



Attribution–NonCommercial–NoDerivs 2.0 KOREA

You are free to :

- **Share** — copy and redistribute the material in any medium or format

Under the following terms :



Attribution — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



NonCommercial — You may not use the material for [commercial purposes](#).



NoDerivs — If you [remix, transform, or build upon](#) the material, you may not distribute the modified material.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

This is a human-readable summary of (and not a substitute for) the [license](#).

[Disclaimer](#) 

공학석사 학위논문

Utilization of Depth Camera and Augmented Reality for Efficient 3D Box Packing and Loading Operation

**Depth Camera와 증강현실(Augmented Reality)를
이용한 3D박스 패킹 및 적재 작업 시스템
개발에 관한 연구**

2017년 8월

서울대학교 대학원

산업공학 전공

Nasution Ninan Kara Gicha

Utilization of Depth Camera and Augmented Reality for Efficient 3D Box Packing and Loading Operation

지도교수 박 진 우

이 논문을 공학석사 학위논문으로 제출함
2017년 6월

서울대학교 대학원
산업공학과 전공
Nasution Ninan Kara Gicha

Nasution Ninan Kara Gicha의 석사 학위논문을
인준함
2017년 6월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

Abstract

Utilization of Depth Camera and Augmented Reality for Efficient 3D Box Packing and Loading Operation

Nasution Ninan Kara Gicha

Department of Industrial Engineering

The Graduate School

Seoul National University

Packing and loading operations are very frequent and important activities in logistic services. Not only that, the data input process is also another challenge to overcome for efficient loading. Such input data is used in the planning of packing and loading operation when several objects are packaged into a larger box. Currently, companies utilize certain software package to generate the solution. However, they consumed a lot of time in preparing and inserting the data. Furthermore, even if the software has generated the optimized solution, the delivery and representation of the solution may cause another difficulty for

workers at the loading field to understand the paper-based solution blueprint.

This research proposes a new approach to solve both data collection and solution representation problem occurred in the current situation. Real-time approach has been proposed by utilizing depth camera and Augmented Reality (AR) technology in the system. The new approach leverages a wide possibility in developing the system into smart agent that is able to pack and load the boxes automatically into the container.

Keywords: 3D Box Packing and Loading Operation, Depth Camera, Augmented Reality, Integrated System, Cyber Physical System (CPS), Industry Application, Logistics

Student Number: 2015 – 23295

Table of Contents

1. Introduction	1
1.1. Current Situation	2
1.1.1. Indonesia Situation	2
1.1.2. Other Situation	3
1.1.3. DHL Germany Freight Loading Operation.....	7
1.2. Motivation	9
1.3. Problem Description.....	9
1.4. Research Objective and Scope	12
1.5. Structure of the Thesis.....	13
2. Related Work.....	14
2.1. Three-dimensional Box Loading Problem	14
2.2. Automatic Measurement of Dimension	17
2.3. Augmented Reality Visualization.....	19
2.4. End-to-end System Integration.....	22
3. System Design.....	23
3.1. Design of Operations.....	23
3.1.1. User Models	23
3.1.2. User Scenario	24
3.2. System Design Framework	26
3.2.1. Gap Analysis	26
3.2.2. System Design	26
3.3. Integration Model Architecture	28

3.4.	System Flowchart	30
3.5.	Database Design	33
3.6.	Output Module: AR Visualization	34
3.6.1.	Packing Algorithm	34
3.6.2.	AR Visualization Process	37
3.6.3.	Interaction Features of AR	39
3.7.	Input Module: Data Collection	42
4.	Experiment	45
4.1.	Experiment Setup	45
4.2.	Experiment Result	47
4.2.1.	Input Data	47
4.2.2.	Visualization Output	51
5.	Conclusion and Future Development	57
5.1.	Conclusion	57
5.2.	Future Development	60
5.2.1.	Data Collection Improvement	60
5.2.2.	Packing Algorithm Improvement	60
5.2.3.	AR Visualization Improvement	61
APPENDIX A MEASUREMENT		63
A.1	Depth Camera Specification	63
A.2	Measure Distance	64
A.3	Database Connector	64
APPENDIX B PACKING		66
A.1	Packing Algorithm Flowchart	66

A.2	Packing Pseudo Code	66
A.3	Marshalling Code	67
APPENDIX C VISUALIZATION		69
C.1	Connection PHP Script.....	69
C.2	Data Conversion (DBHandler.cs).....	70
C.3	Box Rendering (ManageBox.cs)	72
C.4	Manage AR Object Behaviour (ARHandler.cs)	74
C.5	Interaction Code	75
C.5.1	Translate.....	75
C.5.2	Scale.....	75
C.5.3	Rotate	76
C.5.4	Select.....	77
Bibliography.....		78
초록		82

List of Tables

[Table 3-1] Database Structure for Box Data	33
[Table 3-2] AR Application Structure Description	38
[Table 3-3] Interaction Description	39
[Table 4-1] Dimension Data and BoxID Data	45
[Table 4-2] Prototype Development Tools	47
[Table 4-3] Measurement Result	50
[Table 4-4] Container's size for experiment	52
[Table 4-5] Visualization AR of Packing Solution	53
[Table 4-6] Size and Coordinate Data Comparison.....	56
[Table A-1] Real Sense and Kinect Specification	63

List of Figures

[Figure 1-1] Cargonesia System Workflow	3
[Figure 1-2] Input Box Information User Interface	4
[Figure 1-3] Side-view of the container	5
[Figure 1-4] Preview of the whole container	6
[Figure 1-5] DHL Freight Loading AR Application	7
[Figure 1-6] Container Loading Process	10
[Figure 2-1] Distance Measurement Classification	17
[Figure 2-2] Core Components of Augmented Reality	21
[Figure 3-1] As-Is System Illustration	24
[Figure 3-2] Proposed System Overview	27
[Figure 3-3] Integration Model	29
[Figure 3-4] User Scenario for Implementation	31
[Figure 3-5] Flowchart of proposed system	32
[Figure 3-6] Structure of AR Application	37
[Figure 3-7] Depth Camera Measurement Algorithm	42
[Figure 3-8] Triangulation Basic Principle	43
[Figure 4-1] Boxes for Experiment	46
[Figure 4-2] Depth Camera attached (Left) Experiment Setup (Right)	46
[Figure 4-3] Box Measurement Interface Application	48
[Figure 4-4] Flowchart of using the interface application	48
[Figure 4-5] MySQL Database Data	49
[Figure 4-6] Image Marker for Experiment	52
[Figure 4-7] Box 2 Object Data	55
[Figure 4-8] Container 1's packing result	55

1. Introduction

Packing and loading operation is one of the important activities in logistic business field. Planning the packing and loading become important when the company is trying to minimize the operation cost. For instance, the application of packing and loading operation can be found in domestic parcel delivery service, container or freight loading, ship building and yard management, transport logistic, designing packaging for wrapping items, storing item in warehouse and other logistic applications. However, not every packing and loading operation needs optimization planning because of the nature of each problem and condition.

In South Korea, box parcel delivery business receives 80,000 – 100,000 boxes every day [EBS Documentary, 2013]. A huge number to handle within one day. Mistake and delay are not allowed in order to satisfy the demands. When the boxes were delivered from post office branch and arrived at the distribution center, the boxes are directly put on the conveyor to be distributed based on the region. Each region has its own truck container to deliver the boxes to customer. However, since the business is running very fast each day, space utilization planning inside the distribution center inventory storage and transport container is not important, because there is no time for loading the boxes according to the plan. In this kind of application, the optimization can be applied in the other operation, such as in the delivery scheduling or routing schedule.

In the other hand, in overseas transport services, the space utilization is crucial because in long distance shipping, the minimum number of used container to ship all the boxes to other places can contribute in decreasing the operation cost.

Planning the cargo loading inside the container is necessary in order to find efficient loading when delivering the cargo to other destination. This kind of business pushes the company what action to decide in order to avoid the underperformance solution. This research focuses on the application of packing and loading operation, where space utilization is important part of the operation. For instance, container loading for long distance shipping.

1.1. Current Situation

This section introduces various current situation related to the application of container loading operation. Understanding the current situation can help defining the limitation of the current operation and find a gap to define the problem.

1.1.1. Indonesia Situation

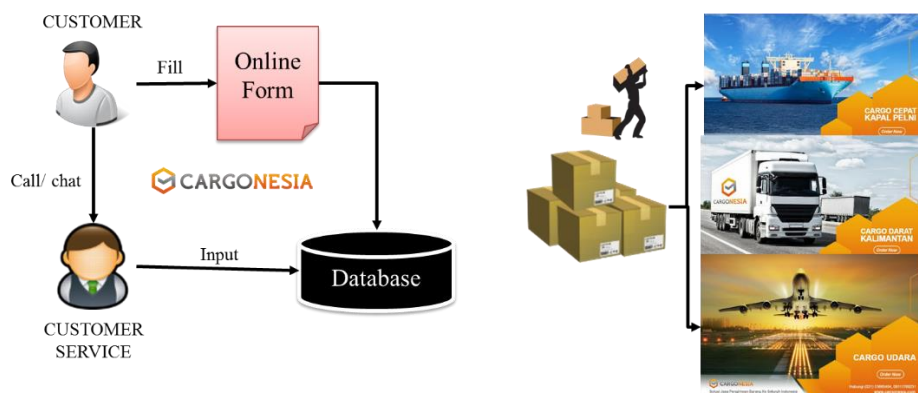
Cargonesia¹ is an Indonesia-based company that provides cargo-shipping services to any province in Indonesia through sea, land and air. Indonesia is the world's largest island country with more than thirteen thousand islands [CIA, 2017]. Shipping service between islands is a big business within the country.

[Figure 1-1] is the workflow of Cargonesia system when customer orders the service. Customer can order using online system or contacting the customer service directly. If customer orders online, He needs to fill the necessary information regarding the weight, size, box content and other information

¹ <https://cargonesia.com/>

through online form. Otherwise, if He calls or sends message through messenger to the customer service, the customer service will fill the information in the company system on behalf of the customer.

When the picking up date is decided, customer can leave the items by himself or Cargonesia can pick them up. Cargonesia does not use optimization software to plan the loading inside the container. They just load the box according to destination and experience. Moreover, the information that stored inside the database is only for historical record. It seems that there is no further utilization of the data in another operation.



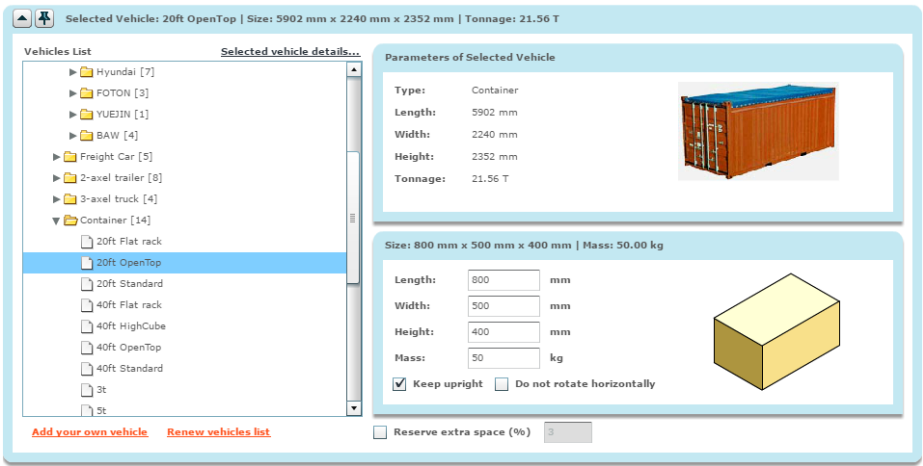
[Figure 1-1] Cargonesia System Workflow

1.1.2. Other Situation

Agility² is logistic service company from Dubai. It has many branches that spread in the world. This company offers freight-shipping service as one of their

² <http://www.agility.com/EN/Pages/Default.aspx>

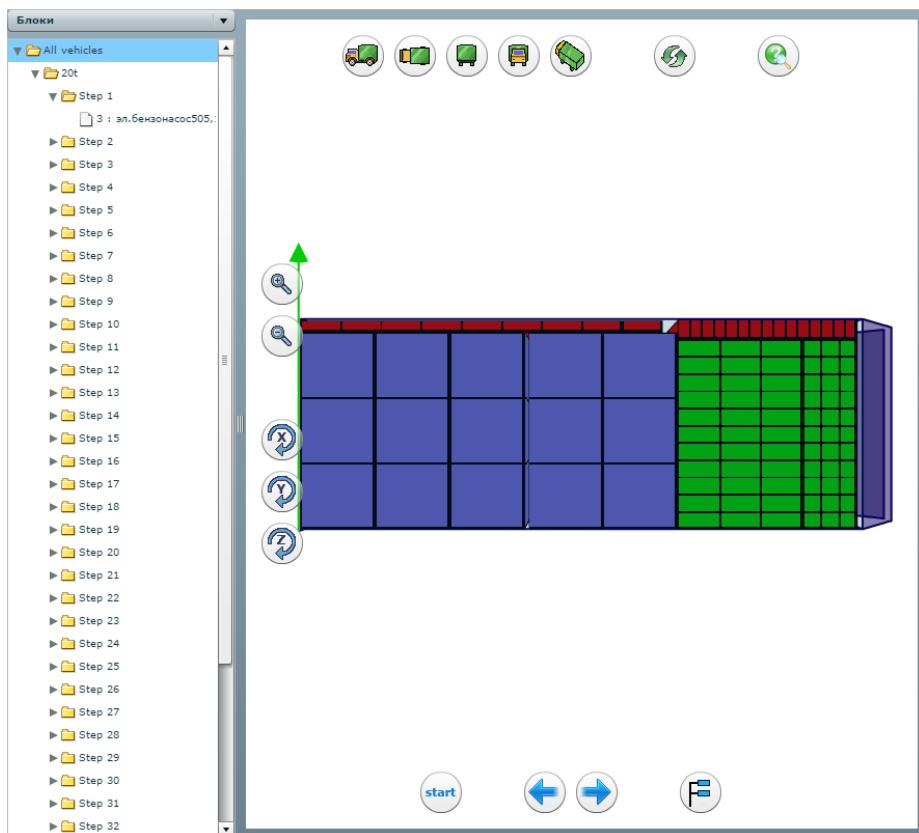
main business. In the loading operation, they use Packer3D³ software to plan an efficient loading. The order process can be done both online and offline. Customer can preview their cargo plan if order online. Packer3D provides web-service application so that He can simply access the software package online. [Figure 1-2] is the interface when customer orders the service online. Customer can customize the vehicles for shipping the cargo and choose the container type and size. Then, customer needs to input the dimension and weight information about the items manually. Each information of each box has to be input into the system because it is a crucial data for planning. If a customer wants to deliver a hundred boxes, then He needs to put hundred information about the boxes.



