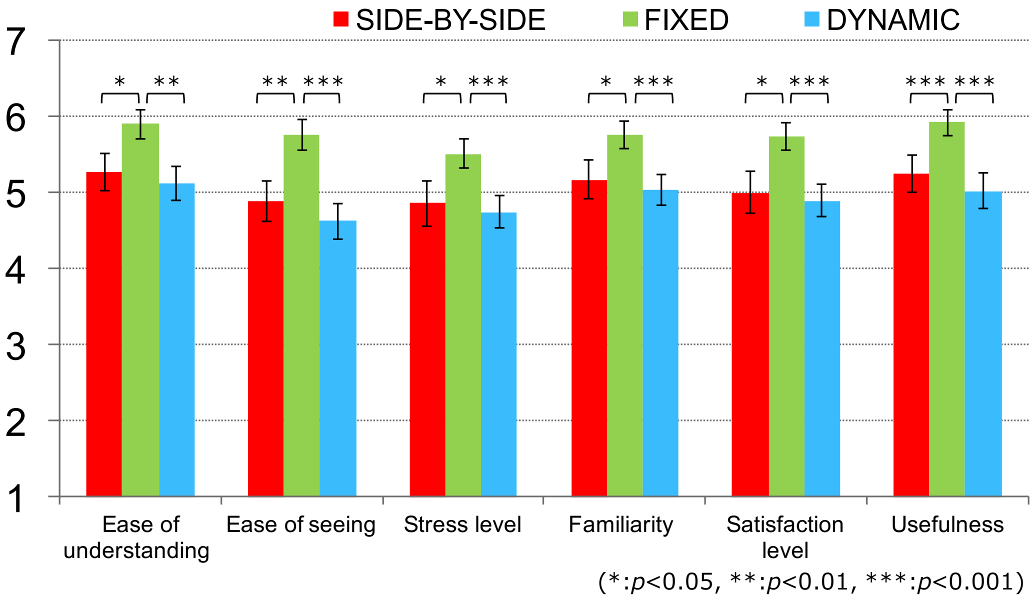


Figure 7.11: Subjective ratings in the questionnaire in the main evaluation.

all aspects.

7.5 Conclusion

In this chapter, we proposed and evaluated the effectiveness of hybrid object- and screen-stabilized visualization modes for AR-based context-aware assembly support systems. Our experimental results indicate that one of our proposed visualization modes, the hybrid fixed mode, is the most preferred in many aspects, such as ease of understanding, stress level, and usefulness, among all visualization modes tested. We also found that the hybrid fixed mode in average yielded the fastest task completion time with a marginally significant difference.

Chapter 8
Conclusion

8.1 Summary of Findings

Our goal of this study is to explore the best representation for guidance information of a context-aware assembly support system using augmented reality which supports the best performance in assembly tasks. In this dissertation we have explored a variety of guidance information representations (visualization methods) and we found some design concepts that support the goal.

In usual conditions that are mostly used in assembly in practice, such as using head mounted display (HMD) devices with a narrow field of view, a marker-based environment, 3-DOF tracking on the table with moderate registration accuracy, models with size that fits within the field of view of the HMD at a normal reaching distance, we found the side-by-side mode - a virtual clone of the real object that is assembled, whose structure and pose are updated in real-time to match those of the real object, has the best performance and user preferences. The side-by-side mode with a colorful, solid virtual representation displayed beside the real models not only avoids effects of misalignment but also makes it easier for users to see and determine the next guided blocks position. Experimental results proved that this mode can help users shorten the completion time and lower stress to finish the models.

In situations that the side-by-side visualization is not always available such as when size of models being assembled is too big, virtual representation rendered adjacent to the real models in the side-by-side mode cannot be displayed due to limited, small size of HMD screens, we found that the partial-wireframe overlay method was the best among three overlay visualization modes proposed. Participants reported that the partial-wireframe overlay mode makes it easier for them to see and determine the position of guided blocks on the real models while still helping them to figure out the spatial correspondence between the guided blocks and the real models being assembled.