[Figure 1-2] Input Box Information User Interface

After inserting the required data customer can generate the cargo planning using the same software package. The visualization of the result is shown in the [Figure 1-3] and [Figure 1-4].

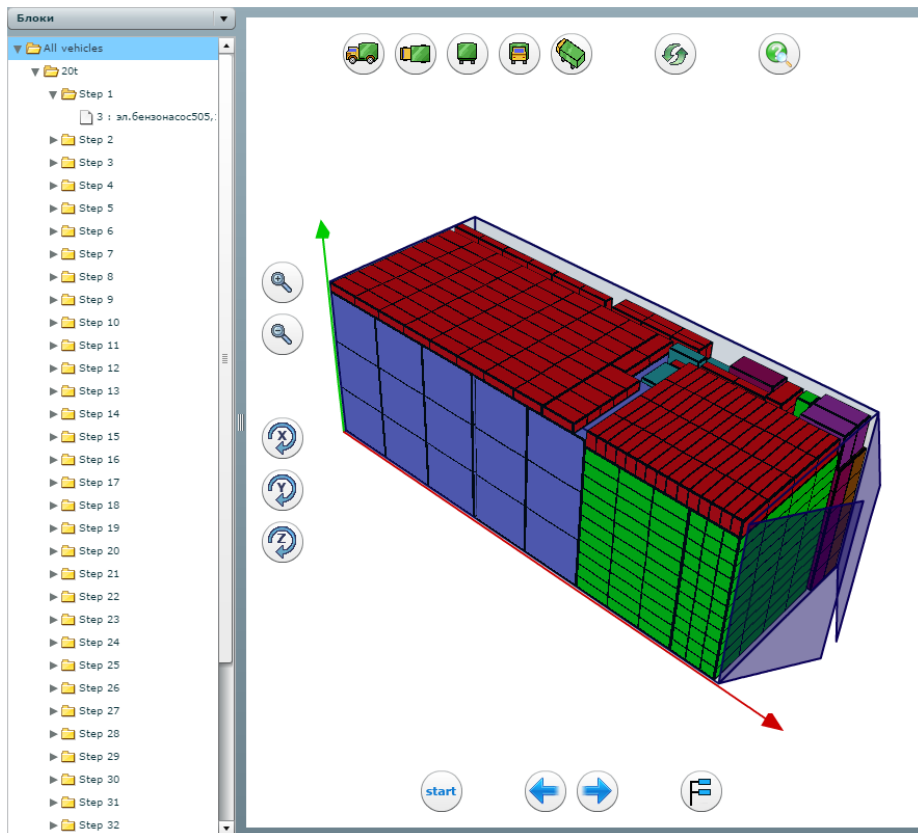
³ <http://www.packer3d.com/>



[Figure 1-3] Side-view of the container

[Figure 1-3] is the side view of the container. In order to view the sequence of loading, customer can explore the items one by one in the left hierarchy structure of the interface. If customer wants to change the angle of the view, customer can click the five view buttons at the top of the screen. So does the rotate and zoom features. [Figure 1-4] shows the other type of view inside the container. The similar software as Packer3D that can be purchased is 3DBP⁴ software. Each application software has their own workflow and interface.

⁴ <https://www.3dbinpacking.com/>



[Figure 1-4] Preview of the whole container

These kind of software packages have improved the representation of the old paper-based blueprint into digital display. The difficulty that was found in reading the manual paper-based solution has been solved by seeing the visualization graphic in digital format. However, the box data information is still input manually and the interface of the visualization is considered as difficult for users who do not have knowledge of how to interact with the software.

In South Korea, the input dimension is solved by implementation of standardize box size. The input process became faster because it is easy to input by choosing

the code number of boxes. For example, post office box in South Korea⁵. However, even though they have standard boxes, even for container cases, sometimes there is a case when one container is packed with boxes from different vendor. In this case, measurement process and data collection is conducted again.

1.1.3. DHL Germany Freight Loading Operation

As the technologies keep improving, a new innovation in similar field has been born. The new way of representing the visualization has been replaced with Augmented Reality (AR). Based on DHL report, DHL implements AR application for freight loading [Glockner, Jannek, Mahn, & Theis, 2014].



[Figure 1-5] DHL Freight Loading AR Application

The field workers are equipped with smart glasses that support AR technologies. When the glasses detect pile of boxes, the field worker will see the arrow in digital form. This arrow will assist the worker and tell which box to pick first, and where to put it inside the container.

⁵ <https://parcel.epost.go.kr/auth.EpostLogin.parcel>

Using this approach, the worker can receive the load instructions in AR visualization format to replace the conventional paper-based load instruction. However, there are some limitation in the system. These limitation are motivated me to do research for filling the existing gap.

1.2. Motivation

Looking at the current practice, there are companies who still use conventional load instruction in the field as described in sub section 1.1.1 and 1.1.2. The newest load instruction application that is described in sub section 1.1.3 is using Augmented Reality (AR). However, there are some limitation in the system. Even though the visualization is more interactive than the conventional load instruction, the visualization system is stand-alone system. Stand-alone means the system is not fully integrated to satisfy the whole process. The information about boxes or cargo items are managed separately. When there are some changes in the data input, the visualization application has to be updated separately to adjust the data input. The current system has not satisfied the handling of data collection yet. Moreover, in the visualization part is also limited to the instruction sequence only. The current system is only focused in assisting the field workers in loading operation.

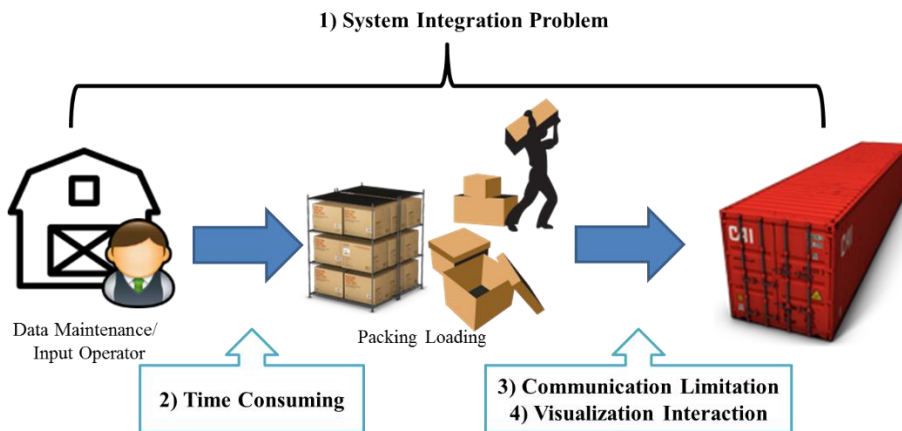
These limitations that are found in the current system are becoming my motivation to conduct research that can solve the problems. By utilizing the newest technology, proposing new approach in improving packing and loading operation is the purpose of the research.

1.3. Problem Description

To standardize the terminology to use in this paper, the large object space to accommodate the small items is called container, while the small items that is packed into the container is called boxes. In container loading application, the objectives can be generalized into two objectives: minimization and

maximization. Minimize the number of container to pack all the available boxes, or maximize the number of boxes into one single container [Bortfeldt & Wäscher, 2013].

This research is not focusing in solving the packing problem itself, but to improve the packing and loading operation execution in the field. The real container loading process is complex, especially when dealing with long distance transportation and the box came from different suppliers [Kappauf, Lauterbach, & Koch, 2012]. To simplify the problem and limit the scope of research, local container loading process inside the distributor center is defined as shown in [Figure 1-6] below.



[Figure 1-6] Container Loading Process

Looking at the scope and features of current application, there are four main problems identified, as follows:

- 1) The current situation does not have integrated system to handle packing and loading operation in the field. The flow of data and decision are managed separately in the current operation.

- 2) The data collection and input consumes time in the current operation since customer or customer service agent has to input each box information to the database system manually. Moreover, the current data collection method is not real-time implementation. If there is some modification, or adjustment in the data, the whole process needs to be re-executed again.
- 3) The communication between field engineer and field worker is limited. In the conventional way, the instruction is communicated using the paper-based solution media, or application media. If mistranslation occurred, the loading error might happen.
- 4) The AR visualization is working as assistance only. The visualization is lacking in interaction factor. The interaction with the AR objects can be the key of real-time simulation to simulate the expected result if the input is changed.

[Figure 1-6] shows where the problems might occur. To run the software optimizer, the application needs input data collected from all the suppliers. Each supplier has their own standardize box's size. Some container companies prepared boxes according to their standard, however if the suppliers are asked to wrap the boxes into their standard boxes, the suppliers will spend additional cost for wrapping the package again. The solution that address this problem is needed. Thus, the solution that is allowing heterogeneous and not a standardized size is needed.

Once the data is collected from different parties or suppliers, the next step is running the software optimizer for finding the packing and loading solution. The field manager who is in charge of making the decision will operate the

software and then decide. The decision of the efficient packing will be delivered to the people in the field who will load the boxes into container. In order to make everything goes as plan, the workers in the field needs to make sure to pack the boxes according to the solution blueprint. Currently, the solution is delivered using application or physical paper. Because of the medium, there is probability to make mistake in loading because of misinterpretation. Thus, integrated and interactive solution for the loading execution process is needed.

1.4. Research Objective and Scope

The objective of the research is to solve problems that are described in section 1.3. They are:

- 1) Reducing the time to input the data manually in the ERP (Enterprise Resource Planning) system database.
- 2) Proposing an approach that allows heterogeneous and unstandardized size boxes to pack.
- 3) Reducing the delay or gap time when delivering the packing solution from field engineer to the workers in the field
- 4) Creating better interface of the solution in the application so that anybody can interpret the solution easy and fast.
- 5) Integrating the whole operation from data collection until the loading operation done.

1.5. Structure of the Thesis

This thesis is organized as follows: Chapter 2 contains the related work that address container loading problem. Chapter 2 also describes some works that might be the answer to solve the identified problem. Chapter 3 describes the proposed architecture of integration model to fill the gap between current situation and new approach. Chapter 4 describes the experiment setup and result of the experiment. Chapter 5 discusses about the conclusion of the thesis, limitation of the proposed prototype, and suggestion to improve it in the future development.

2. Related Work

2.1. Three-dimensional Box Loading Problem

Packing and Loading Operation involves three-dimensional box loading problem within the operation. The problem is how to pack the small boxes into container efficiently. Container loading can be interpreted as geometric assignment where three-dimensional small items have to be packed into three-dimensional rectangular large objects. The objective function is either minimize the number of container or maximize the number of boxes in single container. [Bortfeldt & Wäscher, 2013] summarized that there are 158 papers as for December 2011 that satisfied their criteria. Due to the complexity of practical cases, the number of papers and researches will be developing more.

Based on their paper, the constraints are categorized as follows:

- 1) Container-related Constraints: Weight Limits, Weight Distribution Constraints
- 2) Item-related Constraints: Loading Priorities, Orientation Constraints, Stacking Constraints
- 3) Cargo-related Constraints: Complete-Shipment Constraints, Allocation Constraints
- 4) Positioning Constraints
- 5) Load-related Constraints: Stability Constraints, Complexity Constraints

Three-dimensional item loading problem is particularly NP-hard problem,

which means hard to solve. Thus, only very few exact algorithm exist. Solving such a problem using exact solution may effect in the computation time if the number of instance get larger. Thus, many heuristic approaches are used. This three-dimensional loading problem can be modeled using three-dimensional knapsack model. Martello, Pisinger and Vigo present a main branching tree algorithm. However, they also suggest approximation algorithm to improve the upper bound [Martello, Pisinger, & Vigo, 2000]. Another paper tried to proposed approximation algorithm based on Strip Packing [Diedrich, Harren, Jansen, Thöle, & Thomas, 2008]. However, this algorithm is not practical enough for three-dimensional problem. The algorithm is similar to two-dimensional approach mention in this paper [Burke, Kendall, & Whitwell, 2004], with assuming the infinite dimension for the third dimension. It means that the algorithm only considers two dimension and assume the third one has infinite size.

The container itself may be single container or multiple container. Greedy algorithm can solve single container problem together with the tree-search procedure during the iteration [Lim & Zhang, 2005]. Huang, Hwang and Lu proposed a heuristic algorithm to pack small items from different shops into one single container [Huang, Hwang, & Lu, 2016]. While for multiple container loading problem, modified greedy algorithm can be applied also to solve the multiple container problem [Lim & Zhang, 2005]. Feng, Moon and Shin have proposed hybrid genetic algorithm for the multiple container problem [Feng, Moon, & Shin, 2015].

The type of research in this field is addressing the practical problem happened in the industry and solve it using proposed algorithm. The research keeps

developing with different study cases and different constraints. However, the previous researches do not satisfy the identified problem in section 1.3 yet. It is hard to find the research that address the same problem. The identified problem is not dealing with the technical problem of loading operation, but dealing with the physical operation that almost untouched by many researcher. A lot of system did not record or think about the physical operation. However, this unrecorded operation might cause delay or human error accidentally. In other word, this research is trying to create new approach to solve efficient loading solution with considering the other physical operation.

As summarized by [Bortfeldt & Wäscher, 2013], container loading problem has many applications and constraints. The solution is developed according to the practical study cases. However, it is hard to address specifically one problem, one model, and one algorithm to use in this research. Thus, one general three-dimensional loading problem model has been adapted. The basic and simple model by omitting weight constraints and stability constraints, [Baltacioglu, Moore, & Hill Jr, 2006] model is used. Their research is pallet loading problem for aircraft study case. Their model is a developed model from Ballew's model [Ballew, 2000].

Baltacioglu, Moore and Hill have developed a heuristic algorithm that models human approach in packing the box. The key points of this algorithm are as follows:

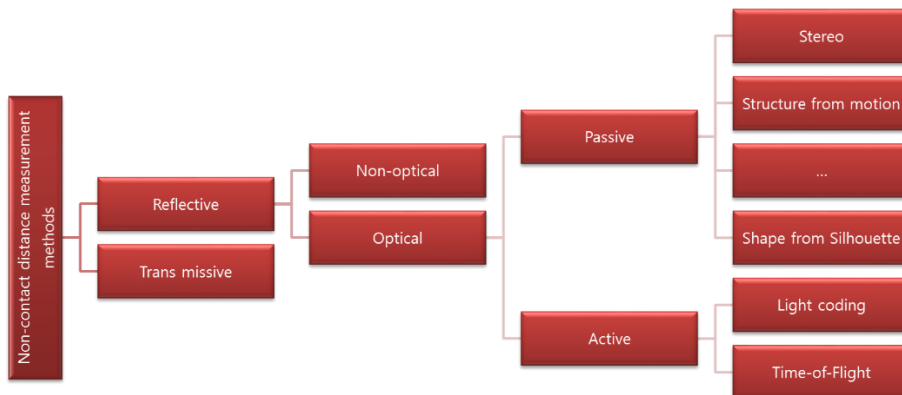
- 1) Human packs from bottom to top
- 2) Human aware of any available gap and pick the most suitable boxes
- 3) Human tends to reduce surface irregularities

They also stated that using greedy heuristic alone is not efficient. While genetic

algorithm is not efficient for solving three-dimensional knapsack problem because of the length of gene. Simulated Annealing is also hard to implement in such a problem [Baltacioglu et al., 2006].

2.2. Automatic Measurement of Dimension

The existence of time of flight camera that can measure distance of an object from camera without any physical contact has motivated many researches to conduct deep research to apply it practically in the industry. Curless [Curless, 1999] has classified the common distance measure used in the practice as shown in [Figure 2-1]. Passive method means measuring three-dimensional distance measurement by way of radiation in visible spectrum that is already present in the scene. The example of the method is stereo method. While active method means measuring three-dimensional distance by projecting the light to the object and get the data when the light is projected back.



[Figure 2-1] Distance Measurement Classification

The example of active method is light coding and time of flight method. Light coding is commonly implemented in cheap camera. Light coding calibrate the camera using triangulation method. Trobina introduced the camera principle to measure the distance using triangulation method and general least squares calibration method [Trobina, 1995]. Triangulation is used to calibrate the camera, so that camera coordinate and world coordinate is similar. Researches related to this work are trying to minimize the difference between coordinate and world coordinate value. Sung Joon Ahn proposed calibration techniques that can measure the dimension of primitive objects, such as cube, cylinder, ball using least squares orthogonal distance fitting [Ahn, 2004].

Many consumer level cameras use light coding method in the measurement process. Some of popular camera that apply active method is Microsoft Kinect and Intel Real sense. Microsoft Kinect applied the similar method to measure the distance [Mutto, Zanuttigh, & Cortelazzo, 2012]. Intel Real Sense has same method in retrieving the distance of an object from the camera. The detail about Microsoft Kinect and Intel Real Sense camera is described in the [APPENDIX A.1].

Many researches in different field utilized these cameras to solve the automatic measurement part. In agriculture field, Kinect can be used to measure the size and volume of onion [Wang & Li, 2014]. Sung Joon Ahn also applied his developed algorithm to tracking certain primitive shapes and measure the dimension of it. His algorithm is applied in Logistic field⁶. However, there is

⁶ <http://curvsurf.com/>

no research has been conducted yet in container loading operation yet. Especially the research that is trying to use depth camera to measure the dimension of the boxes for efficient packing.

2.3. Augmented Reality Visualization

In order to satisfy the demands of customers, reducing the operation time and operation error is necessary. Current software that is used in the industry as described in section 1.1, it still lack in automation, integration and interaction part. The well-known Virtual Reality (VR) technology is used for training and simulation in the industry. For example, Virtual Reality can be utilized to train workers how to maintain the underground lines in power distribution system. Because maintaining power distribution system needs a high-qualified worker due to its complexity. Directly put the worker into the field without proper training is dangerous. Thus, virtual environment is created [Galvan-Bobadilla, Ayala-Garcia, Rodriguez-Gallegos, & Arroyo-Figueroa, 2013]. However, in Virtual Reality, the physical objects are transformed into virtual objects. Users interact with virtual objects and try to alter them without changing the physical environment. This approach is good, however in the practical industry, they are dealing with real objects. In order to satisfy those demands, Augmented Reality (AR) technology is used.

Augmented Reality (AR) is able to combine digital contents with physical environment. Caudel and Mizel develops augmented reality application to guide technician in building a wiring system that forms part of an airplane. Storing the instruction in electronic form, will save space and reduce cost [Caudell & Mizell, 1992]. There are some companies that utilized augmented

reality applications, such as DHL [Glockner et al., 2014]. DHL implements augmented reality technology in logistic sector, such as warehouse, transportation optimization, delivery and so on. In the warehouse, in order to reduce picking errors and search time, picking staffs are equipped with wearable AR to assist them in picking process [Schwerdtfeger, 2010]. However, not many researches in loading problem are talking about the visualization using augmented reality yet. This motivates the author to improve the visualization method in packing and loading from usual graphic representation to augmented reality representation.

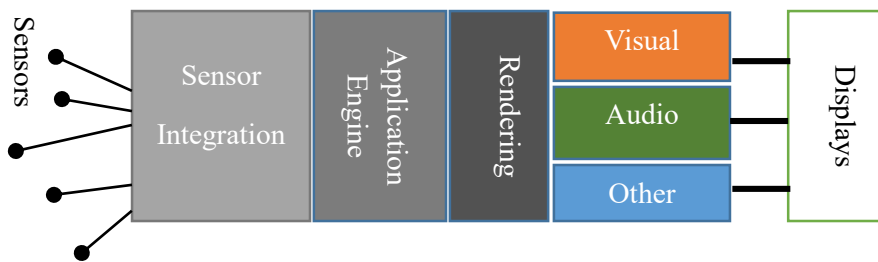
Ronald T. Azuma [Azuma, 1997] defines in his paper that Augmented Reality (AR) has three general characteristics:

- 1) combine real and virtual,
- 2) interactive in real-time,
- 3) and registered in 3D.

In general, Augmented Reality (AR) is applied for improving how human interact with objects by combining real world and virtual objects and allow human to alter or modify the objects in real-time. Augmented Reality is used in many different application areas such as education, entertainment, medical and many more areas.

In his book [Craig, 2013], he explains how Augmented Reality (AR) system works. [Figure 2-2] shows the diagram of relationship between the core components, both hardware and software, of an Augmented Reality system. The result of the components is fed to the main AR application. The system renders the information as signals that are fed to the various displays. When

human alter the virtual objects, the component will detect the action and recognize it as trigger and system updates the information. These processes repeat constantly, and reflect the changes into the display whenever changes occurred.



[Figure 2-2] Core Components of Augmented Reality

Based on DHL report [Glockner et al., 2014], in best practice, Augmented Reality is used 4 (four) aspects, as follows:

- 1) Context-sensitive Information – Information at the Right Time and Place
- 2) Enhances Senses
- 3) Mixed-reality Simulation – Exploring the Virtual in the Real
- 4) Virtual Interfaces – Controlling the Real Through the Virtual

Regarding to these best practices, the idea of utilizing Augmented Reality technology to solve the real-time and result visualization problem in packing loading problem could be a promising solution in the future.

2.4. End-to-end System Integration

Existing applications and previous researches in packing and loading problem have not addressed integration system yet. In order to decrease the time delay between each process in packing and loading operation, to integrate the whole process as one and real-time execution would be better. The idea of connecting depth camera and augmented reality to create natural interaction between human and computer has been conducted [Piumsomboon, 2015]. However, the combination of depth camera and augmented reality in packing and loading operation has not been conducted yet.

In summary, the real-time and integrated approach to help logistic business increase the performance by minimizing the time operation and preventing the human error can be achieved by utilizing depth camera and augmented reality.

3. System Design

In this chapter, the design of new approach of packing application is described. The system has been designed for certain user model and scenario to create a product with better interaction design and also to limit the scope of research at the same time. The aim of the new proposed approach is reducing the execution time of packing in the field and improving the interaction between human and application.

3.1. Design of Operations

Before defining the detail user scenarios, user model and user story has been defined in sub section 3.1.1 and sub section 3.1.2 respectively.

3.1.1. User Models

User models are representing the model of the actual user, including work roles, sub-roles, user class definitions and personas [Hartson & Pyla, 2012]. Refer to the current system process described in section 1.1, there are three users in the system, as follows:

- Data Maintenance Operator

Data maintenance operator has role to maintain the arrived demand from customer, or supplier, and input it into the company database.

- Field Engineer

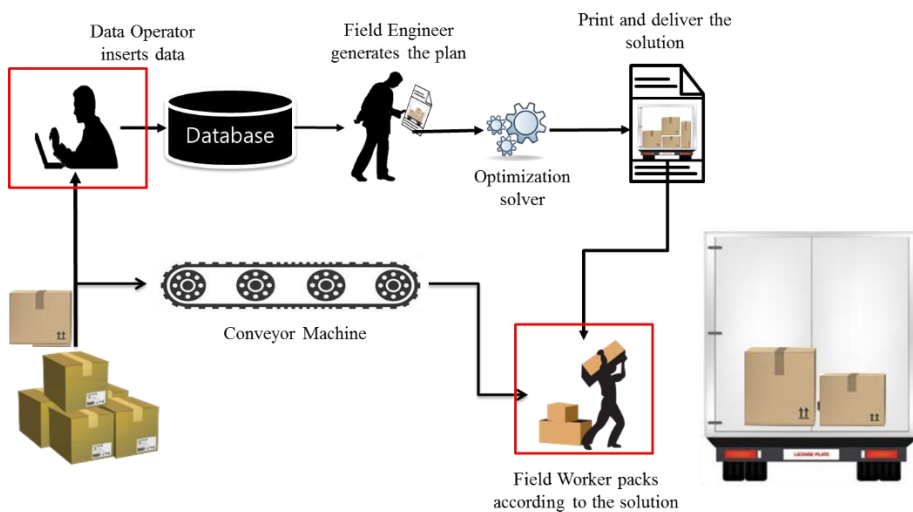
Field Engineer's role is making decision of how to allocate the available boxes to the container. Field Engineer utilizes packing solver solution to find the solution of packing. After getting the result, He sends the solution to worker in the field.

- Field Worker

Field Worker has role to pack the arrived small boxes into a container based on the solution given by the manager.

3.1.2. User Scenario

[Figure 3-1] illustrates the current system flow of packing and loading operation.



[Figure 3-1] As-Is System Illustration

When the order came, data maintenance operator or customer himself needs to

record the necessary data into the company database manually. Field engineer needs to make decision of how to pack and load the boxes into container. Somehow, field engineer wants to see the current situation of the warehouse depending on how many boxes are available currently. However, that information can only be seen if the data operation manager updates the data in the company database regularly. He found that sometimes the data is updated after all the boxes are collected in the end of deadline. Then, he could not preview the real-time situation of the warehouse whenever he wants. When deciding the loading plan, field manager chooses to solve it using software package. After the solution has been generated, field manager prints the solution and sends it to workers in the field.

A field worker has received the blueprint solution from his/ her field engineer through company application. He uses the enterprise application to view the solution. However, He could not understand the exact position of the boxes inside the container. He can only see the sequence of packing from the inside layer to outside layer, but it needs time to translate the information to execute in real environment. Reading the solution and then packing process is done repeatedly which is time consuming. He needs easier and faster method to assist him in packing the boxes inside the container.