We also consider a situation that has a mix size of models with context-aware error detection on/off. We found, somewhat surprisingly, that the side-by-side mode was preferable in every aspect to displaying such information directly overlaying the physical model (the partial-wireframe mode). Participants reported that because of misalignment and the potential to mistake some parts of the real model for parts of the virtual representation of the same color, they spent more time trying to determine the position of next blocks as they did in the side-by-side mode. The partial-wireframe mode, under moderate registration accuracy, reveals weak points such as being highly sensitive to misalignment, latency, or conflicting depth cues.

We did another approach by taking advantage of both object and head (screen) stabilized information display and proposed hybrid object and screen stabilized visualization techniques as a solution for situations that the side-by-side mode is not available. We found that one of the two hybrid world and screen stabilized visualization modes proposed that shows virtual target status of real assembling objects at a fixed position on the HMD screen with real-time pose updating has the best performance and user preferences under the context considered in this study.

8.2 Future Directions

Although the study focused on simple settings and usual working conditions that are mostly used in assembly in practice, we would consider the cases of using better HMDs with higher resolution, larger field of view, a system with more complex settings in combination with other external tracking support sensors or devices for getting higher registration accuracy. This could affect the findings between the traditional overlay mode and the side-by-side mode in this study, possibly make it harder to find a significant difference between the two modes.

Under limitation of using only one first generation of Kinect with a small field of view (FOV) in the current system, only small models can be considered and users need to rotate models on the table to let the Kinect to recognize the models for tracking and error detection purposes. We would like to use a new RGBD with a HMD with larger FOV to consider experimental models with bigger size and more complex structure, using multiple Kinect at the same time from different point of view to accelerate tracking and error detection process, thus decrease or get rid of rotation time of assembling models required in current setting, naturally decrease the total time of task completion. When more complex experimental models are considered, AR context-aware guidance information possibly becomes more essential and helpful for users during assembly processes. This would make a significant difference on effectiveness of the AR system in comparison with traditional paper manual (under the current conditions in this study, we did not find a significant difference between them in task completion time aspects).

With current algorithms, tracking objects are limited to block shapes and motion tracking is limited to 3DOF (two dimensions for translation on the table and one dimension for rotation around the table surface normal. In the future work, we would like to improve the tracking algorithms to allow tracking objects with variant shapes and in 6DOF in the working environments. In combination with a high resolution optical see-through HMDs with no cables attached would help the system to be more widely used in object assembly in practice.

Bibliography

[1] Artesas, http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=artesas.

[2] Arvika, http://www.arvika.de/.

[3] Cognito, http://www.ict-cognito.org/index.html.

[4] Create tool, http://www.informationinplace.com/.

[5] Dart toolkit, http://www.gvu.gatech.edu/dart/.

[6] Inition, http://www.inition.co.uk/.

[7] Kinect, http://msdn.microsoft.com/en-us/library/jj131033.aspx.

[8] Mars authoring tool, http://www.cs.columbia.edu/graphics/projects/mars/.

[9] Thinglab, http://www.thinglab.co.uk/.

[10] Vuzix, http://www.vuzix.com.

[11] *Augmented reality: an application of heads-up display technology to manual manufacturing processes*, volume ii, August 2002.

[12] M. Andersen, R. Andersen, C. Larsen, T. B. Moeslund, and O. Madsen. *Interactive Assembly Guide Using Augmented Reality*, pages 999–1008. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[13] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[14] K. M. Baird and Woodrow Barfield. Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality*, (4):250–259.

[15] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphett. A wearable spatial conferencing space. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 76–83, Oct 1998.

[16] Oliver Bimber and Ramesh Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds.* A. K. Peters, Ltd., Natick, MA, USA, 2005.

[17] Gabriele Bleser, Yulian Pastarmov, and Didier Stricker. Real-time 3d camera tracking for industrial augmented reality applications. *Journal of WSCG*, pages 47–54, 2005.

[18] Gabriele Bleser, Harald Wuest, and Didier Stricker. Online camera pose estimation in partially known and dynamic scenes. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 56–65, Washington, DC, USA, 2006. IEEE Computer Society.

[19] Volkert Buchmann, Trond Nilsen, and Mark Billinghurst. Interaction with partially transparent hands and objects. In *Proceedings of the Sixth Australasian Conference on User Interface - Volume 40*, AUIC '05, pages 17–20, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[20] Jiajian Chen and Blair MacIntyre. Uncertainty boundaries for complex objects in augmented reality. In *Virtual Reality, 2008, VR 2008, Proceedings. International Conference on*, pages 247–248. IEEE, 2008.