3.2. System Design Framework

3.2.1. Gap Analysis

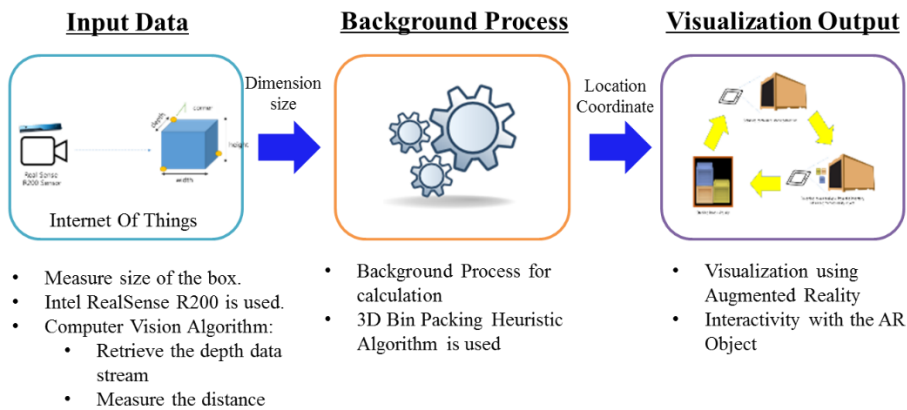
By analyzing the user scenario described in sub section 3.1.2, the sub section has specified the gaps in user's knowledge while operating the current application. This gaps leads to interaction difficulties and data collection problem.

Having knowledge of related works that has been described in Chapter 2, the idea of using Augmented Reality as visualization media has been chosen for the new proposed approach. Furthermore, the depth camera is also utilized to automatically measure the box dimension automatically. Finally, both depth camera and augmented reality, both input and output, are integrated as one system.

3.2.2. System Design

In 3D box packing case, the key point data is dimension of the boxes. In order to support the real-time augmented reality visualization, the system is designed to record the dimension data automatically instead of manually record.

In order to generate the packing solution, certain algorithm to calculate the efficient packing has to be implemented inside the solution. Last but not least, the result of the packing decision has to be visualized as easy as possible to assist the operator worker in the field.



[Figure 3-2] Proposed System Overview

[Figure 3-2] represents the overview of the new proposed solution. The new system is designed as modular system to create the flexibility both development and implementation when using the application. The system is divided into three modular parts and integrate it with developed interface to manage the data translation and data flow between the systems. The system consists of three modules described as follows:

- **Input Module**

Input module handles the data collection of the dimension boxes' data. To automate the collection process, depth camera is utilized as the sensor. Depth camera has capability to retrieve distance data to make the box dimension's measurement get easier.

- **Process Module**

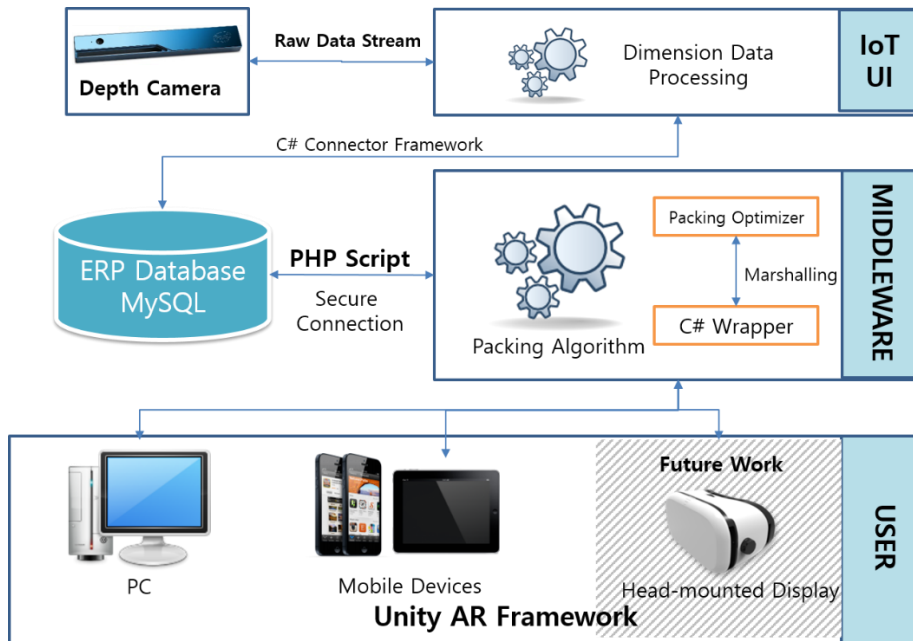
The process module contains packing algorithm to solve the 3D box packing problem and give best decision to pack.

- Output Module

In order to create an interactive user-friendly visualization, the result of packing solution is visualized with 3D graphical model that is rendered in world physical environment. Augmented reality technologies allow people to interact with digital object in real physical world.

3.3. Integration Model Architecture

The challenge of the new approach is how to design a framework which able to connect the camera as physical equipment and human as end-user. Naturally, camera does not have ability to communicate with end-user without having specified middleware and algorithm. The middleware also has to be developed based on problem case that needs to be solved. Each module has different foundation of technology to start with. To make those three modules interact with each other, some integration model is developed.



[Figure 3-3] Integration Model

[Figure 3-3] represents the integration model that let the camera communicates with the end-users. The raw data stream is captured by the developed interface. This interface application will map the world coordinate to camera coordinate. World coordinate is the coordinate of the object in the real world. The mechanism to transform the coordinate into camera coordinate is needed. After the dimension data is obtained, the data is stored in ERP (Enterprise Resource Planning) system database so that any other application can re-use the data for other requirement. Database is built in MySQL platform.

Middleware in this system is the program to connect the database with the end-user application. End-user application is built using Unity Framework⁷ as the

⁷ <https://unity3d.com/>

development environment and Vuforia⁸ as the Augmented Reality's Library. Inside the middleware, a packing library that use heuristic algorithm [Baltacioglu et al., 2006] has been implemented using C language. This packing algorithm needs dimension boxes from database as input before processing the calculation. To connect with the database, connection interface has been developed in PHP script language⁹.

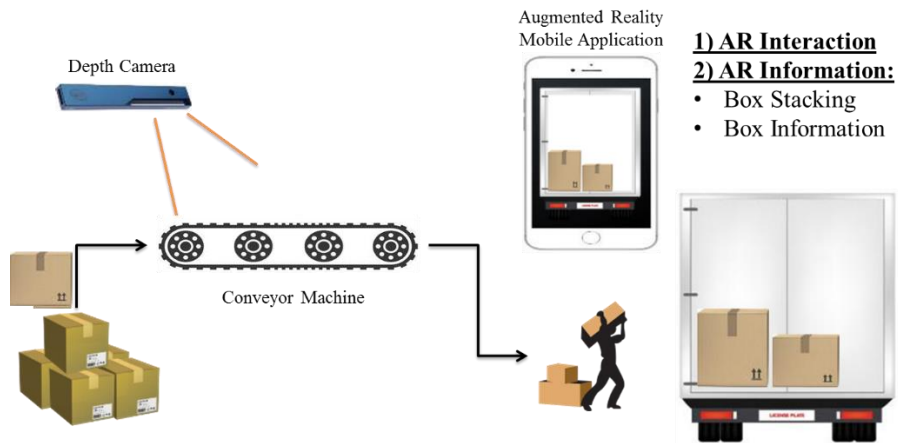
The application user interface is developed for desktop application and mobile application. Since the application user interface is built using Unity framework, there is possibility to extend it into Head-mounted Display Gear [Wheeler, 2016] in the future. Head-mounted Display is display device that is worn on the head. This device is equipped with small display optic that reflect the digital objects and allow the user to see it. This application is used to see the visualization of the packing solution. In order to make it interactive and more real, the interaction with the digital object is implemented.

3.4. System Flowchart

This proposed approach needs assumption in the system to work. The area of bin packing application of the new proposed approach is wide. Any area that needs bin packing solution to utilize the space can apply this approach for their system. Thus, the prototype development has been designed for general space utilization purpose and assume some user scenarios to be applied.

⁸ <https://www.vuforia.com/>

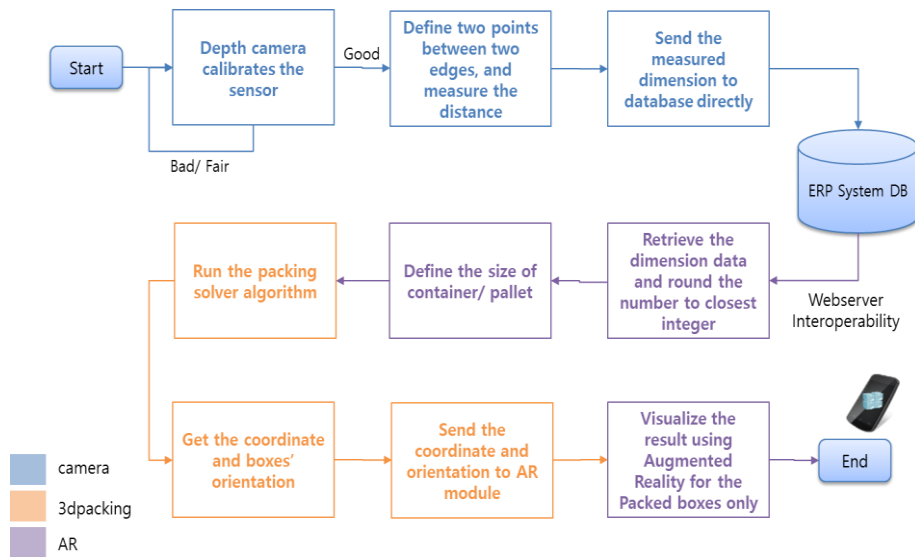
⁹ <https://secure.php.net/>



[Figure 3-4] User Scenario for Implementation

[Figure 3-4] shows the illustration of user scenario for the implementation of new proposed approach. Assume that the depth camera is installed in fix position near the conveyor belt. Positioning the camera so that the camera can see the boxes when put on top of the conveyor. In order to get the information of each box, all the boxes have to be passed through the conveyor. After all of the boxes dimension has been recorded, the boxes are ready to be loaded into the container.

The field worker may use the application to see the load instruction in augmented object form. Then the digital object can be modified to simulate the new result based on new modification or other condition.



[Figure 3-5] Flowchart of proposed system

[Figure 3-5] is the flowchart of whole system scenario. When data maintenance operator wants to record the information of each box, He put the box on the conveyor. He positions it within the camera range. Data maintenance operator starts to operate the measurement application interface and let the depth camera starts to calibrate the camera. Once the good signal is caught, the edge of the box can be designated. In order to get the third dimension, the box is rotated 90 degrees clockwise.

Field workers wants to get the solution of loading. He opens his mobile device and run the program application. The program calls the dimension data from database. Using those data, it triggers the packing optimization solver directly. The solver gives the coordinate and orientation of the available boxes inside the database. Finally, the result is visualized and displayed in the mobile screen.

3.5. Database Design

This database prototype is created to represent the integrated ERP (Enterprise Resource Planning) system inside the company. This database is simpler than the real system, however enough to support the prototype system. The database is prepared to store the dimension data of the boxes that are scanned by the depth camera. The structure of the database table is described in [Table 3-1].

[Table 3-1] Database Structure for Box Data

Column Name	Data Type	Notes
boxID	Int	Primary Key, Auto Increment, Not Null
dim1	Double	Store the first dimension of the boxes (width)
dim2	Double	Store the second dimension of the boxes (depth)
dim3	Double	Store the second dimension of the boxes (height)

C# connector in .NET framework handles the connection from depth camera interface. While the connection between database and unity framework is handled by PHP script. Using this method is more secure than direct connection because in PHP provide mechanism for secure connection.

3.6. Output Module: AR Visualization

The most important part of the solution is the visualization part. The idea is using augmented reality technology to improve the interaction between human and system. The visualization will project the digital boxes into the physical world depending on what is available inside the database. In other word, the digital boxes that are seen is the boxes is the boxes that are available physically inside the inventory or distribution center. Refer to system flowchart in section 3.4, the system needs the dimension data from database for the input, and needs the coordinate where the boxes will be put inside the container. In order to make the packing solution is optimized, optimization algorithm has been deployed. The algorithm is described in sub section 3.6.1 below. The optimization algorithm can be changed or modified depending on the practical requirement. The reason of implementing this algorithm is because of simplicity and generality.

3.6.1. Packing Algorithm

As described in section 2.1, the model used in this research is adapted from Ballew's formulation [Ballew, 2000]. Baltacioglu, Moore and Hill developed Ballew's formulation by omitting weight and stability constraints. They proposed new heuristic approach to solve the problem [Baltacioglu et al., 2006]. The summary of the algorithm is described in [APPENDIX B].

This formulation is designed for three-dimensional packing problem.
Following is the description the variables used in the formulation:

E	=	The amount of unoccupied space on the container
L	=	The length of the container
W	=	The width of the container
H	=	The height of the container
N	=	The number of boxes to be packed
PV	=	The pallet volume available for packing
L _x	=	The length of box x; where x goes from 1 to N
W _x	=	The width of box x; where x goes from 1 to N
H _x	=	The height of box x; where x goes from 1 to N
V _x	=	The volume of box x; where x goes from 1 to N
B _x	=	Binary variable used to represent whether box x is packed or not; where x goes from 1 to N 0 = Box x is not packed 1 = Box x is packed
B _{ijkx}	=	Binary variable used to represent whether part of box x is packed in container location i, j, k; where i goes from 1 to L, where j goes from 1 to W, where k goes from 1 to H, where x goes from 1 to N. 0 = Box x is not packed in container location i, j, k 1 = Box x is packed in container location i, j, k

The formulation is as follows:

Minimize E

Subject to following constraints:

$$\left(\sum_{x=1}^N B_x * V_x \right) + E = PV \quad (1)$$

$$\sum_{x=1}^N B_{ijkx} \leq 1; \quad \forall i \text{ from } 1 \text{ to } L, j \text{ from } 1 \text{ to } W, k \text{ from } 1 \text{ to } H \quad (2)$$

$$\left(\sum_{i=1}^L \sum_{j=1}^W \sum_{k=1}^H B_{ijkx} \right) - B_x * V_x = 0; \quad \forall x \text{ from } 1 \text{ to } N \quad (3)$$

The objective of the model is to minimize the unused space, or maximize the utilization space of the container by calculating the how many percentage of container volume utilized. The assumptions of the model are following:

- 1) All items are rectangular boxes
- 2) Allow overhang and unstable packing
- 3) Omit all constraints dealing with weight and center of gravity
- 4) Boxes can be rotated and packed with one of six different options

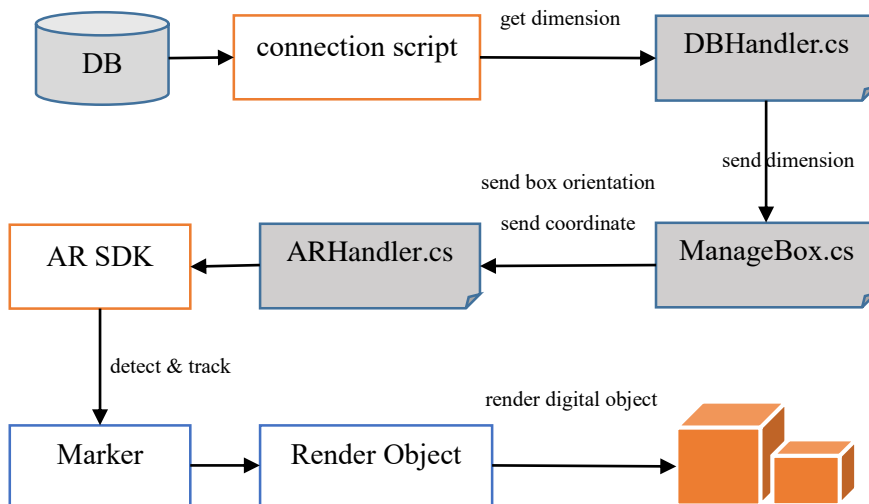
Other than these assumptions, the constraints have been applied to the model. Constraint (1) ensures the volume of the packed boxes is less than the available volume that can be packed. Constraint (2) is a set of constraint ensuring no more than one box is placed in each cubic location on the container or no overlapping is allowed. Constraint (3) ensures each packed box is packed with the correct volume.

[Baltacioglu et al., 2006] mentions that his heuristic approach is called adaptive heuristic algorithm because the algorithm is modelling human intelligence

approach when packing the box. Human packs the boxes from bottom to top, from the deepest location to the most outers location of the container. Human aware of any available gap and pick the most suitable boxes to keep the stacking as smooth as possible.

3.6.2. AR Visualization Process

The algorithm described in sub section 3.6.1 gives the boxes orientation and coordinate of each box that is packed inside the container. Using these known data, digital boxes is rendered into the scene. The application is developed using Unity framework and Vuforia library for their development flexibility, because more devices are targeted for future development. [Figure 3-6] shows the structure of the developed application for visualization.



[Figure 3-6] Structure of AR Application

[Table 3-2] AR Application Structure Description

Object Name	Description
DBHandler.cs	A code file to manage the connection between Database and Application
ManageBox.cs	A code file to create the digital boxes in the correct coordinate location
ARHandler.cs	A code file to render the digital boxes to the display
AR SDK	This research used Vuforia as the AR SDK Library

Direct connection from AR application to Database is not recommended because of security issues. Therefore, a separated script has been developed to manage the connection. There are three C# file source codes inside the AR application. DBHandler.cs retrieves the box dimension data from database and convert the unit measurement to match the Unity's unit measurement. The unit measurement from database is centimeters. If the scale were not scaled down, the rendered digital objects would appear outside the boundary of the AR region. Consider the Unity treats one unit as one meter, then the original size from database has to be divided by 100.

After the scale is converted, the ManageBox.cs create the digital boxes according to the scale and position of each boxes. These boxes are stored inside the AR application and ready to be displayed in the monitor or mobile devices. This displaying role is handled by ARHandler.cs file and Vuforia Library. The detail of the source code is attached in the [APPENDIX C].

3.6.3. Interaction Features of AR

Displaying the augmented contents only is not the only purpose of using augmented technology in this research. Adding more interaction capability to the digital contents is another benefit that the users can get through the application. Henrysson, Marshall and Billinghamurst has experimented several 3D interaction with mobile phone and augmented reality in [Henrysson, Marshall, & Billinghamurst, 2007]. This research is trying to experiment with the simple interaction features using finger touch. There are four (4) touch features: Select, Translate, Rotate and Scale, that have been developed in this research. The description is shown in [Table 3-3].

[Table 3-3] Interaction Description

Interaction Features	Description
Select	Track the finger touch input location to identify which object is selected. Change the current selectable object to green color.
Translate	Translation coordinate (Tx,Ty,Tz) is represented as follows: $T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$ Thus, the new coordinate when translation applied is,

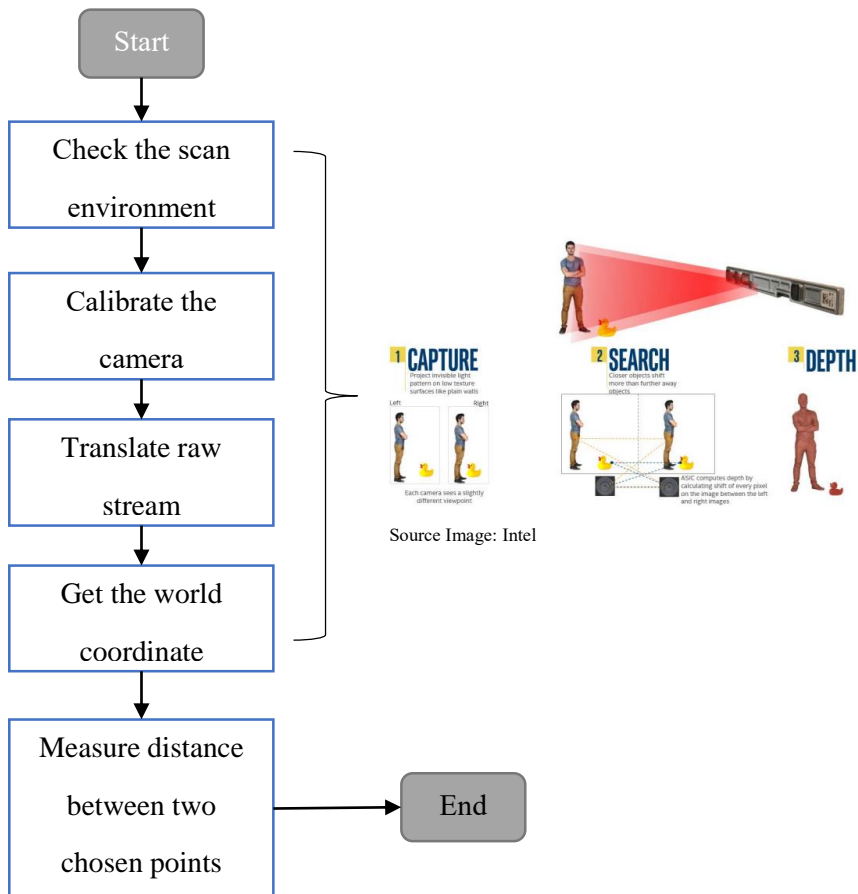
Interaction Features	Description
	$[X' \ Y' \ Z' \ 1]$ $= [X \ Y \ Z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$ $= [X + t_x \ Y + t_y \ Z + t_z \ 1]$
Rotate	$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$ $R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$ $R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Scale	$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ <p>S is scaling factor. Mathematically, the new scale is represented as follows:</p> $[X' \ Y' \ Z' \ 1]$ $= [X \ Y \ Z \ 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $= [X.S_x \ Y.S_y \ Z.S_z \ 1]$

Adding interaction using touch gesture in AR application is not easy to implement. The mechanism to track and store the information about the finger

movement is needed. Furthermore, the object location in the digital space and world space needs to be synchronized also.

3.7. Input Module: Data Collection

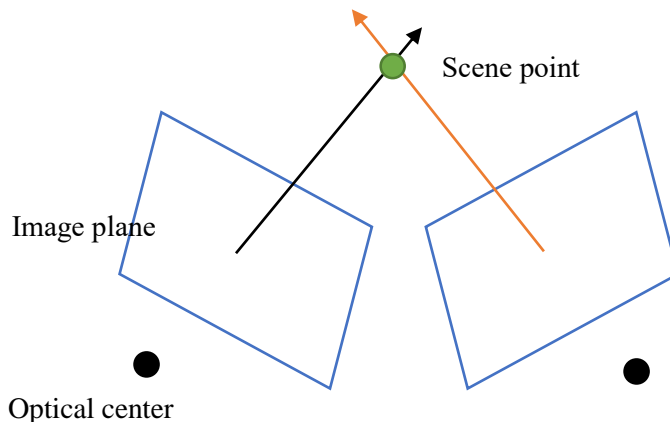
Data is mandatory element to have before decide the packing solution. In order to make decision, valid data need to be maintained before deciding. Instead of manual input that might be time consuming, automatic data collection has been proposed. Depth camera has capability to see as what human can see. This research utilizes depth camera for dimension measurement's role. In packing solution, the dimension of small boxes is important input to make decision.



[Figure 3-7] Depth Camera Measurement Algorithm

Checking the quality of scanning environment is important since the condition of environment affects the tracking and detecting quality of the camera towards the objects. Depth camera that is used in this research is Structured-Light-based camera. Patterns, surfaces, perspective and geometric distortion has to take into account when looking for a good quality environment to identify the object models.

After good signal is detected by the camera, the next step is calibrating the world coordinate between RGB color sensor and two infrared sensors. Depth camera has two infrared sensors that are located separately. The location of RGB color sensor is also different. In order to translate it into one single point, calibration method is needed. Triangulation method is used in this research.



[Figure 3-8] Triangulation Basic Principle

Triangulation basic principle is determining the scene point from a set of corresponding image plane locations and known camera optic position. Two optical centers in [Figure 3-8] represents the two infrared sensors in the depth camera. Scene point refers to a point in world coordinate that the camera is

trying to locate [Szeliski, 2010]. After getting the estimated world coordinate, the designated two points are obtained Euclidean distance formula is used to calculate the distance between two points. The Euclidean distance formula is described as follows:

$$\text{distance } (p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$$

These calculation is encapsulated in the Intel Real Sense SDK [Intel]. The SDK also will show the confidence level of the camera measurement's result. The code snippet is described in [APPENDIX A.2].

4. Experiment

Using the methodology that is described in Chapter 3, prototype system is developed to validate the methodology. Prototype is also to provide a more realistic mock-up other than simple paper mock-up of the new system under development [Maguire, 2001]. Eleven (11) boxes from three (3) different suppliers are prepared for testing experiment. Suppliers have their own standard box's size. This experiment prepares three (3) different size of boxes to represent the standard size from each suppliers. The use case scenario used is these three suppliers are going to deliver package to another country using shipping container. In the distribution center, they place and register the boxes by themselves using the new system that has been developed according to proposed approach explain in chapter 3.

4.1. Experiment Setup

The boxes for experiment is shown in [Figure 4-1] below. There are eleven (11) boxes with three (30 different sizes that came from 3 different suppliers. The detail of dimension is described in [Table 4-1].

[Table 4-1] Dimension Data and BoxID Data

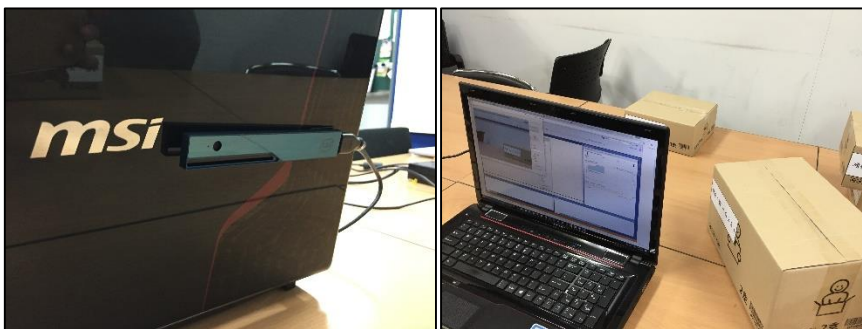
Label	boxID	Width	Depth	Height	Quantity
1 ☐	3,4,5,6	22 cm	19 cm	9 cm	4
2 ☐	7,8,9,10,11	27 cm	18 cm	15 cm	5
3 ☐	1,2	34 cm	25 cm	21 cm	2

boxID is the identification number that is generated by the database as primary key of the boxes. Assume that each label represents each supplier's standard boxes. The dimension of width, depth and height written in [Table 4-1] is the actual dimension data of the boxes.