[21] Youngkwan Cho, Jongweon Lee, and Ulrich Neumann. A multi-ring fiducial system and an intensity-invariant detection method for scalable augmented reality. In *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes: placing artificial objects in real scenes*, IWAR '98, pages 147–165, Natick, MA, USA, 1999. A. K. Peters, Ltd.

[22] E. M. Coelho, B. MacIntyre, and S. Julier. Supporting interaction in augmented reality in the presence of uncertain spatial knowledge. In *User Interface Software and Technology, 2005. UIST 2005. Proceedings. International Symposium on*, pages 111–114. ACM, 2005.

[23] Andrew I. Comport, Eric Marchand, Muriel Pressigout, and Francois Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, July 2006.

[24] Chris Furmanski, Ronald Azuma, and Michael Daily. Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 215–320. IEEE, 2002.

[25] R. Haines and T. Fischer, E.and Price. Head-up transition behavior of pilots with and without head-up display in simulated low-visibility approaches. Technical Report Technical Report: NASA Ames Research Center,Moffett Field., 1980.

[26] S.J. Henderson and Steven K. Feiner. Augmented reality in the psychomotor phase of a procedural task. In *Mixed and Augmented Reality*

(ISMAR), 2011 10th IEEE International Symposium on, pages 191–200, 2011.

[27] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[28] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[29] Berthold K. P. Horn and John G. Harris. Rigid body motion from range image sequences. *CVGIP: Image Underst.*, 53(1):1–13, January 1991.

[30] Hong Hua, Chunyu Gao, Leonard D. Brown, Narendra Ahuja, and Jannick P. Rolland. Using a head-mounted projective display in interactive augmented environments. In *IN PROCEEDINGS OF IEEE AND ACM INTERNATIONAL SYMPOSIUM ON AUGMENTED REALITY 2001 , (ACM*, pages 217–223. Press, 2001.

[31] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *ACM SIGGRAPH 2011 Talks*, SIGGRAPH '11, pages 23:1–23:1, New York, NY, USA, 2011. ACM.

[32] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, IWAR '99, pages 85–, Washington, DC, USA, 1999. IEEE Computer Society.

[33] Bui Khuong, Kiyoshi Kiyokawa, Andrew Miller, Joseph LaViola Jr., Tomohiro Mashita, and Haruo Takemura. The effectiveness of an ar-based context-aware assembly support system in object assembly. In *Proc. IEEE Virtual Reality 2014*, Minneapolis, 2014.

[34] Bui Minh Khuong, Kiyoshi Kiyokawa, Andrew Miller, Joseph LaViola Jr., Tomohiro Mashita, and Haruo Takemura. The effectiveness of an ar-based context-aware assembly support system in object assembly. In *Proc. IEEE Virtual Reality 2014*, Minneapolis, 2014.

[35] Georg Klein and Tom Drummond. Robust visual tracking for non-instrumented augmented reality. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 113–, Washington, DC, USA, 2003. IEEE Computer Society.

[36] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, July 2000.

[37] Mark A. Livingston, J. Edward Swan II, Joseph L. Gabbard, Tobias Höllerer, Deborah Hix, Simon Julier, Yohan Baillot, and Dennis Brown. Resolving multiple occluded layers in augmented reality. In *Mixed and Augmented Reality, 2003. ISMAR 2003. Proceedings. International Symposium on*, pages 56–65. IEEE, 2003.

[38] Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, and Joseph J. LaViola Jr. Interactive 3d model acquisition and tracking of building block structures. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):651–659, April 2012.

[39] Leonid Naimark and Eric Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 27–, Washington, DC, USA, 2002. IEEE Computer Society.

[40] U. Neumann and A. Majoros. Cognitive, Performance, and Systems Issues for Augmented Reality Applications in Manufacturing and Maintenance. In *Proceedings of the Virtual Reality Annual International Symposium*, VRAIS '98, Washington, DC, USA, 1998. IEEE Computer Society.

[41] Ulrich Neumann and Anthony Majoros. Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 4–11. IEEE.

[42] Wayne Piekarski and Bruce H. Thomas. Augmented reality working planes: A foundation for action and construction at a distance. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 162–171, Washington, DC, USA, 2004. IEEE Computer Society.