[Figure 4-1] Boxes for Experiment

The environment setup for measuring the boxes is shown in [Figure 4-2]. The depth camera is attached in front of notebook. The interface application that has been developed is installed on the notebook. The detail of application is described in sub section 4.2.1. Place the box in front of the notebook to be measured.



[Figure 4-2] Depth Camera attached (Left) Experiment Setup (Right)

The detail equipment and software that is used for this research is described in [Table 4-2] as follows:

[Table 4-2] Prototype Development Tools

System Module	Equipment	Development Environment	Library	Language
Input Module	Real Sense R200	Visual Studio 2015	RealSense SDK 10.0.26.396	C#
Packing Optimizer	-	Visual Studio 2015	-	C
Visualization	Webcam/ inbuilt camera	Visual Studio 2015, Unity 5.4.2	Vuforia 6.2.10	C#
Database Engine & Connection	MySQL Workbench 6.3	Notepad++	-	PHP

4.2. Experiment Result

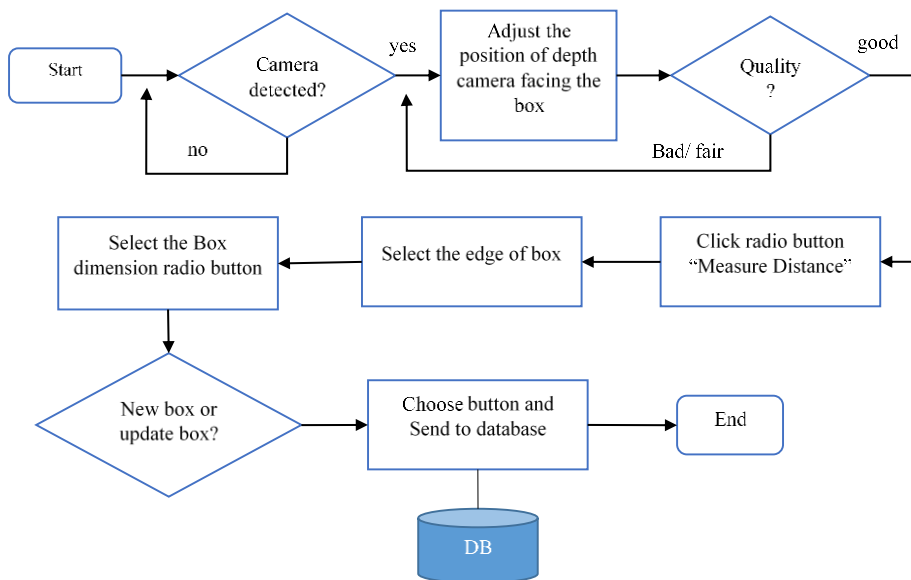
4.2.1. Input Data

Before do the experiment, the depth camera has to be plugged into the notebook and the developed interface application, as shown in [Figure 4-3], has to be installed as prerequisite.



[Figure 4-3] Box Measurement Interface Application

The process of collecting the dimension of each box using the interface is as shown in [Figure 4-4].



[Figure 4-4] Flowchart of using the interface application

4.2.1.1. Result

The result of measurement is stored in database with automatic generated boxID based on the sequence of when the boxes were scanned. [Table 4-3] is the summary of the measurement result. The table shows the difference between dimension from the sensor and dimension of actual size. The mean difference of each dimensions: width, depth and height is calculated using following equation:

$$\text{Mean Difference} = \frac{\sum_{i=1}^n |\text{sensor}_i - \text{actual}_i|}{n}$$

[Figure 4-5] is the screenshot of database that is connected directly with the interface application.

	boxID	dim1	dim2	dim3
	1	33.53	24.63	20.58
	2	33.8	24.36	20.48
	3	21.16	18.51	8.57
	4	21.53	18.45	8.94
	5	21.43	18.44	9.05
	6	21.41	18.53	8.93
	7	26.41	17.63	14.91
	8	26.41	17.6	14.81
	9	26.39	17.37	14.4
	10	26	17.28	14.43
	11	25.99	17.59	14.46
	NULL	NULL	NULL	NULL

[Figure 4-5] MySQL Database Data

[Table 4-3] Measurement Result

boxID	Width (cm)		Depth (cm)		Height (cm)		Difference between Sensor and Actual		
	Xsensor	Xactual	Ysensor	Yactual	Zsensor	Zactual	Xsensor - Xactual	Ysensor - Yactual	Zsensor - Zactual
1	33.53	34	24.63	25	20.58	21	0.47	0.37	0.42
2	33.8	34	24.36	25	20.48	21	0.2	0.64	0.52
3	21.16	22	18.51	19	8.57	9	0.84	0.49	0.43
4	21.53	22	18.45	19	8.94	9	0.47	0.55	0.06
5	21.43	22	18.44	19	9.05	9	0.57	0.56	0.05
6	21.41	22	18.53	19	8.93	9	0.59	0.47	0.07
7	26.41	27	17.63	18	14.91	15	0.59	0.37	0.09
8	26.41	27	17.6	18	14.81	15	0.59	0.4	0.19
9	26.39	27	17.37	18	14.4	15	0.61	0.63	0.6
10	26	27	17.28	18	14.43	15	1	0.72	0.57
11	25.99	27	17.59	18	14.46	15	1.01	0.41	0.54
Mean Difference							0.63 cm	0.51 cm	0.54 cm

4.2.1.2. Analysis

The result of experiment that is summarized in [Table 4-3] shows that the average difference of each dimension is less than 1 cm. In container loading application, one centimeter different is acceptable because the space of container is very huge. Thus, the result of measurement from sensor is acceptable to use for next process.

Additional notes for this distance measurement experiment is that the boxes need to be rotated to retrieve the third dimension. This is the limitation of this system. The mechanism of detecting and measuring the sizes will be improved in the future work.

4.2.2. Visualization Output

The packing solution can be generated anytime and anywhere by field worker or by field manager. The application will read any available data inside the database at that moment when generating the solution through application. The visualization is shown using Augmented Reality (AR) technology for more interaction.

The experiment is trying to pack all available eleven (11) boxes into four (4) different containers. The container data is shown in [Table 4-4].

[Table 4-4] Container's size for experiment

No	Container (cm)		
	Width	Depth	Height
1	104	96	84
2	50	150	100
3	50	50	50
4	30	30	30

In order to trigger the application to generate the digital graphics of boxes in the display screen, the image marker that can be stuck into surface of the container is prepared.



[Figure 4-6] Image Marker for Experiment


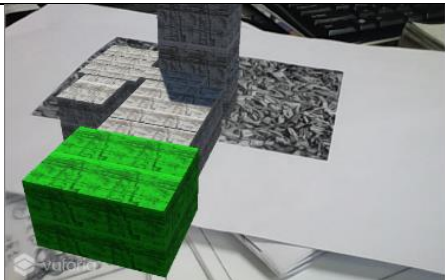
[Figure 4-6] is the image marker used in the experiment. The image that is used for target marker has to have many features because it influences the detection

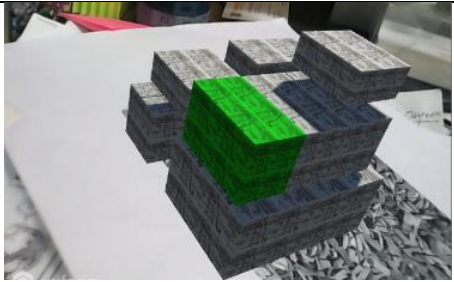
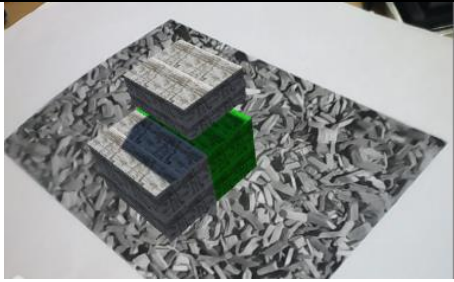
and tracking process of the camera. The PTC, developer of Vuforia library, defines the features of an image by analyzing whether a sharp, spiked or chiseled detail is present in the image [PTC].

4.2.2.1. Result

The experiment is conducted using desktop application interface and webcam. The webcam tracks and detects the image target prepared. The result of each packing solution is shown in [Table 4-5] below.

[Table 4-5] Visualization AR of Packing Solution

No	Container's Size	Result
1	104 cm x 96 cm x 84 cm	
2	50 cm x 150 cm x 100 cm	

No	Container's Size	Result
3	50 cm x 50 cm x 50 cm	
4	30 cm x 30 cm x 30 cm	

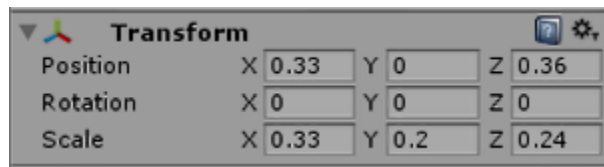
The digital object is generated in the physical world. The green part shows that user choose one box to view the detail information about that box. The information can be anything depends on the requirements, such as dimension of the boxes, the name of the boxes and so on. The features that are available and applied in this research is not only choosing by clicking the digital boxes, but also modifying the shape of the digital boxes, such as moving, rotate and scaling. This features show the interactivity benefit of the augmented reality technology compare to other technologies.

4.2.2.2. Analysis

The digital object of the boxes is located based on the coordinate that is calculated from the packing algorithm. The heuristic algorithm that is described in sub section 3.6.1, takes the dimension from database as input and determine the coordinate as output. This output is used by the end-user application to

display the digital object in physical world as augmented reality application.

To validate the result, the packing result and coordinate that are calculated by the augmented reality application is compared with the packing algorithm by running it separately. Unity has inspector window as shown in [Figure 4-7] that can be used to analyze the position and location of the objects in the plane axis.



[Figure 4-7] Box 2 Object Data

To compare both result, data of container 1 is used. [Figure 4-8] is the result of container 1 (104 cm x 96 cm x 84 cm) by running it separately using command prompt in Windows OS.

```
VARIANT: 1; ITERATION (TOTAL): 1; BEST SO FAR: 0.000 %; TIME: 0
1 33 24 20 0 0 36 33 20 24
2 33 24 20 33 0 36 33 20 24
3 21 18 8 89 0 0 8 18 21
4 21 18 9 26 0 0 21 18 9
5 21 18 9 47 0 0 21 18 9
6 21 18 9 68 0 0 21 18 9
7 26 17 15 0 0 14 26 17 15
8 26 17 15 78 0 21 26 17 15
9 26 17 14 26 0 9 26 17 14
10 26 17 14 52 0 9 26 17 14
11 26 18 14 0 0 0 26 18 14
ELAPSED TIME : Almost 0 sec
TOTAL NUMBER OF ITERATIONS DONE : 1
BEST SOLUTION FOUND AT : ITERATION: 1 OF VARIANT: 1
TOTAL NUMBER OF BOXES : 11
PACKED NUMBER OF BOXES : 11
TOTAL VOLUME OF ALL BOXES : 77098
PALLET VOLUME : 838656
BEST SOLUTION'S VOLUME UTILIZATION : 77098 OUT OF 838656
PERCENTAGE OF PALLET VOLUME USED : 9.193042 %
PERCENTAGE OF PACKEDBOXES (VOLUME) : 100.000000%
WHILE PALLET ORIENTATION : X=104; Y=96; Z= 84
```

[Figure 4-8] Container 1's packing result

Using the algorithm, all of available boxes can be loaded into the container. In the case of container 3, the size 30 cm x 30 cm x 30 cm is small enough to store all the 11 boxes. The result as shown in [Table 4-5], only 3 boxes can be pack inside this container.

[Table 4-6] is the summary of comparison between both results. The AR application's result is the original size divided by 100 to let the object appear within the AR region boundary. Without dividing it by 100, the digital object will be very huge. The measurement unit in Unity is different from reality. To make the object visible by the camera, the object is scaled down.

[Table 4-6] Size and Coordinate Data Comparison

BoxID	Size (w x d x h)		Coordinate (x,y,z)	
	Algorithm	AR Application	Algorithm	AR Application
1	33 x 20 x 24	0.33 x 0.2 x 0.24	0,0,36	0,0,0.36
2	33 x 20 x 24	0.33 x 0.2 x 0.24	33,0,36	0.33,0,0.36
3	8 x 18 x 21	0.08 x 0.18 x 0.21	89,0,0	0.89,0,0
4	21 x 18 x 9	0.21 x 0.18 x 0.09	26,0,0	0.26,0,0
5	21 x 18 x 9	0.21 x 0.18 x 0.09	47,0,0	0.47,0,0
6	21 x 18 x 9	0.21 x 0.18 x 0.09	68,0,0	0.68,0,0
7	26 x 17 x15	0.26 x 0.17 x 0.15	0,0,14	0,0,0.14
8	26 x 17 x15	0.26 x 0.17 x 0.15	78,0,21	0.78,0,0.21
9	26 x 17 x14	0.26 x 0.17 x 0.14	26,0,9	0.26,0,0.09
10	26 x 17 x14	0.26 x 0.17 x 0.14	52,0,9	0.52,0,0.09
11	26 x 18 x14	0.26 x 0.18 x 0.14	0,0,0	0,0,0

5. Conclusion and Future Development

5.1. Conclusion

This paper proposed a new approach in packing and loading process in the industry. The proposed system utilizes Augmented Reality (AR) technology to improve the interaction and reduce loading operation time for field workers. In the other hand, the utilization of depth camera gives possibility of automated data collection to reduce the time consumption when preparing the data manually.

The proposed integration model is designed for integrating depth camera and Augmented Reality (AR) to communicate. The model architecture is also designed using the newest technology and ready to implement in real application.

The advantages of using this new approach are as follows:

- 1) Modularity

The new approach model is designed as modular. The solution shown in [Figure 3-2] has been designed in separate module so that it does not affect each other when developer modifies one of the module.

- 2) Flexibility

This new approach can be deployed in any type of applications that need space utilization solution. It can be used also in any type of

devices: Desktop PC, mobile devices or Head-mounted Display.

3) Integration and Interoperability

Depth camera and other end-user devices are impossible to communicate each other directly. Because they are built in different platform and hardware. This research has proposed the integration model that is able to make all different technology communicate each other.

4) Real-Time Solution

The system integrity allow the user preview the solution whenever and wherever he wants. The delay of waiting previous process to be committed is omitted in the new approach. The automatic process of box data collection, lets the information is collected in real-time. Without waiting all the boxes are arrived, the worker or manager can see the packing solution whenever they want since the data is ready inside the database system.

5) Interactivity

Applying augmented reality in the new approach has given a broad capability to interact with the boxes and container. User can explore the packing solution and see it directly inside the physical container. The usage of augmented reality is to store the loading instruction in digital form. This digital form of boxes can assist the field worker to pack the boxes according to the instruction.

6) Practical Advantages

This prototype can replace one or two roles in the real process. There is no need to maintain the data since it can be done automatically. There is no need to wait for field manager to decide because the application itself can give the same decision. Moreover, the application can be used anywhere. The possible case is when manager wants to see the condition of cargo container loading operation in the field. He can run the application to simulate the packing from his office.

7) Smart Packing Agent

In the future, this prototype can be the smart robot packing. Let the camera become the robot's eyes that can read the dimension of boxes to identify which box to pack. Using the dimension and location coordinate information, the robot can pick the box and locate it in the right place.

8) Real-Time Resource Awareness

Another advantages of using camera as the system's vision is the camera can detect the surrounding environment in real-time. Sometimes, the container situation might not be empty. If several objects were exist inside and the system has not record it yet, the application can recognize the situation and adjust the solution according to the current situation.

5.2. Future Development

The current stage of the research is showing the whole system concept of the future packing process application. This research is a first step of a bigger idea in the future. Despite of many limitations in the current stage development, this research has a wide potential of development and improvement. The improvement can be divided into three parts as described in following sub section.

5.2.1. Data Collection Improvement

The camera has capability to recognize the shape and size of objects as what human see. The idea of using depth camera in the solution is to change the manual data collection to automated data collection one. However, this research only shows the idea of automated dimension measurement of the boxes using camera. The limitation of this measurement method can be improved by adding algorithm of tracking and segmentation to the objects. In the future, the dimension measurement can be done automatically.

The second improvement is adding validation before sending the dimension data to database. Currently, the validation is done manual. When the automated data collection is applied completely, the validation and checking method has to be improved.

5.2.2. Packing Algorithm Improvement

Current packing algorithm that is used in the research is the general packing

optimization algorithm. The heuristic algorithm is developed for simple model in general packing problem without special constraints. This new approach can be applied in any packing application in the industry. In order to match with the practical requirement, current algorithm can be modified according to the scenario.

5.2.3. AR Visualization Improvement

Since the main idea is improving the interaction of the end-user application, there are a lot of improvement can be developed in the future in the AR visualization part.

First improvement is improving the marker tracking and detection methodology. Currently the marker used in the research is image-based marker. In the future, it can be changed into text-based marker, because it can improve the container size detection also when applied. In the practice, the container can have various sizes. By detecting those text markers, it can give variable input of the container size in the packing algorithm.

Second improvement is improving the interaction part. Some scenario might have existing boxes placed inside the container. The AR camera is able to detect the container's environment in real-time. By collecting those information, it can help the system creates packing decision in real-time.

Third improvement is integrating depth camera and AR application using real-time connection other than database. Current stage is hard to realize because of Unity license.

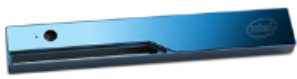

In summary, in the future, this research has potential to develop a smart agent to pack the boxes inside the container. By using depth camera to detect the size and identify the identity of each boxes, the smart agent can take, move and pack the boxes according to the optimized algorithm.

APPENDIX A MEASUREMENT

A.1 Depth Camera Specification

This experiment is utilizing depth camera from Intel named Real Sense Camera. The camera version that is used is R200, which is long-range camera [Intel, 2016a]. Microsoft Kinect is the most popular consumer level and cheap level camera in the market. Intel came with new camera called Real Sense Camera. Intel divides the camera into two cameras: short-range camera and long-range camera. The example of short-range cameras is SR300 [Intel, 2016b] series.

[Table A-1] Real Sense and Kinect Specification

		
Component	Real Sense R200	Kinect v2
Color Resolution	1920 x 1080 (2M)	1920 x 1080
Depth Resolution	640 x 480 (VGA)	512 x 424
USB	3.0	3.0
Frame Rate	30 fps	30 fps
Data Ranges	51 – 400 cm	40 – 450 cm

[Table A-1] shows the comparison between the popular depth camera from Microsoft, named Kinect and the camera that is used in this research, Intel Real Sense R200.

A.2 Measure Distance

```
case EPFeatures.MeasureDistance:
    SetStatus("Measure Distance Selected. Processing...");
    SelectedItem = EPFeatures.None; reset = false;
    PXCMEhancedPhoto.Measurement.MeasureData measureData;
    photoMeasure.MeasureUADistance(photo, Start, End, out
measureData);
    SetStatus("Distance between (" + Start.x + "," + Start.y + ")
- (" + End.x + "," + End.y + ") = " + (measureData.distance /
10).ToString("0.00") + "cm, Confidence =" +
measureData.confidence.ToString("0.00") + ", Precision =" +
measureData.precision.ToString("0.00") + "mm");

    MainForm.measureResult =
System.Math.Round((measureData.distance/10),2);

    int sz = photoMeasure.QueryUADataSize();
    PXCMEhancedPhoto.Measurement.MeasureData[] measureDataArr;
    photoMeasure.QueryUAData(out measureDataArr);
```

A.3 Database Connector

```
private void insertNew_Click(object sender, EventArgs e)
{
    string dimchoice = "";
    if (widthBtn.Checked) dimchoice = "dim1";
    if (depthBtn.Checked) dimchoice = "dim2";
    if (heightBtn.Checked) dimchoice = "dim3";

    string insertQuery = "INSERT INTO your table here (" +
dimchoice + ") VALUES(" + measureResult + ")";

    try
    {
        //Create connection string
        string connString = server, database, username, password;
        MySqlConnection cnn = new MySqlConnection(connString);
        MySqlDataReader myreader;
        cnn.Open();

        MySqlCommand insertDim = new MySqlCommand(insertQuery,
cnn);

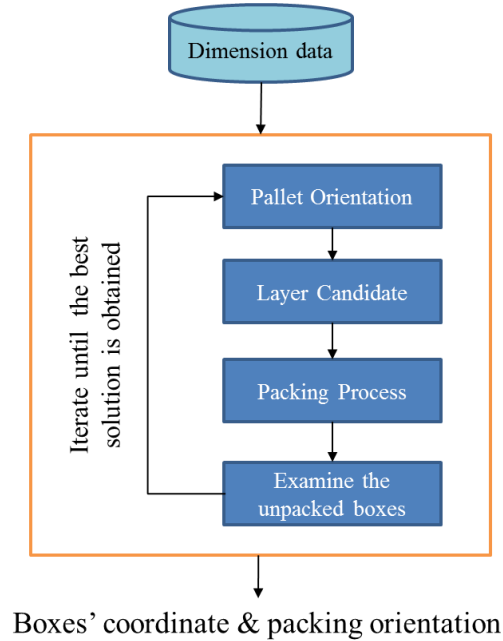
        myreader = insertDim.ExecuteReader();

        while (myreader.Read())
```

```
{  
    }  
    cnn.Close();  
    MessageBox.Show("New Data Added");  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
    MessageBox.Show("Adding Failed");  
}  
}
```

APPENDIX B PACKING

A.1 Packing Algorithm Flowchart



A.2 Packing Pseudo Code

Initialize

Execute Iteration

 List all possible layer height by giving weight

 Pack boxes found and record the information (box dimension)

 Find the most proper layer height by looking at the remaining space

 Find the most proper boxes

 Analyze each unpacked box to find the best fitting one to the empty

space

 Find the first to be packed gap in the layer edge

Examine candidate boxes and condition of the layer

Check the final condition and final volume utilization

Report

A.3 Marshalling Code

Marshalling source code in the C code as the gate input:

```
extern "C"
{
    __declspec(dllexport) void DoBinPacking(int pXX, int pYY, int pZZ,
    int boxn, int dim1[], int dim2[], int dim3[])
    {
        short int n;

        xx = pXX;
        yy = pYY;
        zz = pZZ;

        total_boxes = boxn;
        //int reportbox[];

        for (int i = 1; i <= boxn; i++) //get the length of boxdata
        {
            boxlist[i].dim1 = dim1[i-1];
            boxlist[i].dim2 = dim2[i-1];
            boxlist[i].dim3 = dim3[i-1];

            boxlist[i].vol = boxlist[i].dim1 * boxlist[i].dim2 *
boxlist[i].dim3;
        }

        initialize();
        time(&start);
        execute_iterations();
        time(&finish);
        report_results();

    }
}
```

Calling the C code from C# code:

```
class Program
{
    [DllImport("boxologic.dll")]
```

```

    public static extern void DoBinPacking(int pXX, int pYY, int
pZZ, int boxn, int[] dim1, int[] dim2, int[] dim3);

    [HandleProcessCorruptedStateExceptions]
    [SecurityCritical]
    static void Main(string[] args)
    {
        Console.WriteLine("This is C# Program");

        try
        {
            DoBinPacking(palletx, pallety, palletz, totalbox, size1,
size2, size3);
            Console.ReadKey();

        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
        Console.ReadKey();
    }
}

```

APPENDIX C VISUALIZATION

Full video demo can be found in the following link:

<https://youtu.be/74AmLibZptw>

C.1 Connection PHP Script

```
<?php
    $servername = "localhost";
    $username = "";
    $password = "";
    $dbName = "";

//Make connection
    $conn = new mysqli($servername, $username, $password, $dbName);

//Check connection
    if (!$conn){
        die("Connection failed.");
    }

//database
    $sql = "SELECT boxID, dim1, dim2, dim3 FROM box";
    $sql = "SELECT fkBoxID, rotatedDim1, rotatedDim2, rotatedDim3, coorX,
    coorY, coorZ FROM boxlist WHERE packedStat = 1";
    $result = mysqli_query($conn,$sql);

    if(mysqli_num_rows($result) > 0){
        //show data for each row
        while ($row = mysqli_fetch_assoc($result)){
            echo "ID:".$row['fkBoxID'] . " |Dimension 1:
".$row['rotatedDim1'] . " |Dimension 2: ".$row['rotatedDim2'] . " |Dimension 3:
".$row['rotatedDim3'] . " |X: ".$row['coorX'] . " |Y: ".$row['coorY'] . " |Z:
".$row['coorZ'] . " ";
        }
    }
?>
```

C.2 Data Conversion (DBHandler.cs)

```
using UnityEngine;
using System.Collections;
using System;

public class DBHandler : MonoBehaviour {

    private string[] boxes;
    private Vector3[] boxSize;
    private Vector3[] boxLocation;
    public ManageBox dbBox;

    // Use this for initialization
    IEnumerator Start () {
        //boxes data
        WWW boxesData = new WWW("fill with your webserver
address");
        yield return boxesData;
        string boxesDataString = boxesData.text;
        print(boxesDataString);
        boxes = boxesDataString.Split(';');

        dbBox.boxName = GetBoxID();
        dbBox.boxScale = GetPackedBoxScale();
        dbBox.boxPosition = GetPackedBoxPosition();

    }

    string GetDataValue(string data, string index)
    {
        string value = data.Substring(data.IndexOf(index) +
index.Length);
        if (value.Contains("|")) value =
value.Remove(value.IndexOf("|"));
        return value;
    }

    public Vector3[] GetPackedBoxScale()
    {
        Vector3[] boxScale = new Vector3[boxes.Length - 1];
        for (int i = 0; i < (boxes.Length - 1); i++)
        {
            var boxDim1 =
System.Convert.ToDecimal(GetDataValue(boxes[i], "Dimension 1:"));
            var boxDim2 =
System.Convert.ToDecimal(GetDataValue(boxes[i], "Dimension 2:"));
            var boxDim3 =
System.Convert.ToDecimal(GetDataValue(boxes[i], "Dimension 3:"));
            var boxDimension = new Vector3((float)boxDim1/100,
(float)boxDim2/100, (float)boxDim3/100);
        }
    }
}
```



```

        boxScale[i] = boxDimension;
    }
    return boxScale;
}

public Vector3[] GetPackedBoxPosition()
{
    Vector3[] boxPosition = new Vector3[boxes.Length - 1];
    for (int i = 0; i < (boxes.Length - 1); i++)
    {
        var posX =
System.Convert.ToDecimal(GetDataValue(boxes[i], "X:"));
        var posY =
System.Convert.ToDecimal(GetDataValue(boxes[i], "Y:"));
        var posZ =
System.Convert.ToDecimal(GetDataValue(boxes[i], "Z:"));
        var boxCoor = new Vector3((float)posX/100,
(float)posY/100, (float)posZ/100);
        boxPosition[i] = boxCoor;
    }
    return boxPosition;
}

public int[] GetBoxID()
{
    int[] boxID = new int[boxes.Length - 1];
    for (int i = 0; i < (boxes.Length - 1); i++)
    {
        boxID[i] =
Convert.ToInt16(GetDataValue(boxes[i], "ID:"));
    }
    return boxID;
}
}