[43] M. Pressigout and E. Marchand. Real-time 3d model-based tracking: combining edge and texture information. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2726 –2731, may 2006.

[44] Muriel Pressigout and Eric Marchand. Hybrid tracking algorithms for planar and non-planar structures subject to illumination changes. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 52–55, Washington, DC, USA, 2006. IEEE Computer Society.

[45] Dirk Reiners, Didier Stricker, Gudrun Klinker, and Stefan Müller. Augmented reality for construction tasks: Doorlock assembly. In *Proceedings*

of the IEEE and ACM IWAR?98 (1. International workshop on augmented reality, pages 31–46. AK Peters, 1998.

[46] Cindy M Robertson, Blair MacIntyre, and Bruce N Walker. An evaluation of graphical context as a means for ameliorating the effects of registration error. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):179–192, 2009.

[47] Jannick P. Rolland, Yohan Baillot, and Alexei A. Goon. A survey of tracking technology for virtual environments, 2001.

[48] Juha Sääski, Tapio Salonen, Mika Hakkarainen, Sanni Siltanen, Charles Woodward, and Juhani Lempiäinen. Integration of design and assembly using augmented reality. In *Micro-Assembly Technologies and Applications, IFIP TC5 WG5.5 Fourth International Precision Assembly Seminar (IPAS'2008), Chamonix, France, February 10-13, 2008*, pages 395–404, 2008.

[49] Juha Sääski, Tapio Salonen, Mika Hakkarainen, Sanni Siltanen, Charles Woodward, and Juhani Lempiäinen. Integration of design and assembly using augmented reality. In Svetan Ratchev and Sandra Koelemeijer, editors, *Micro-Assembly Technologies and Applications*, volume 260 of *IFIP ? International Federation for Information Processing*, pages 395–404. Springer US, 2008.

[50] B. Schwald, J. Figue, E. Chauvineau, and et al. Starmate: Using augmented reality technology for computer guided maintenance of complex mechanical elements. *E-work and ECommerce*, 1(196-202), 1992.

[51] B. Schwald, J. Figue, E. Chauvineau, and et al. Starmate: Using augmented reality technology for computer guided maintenance of complex mechanical elements. *E-work and ECommerce*, 1(196-202), 1992.

[52] Didier Stricker, Gundrun Klinker, and Dirk Reiners. A fast and robust line-based optical tracker for augmented reality applications. In *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes: placing artificial objects in real scenes*, IWAR '98, pages 129–145, Natick, MA, USA, 1999. A. K. Peters, Ltd.

[53] Ivan E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

[54] Anna Syberfeldt, Oscar Danielsson, Magnus Holm, and Lihui Wang. Visual assembling guidance using augmented reality. *Procedia Manufacturing*, 1(Supplement C):98 – 109, 2015. 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States.

[55] X. Wang, S. K. Ong, and A. Y. C. Nee. A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing*, 4(1):1–22, Mar 2016.

[56] Harald Wuest, Florent Vial, and Didier Stricker. Adaptive line tracking with multiple hypotheses for augmented reality. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 62–69, Washington, DC, USA, 2005. IEEE Computer Society.

[57] Jürgen Zauner, Michael Haller, Alexander Brandl, and Werner Hartmann. Authoring of a mixed reality furniture assembly instructor. In *Proceedings of the SIGGRAPH 2003 Conference on Sketches & Applications: in conjunction with the 30th annual conference on Computer graphics and interactive techniques, 2003, San Diego, California, USA, July 27-31*, 2003.

[58] J. Zhang, S.K. Ong, and A.Y.C. Nee. Rfid-assisted assembly guidance system in an augmented reality environment. *International Journal of Production Research*, 49(13):3919–3938, 2011.

[59] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 193–202, Washington, DC, USA, 2008. IEEE Computer Society.

[60] Zhiwei Zhu, Vlad Branzoi, Michael Wolverton, Glen Murray, Nicholas Vitovitch, Louise Yarnall, Girish Acharya, Supun Samarasekera, and Rakesh Kumar. Ar-mentor: Augmented reality based mentoring system. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2014, Munich, Germany, September 10-12, 2014*, pages 17–22, 2014.