```

C.3 Box Rendering (ManageBox.cs)

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class ManageBox : MonoBehaviour {

    //public GameObject kotak;
    public DBHandler dbconnect;
    public Transform tKotak;
    public Vector3[] boxScale;
    public Vector3[] boxPosition;
    public int[] boxName;

    List<GameObject> listKotak;

    void Awake()
    {
        listKotak = new List<GameObject>();
    }

    // Use this for initialization
    void Start()
    {
        StartCoroutine(GenerateBox());
    }

    private IEnumerator GenerateBox()
    {
        yield return boxScale;
        yield return boxPosition;
        yield return boxName;
        Debug.Log("Fungsi generate box");

        if (tKotak != null)
        {
            for (int i = 0; i < boxScale.Length; i++)
            {
                Debug.Log(boxScale.Length);
                Transform t = Instantiate(tKotak) as Transform;
                t.parent = transform;
                t.name = "Box " + boxName[i];
                t.localScale = boxScale[i];
                t.localPosition = boxPosition[i];
                Debug.Log(boxScale[i]);
                t.gameObject.SetActive(false);
                listKotak.Add(t.gameObject);
                Debug.Log(boxPosition[i]);
            }
        }
    }
}
```

```

        }
    }
}

public void CallBox()
{
    foreach (var kotak in listKotak)
    {
        kotak.gameObject.SetActive(true);
    }
}

public void DisableBox()
{
    foreach (var kotak in listKotak)
    {
        kotak.gameObject.SetActive(false);
    }
}
}

```

C.4 Manage AR Object Behaviour (ARHandler.cs)

```
using UnityEngine;
using System.Collections;
using Vuforia;

public class ARHandler : MonoBehaviour, ITrackableEventHandler
{
    private TrackableBehaviour mTrackableBehaviour;
    public ManageBox box;

    void Start () {
        Debug.Log("mulai");

        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
        {
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
        }
        OnTrackingLost();
    }

    public void OnTrackableStateChanged(TrackableBehaviour.Status
previousStatus, TrackableBehaviour.Status newStatus)
    {
        Debug.Log("this is state change");
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus ==
TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            OnTrackingFound();
        }
        else
        {
            OnTrackingLost();
        }
    }

    private void OnTrackingFound()
    {
        Debug.Log("track ketemu");
        box.CallBox();
    }

    private void OnTrackingLost()
    {
        Debug.Log("track hilang");
        box.DisableBox();}
}
```

C.5 Interaction Code

Detecting finger touch interaction is not an easy task to do in programming. Fortunately, there is open source library that helps programmer to add interaction without developing the code from scratch [Wilkes, 2015]. Code snippet that represent the object transformation is described in following subsection.

C.5.1 Translate

```
private void Translate(Vector2 screenDelta)
{
    // If camera is null, try and get the main camera, return true if
    a camera was found
    if (LeanTouch.GetCamera(ref Camera) == true)
    {
        // Screen position of the transform
        var screenPosition =
        Camera.WorldToScreenPoint(transform.position);

        // Add the deltaPosition
        screenPosition += (Vector3)screenDelta;

        // Convert back to world space
        transform.position =
        Camera.ScreenToWorldPoint(screenPosition);
    }
}
```

C.5.2 Scale

```
private void Scale(float scale, Vector2 screenCenter)
{
    // Make sure the scale is valid
    if (scale > 0.0f)
    {
        if (Relative == true)
        {
            // If camera is null, try and get the main camera, return
            true if a camera was found
            if (LeanTouch.GetCamera(ref Camera) == true)
            {
```

```

        // Screen position of the transform
        var screenPosition =
Camera.WorldToScreenPoint(transform.position);

        // Push the screen position away from the reference
point based on the scale
        screenPosition.x = screenCenter.x + (screenPosition.x -
screenCenter.x) * scale;
        screenPosition.y = screenCenter.y + (screenPosition.y -
screenCenter.y) * scale;

        // Convert back to world space
transform.position = Camera.ScreenToWorldPoint(screenPosition);

// Grow the local scale by scale
transform.localScale *= scale;
    }
}
else
{
// Grow the local scale by scale
transform.localScale *= scale;
    }
}
}

```

C.5.3 Rotate

```

private void Rotate(Vector3 center, float degrees)
{
    if (Relative == true)
    {
        // If camera is null, try and get the main camera, return
true if a camera was found
        if (LeanTouch.GetCamera(ref Camera) == true)
        {
            // World position of the reference point
            var worldReferencePoint =
Camera.ScreenToWorldPoint(center);
            // Rotate the transform around the world reference point
            transform.RotateAround(worldReferencePoint,
Camera.transform.forward, degrees);
        }
    }
    else
    {
        transform.rotation *= Quaternion.AngleAxis(degrees,
RotateAxis);
    }
}

```

C.5.4 Select

```
public void Select(LeanFinger finger)
{
    // Stores the component we hit (Collider or Collider2D)
    var component = default(Component);

    switch (SelectUsing)
    {
        case SelectType.Raycast3D:
        {
            // Get ray for finger
            var ray = finger.GetRay();

            // Stores the raycast hit info
            var hit = default(RaycastHit);

            // Was this finger pressed down on a collider?
            if (Physics.Raycast(ray, out hit,
float.PositiveInfinity, LayerMask) == true)
            {
                component = hit.collider;
            }
            break;

            case SelectType.Overlap2D:
            {
                // Find the position under the current finger
                var point = finger.GetWorldPosition(1.0f);

                // Find the collider at this position
                component = Physics2D.OverlapPoint(point,
LayerMask);
            }
            break;
        }

        // Select the component
        Select(finger, component);
    }
}
```

Bibliography

- Ahn, S. J. (2004). *Least squares orthogonal distance fitting of curves and surfaces in space* (Vol. 3151): Springer Science & Business Media.
- Azuma, R. T. (1997). *A survey of augmented reality. Presence: Teleoperators and virtual environments*, 6(4), 355-385.
- Ballem, B. P. (2000). *The distributor's three-dimensional pallet-packing problem: a mathematical formulation and heuristic solution approach*: DTIC Document.
- Baltacioglu, E., Moore, J. T., & Hill Jr, R. R. (2006). *The distributor's three-dimensional pallet-packing problem: a human intelligence-based heuristic approach. International Journal of Operational Research*, 1(3), 249-266.
- Bortfeldt, A., & Wäscher, G. (2013). *Constraints in container loading—A state-of-the-art review. European Journal of Operational Research*, 229(1), 1-20.
- Burke, E. K., Kendall, G., & Whitwell, G. (2004). *A new placement heuristic for the orthogonal stock-cutting problem. Operations Research*, 52(4), 655-671.
- Caudell, T. P., & Mizell, D. W. (1992). *Augmented reality: An application of heads-up display technology to manual manufacturing processes*. Paper presented at the System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on.
- CIA. (2017, 2017 June 07). *Indonesia*. Retrieved 2017 June 14, from <https://www.cia.gov/library/publications/the-world-factbook/geos/id.html>
- Craig, A. B. (2013). *Understanding augmented reality: Concepts and applications*: Newnes.

- Curless, B. (1999). *Overview of active vision techniques. SIGGRAPH 99 Course on 3D Photography.*
- Diedrich, F., Harren, R., Jansen, K., Thöle, R., & Thomas, H. (2008). *Approximation algorithms for 3D orthogonal knapsack. Journal of Computer Science and Technology, 23(5), 749-762.*
- EBSDocumentary. (2013, 2017 June 14). 극한직업 - Extreme JOB_택배물류센터 1부_#001. from https://www.youtube.com/watch?v=k77K2IY_scU
- Feng, X., Moon, I., & Shin, J. (2015). *Hybrid genetic algorithms for the three-dimensional multiple container packing problem. Flexible Services and Manufacturing Journal, 27(2-3), 451-477.*
- Galvan-Bobadilla, I., Ayala-Garcia, A., Rodriguez-Gallegos, E., & Arroyo-Figueroa, G. (2013). *Virtual reality training system for the maintenance of underground lines in power distribution system.* Paper presented at the Innovative Computing Technology (INTECH), 2013 Third International Conference on.
- Glockner, H., Jannek, K., Mahn, J., & Theis, B. (2014). *Augmented Reality in Logistics.* Online: http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/csi_augmented_reality_report_290414.pdf visited on 14th of August, 7, 45.
- Hartson, R., & Pyla, P. S. (2012). *The UX Book: Process and guidelines for ensuring a quality user experience:* Elsevier.
- Henrysson, A., Marshall, J., & Billinghurst, M. (2007). *Experiments in 3D interaction for mobile phone AR.* Paper presented at the Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia.
- Huang, Y.-H., Hwang, F., & Lu, H.-C. (2016). *An effective placement method for the single container loading problem. Computers & Industrial Engineering, 97, 212-221.*

- Intel. *Intel® RealSense™ SDK 2016 R2 Documentation: Measurement*. Retrieved 07/11/2016, from https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_epv_measurement.html
- Intel. (2016a). *Intel® RealSense™ Camera R200 Product Datasheet*. from <https://software.intel.com/sites/default/files/managed/d7/a9/realsense-camera-r200-product-datasheet.pdf>
- Intel. (2016b). *Intel® RealSense™ Camera SR300*. Retrieved 2017 June 15, from <https://software.intel.com/sites/default/files/managed/0c/ec/realsense-sr300-product-datasheet-rev-1-0.pdf>
- Kappauf, J., Lauterbach, B., & Koch, M. (2012). *Logistic Core Operations with SAP: Inventory Management, Warehousing, Transportation, and Compliance*: Springer Science & Business Media.
- Lim, A., & Zhang, X. (2005). *The container loading problem*. Paper presented at the Proceedings of the 2005 ACM symposium on Applied computing.
- Maguire, M. (2001). *Methods to support human-centred design*. *International journal of human-computer studies*, 55(4), 587-634.
- Martello, S., Pisinger, D., & Vigo, D. (2000). *The three-dimensional bin packing problem*. *Operations Research*, 48(2), 256-267.
- Mutto, C. D., Zanuttigh, P., & Cortelazzo, G. M. (2012). *Time-of-flight cameras and microsoft kinect (TM)*: Springer Publishing Company, Incorporated.
- Piumsomboon, T. (2015). *Natural hand interaction for augmented reality*.
- PTC. *Natural Features and Image Ratings*. Retrieved 2017 May 16, from <https://library.vuforia.com/articles/Solution/Natural->

Features-and-Ratings

- Schwerdtfeger, B. (2010). *Pick-by-vision: Bringing hmd-based augmented reality into the warehouse*. München, Techn. Univ., Diss., 2010.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*: Springer Science & Business Media.
- Trobina, M. (1995). *Error model of a coded-light range sensor*. Technical report.
- Wang, W., & Li, C. (2014). *Size estimation of sweet onions using consumer-grade RGB-depth sensor*. *Journal of Food Engineering*, 142, 153-162.
- Wheeler, A. (2016). *Understanding Virtual Reality Headsets*. Retrieved 2017 May 24, from <http://www.engineering.com/Hardware/ArticleID/12699/Understanding-Virtual-Reality-Headsets.aspx>
- Wilkes, C. (2015, 2017 June 15). *Lean Touch*. Retrieved 2017 April 06, from <https://www.assetstore.unity3d.com/kr#!/content/30111>

초록

박스 패킹 및 적재 작업은 물류 서비스에서 매우 빈번하고 중요한 과정이다. 뿐만 아니라 박스 크기 데이터 입력 프로세스는 효율적인 적재를 위해 제대로 수행되어야 할 과제이다. 입력 데이터는 패킹 및 적재 잡업을 계획 할 시 매우 중요한 역할을 한다. 현재, 기업들은 최적화 패킹을 위하여 패킹 소프트웨어를 사용하고 있다. 그러나 그것들을 실행하기 전에 올바르게 적절한 데이터를 준비하고 사용해야 한다. 또한 패킹 소프트웨어에서 도출된 최적화 솔루션은 종이 나 디지털 디스플레이에 기반한 솔루션이기 때문에, 실제 현장에서는 이해하기 어려움이 있다. 그래서 실제 현장 적용을 위해 또 다른 솔루션의 시각화 및 전달 방법이 필요하다.

본 연구에서는 데이터 수집과 솔루션 전달 및 시각화 문제를 해결하기 위해 새로운 접근 방식을 제안했다. Depth Camera와 증강 현실 기술(AR)이 통합된 실시간 접근 방식을 제안했다. 앞으로 이 방식은 자동적으로 패킹 및 적재할 수 있는 스마트 에이전트 분야에 응용될 것이라 기대된다.

주요어 : 3D 패킹 및 적재 작업, Depth Camera, 증강현실 (AR), 통합 시스템, 가상물리시스템, 물류 시스템, 산업 응용

학번 : 2015 - 23295