



**Sérgio Filipe
Gonçalves Vieira**

**Grupagem de Objetos em Espaços Limitados
Object Grouping in Limited Spaces**



**Sérgio Filipe
Gonçalves Vieira**

**Grupagem de Objetos em Espaços Limitados
Object Grouping in Limited Spaces**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor A. Manuel de Oliveira Duarte, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Professor Doutor Pedro Fonseca, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dissertation submitted to the University of Aveiro for compliance with the requirements for the degree of Master in Electronic and Telecommunications Engineering, held under the scientific guidance of Prof. Dr. A. Manuel de Oliveira Duarte, Full Professor at the Department of Electronics, Telecommunications and Informatics of the University of Aveiro, and Prof. Dr. Pedro Fonseca, Auxiliary Professor at the Department of Electronics, Telecommunications and Informatics of the University of Aveiro.

بِسْمِ اللَّهِ

In the name of God

Júri / Jury

Presidente / President

António Rui de Oliveira e Silva Borges
Professor Associado da Universidade de Aveiro
Associate Professor at the University of Aveiro

Vogais / Members

Jorge Manuel Pinho de Sousa
Professor Catedrático da Universidade do Porto
Full Professor at the University of Porto

A. Manuel de Oliveira Duarte
Professor Catedrático da Universidade de Aveiro
Full Professor at the University of Aveiro

Agradecimentos / Acknowledgments

Em primeiro lugar, gostava de agradecer aos meus orientadores: o Professor A. Manuel de Oliveira Duarte e o Professor Pedro Fonseca, pelos seus conselhos e sugestões feitos ao longo do desenvolvimento deste trabalho. Quero também agradecer ao meu colega de investigação Eng. António Alves pelo seu contínuo apoio ao longo dos últimos anos em que desenvolvemos vários projetos relacionados com soluções logísticas.

Estou especialmente agradecido ao meu amigo Jesse Brandão que me aconselhou e apoiou inúmeras vezes, ajudando a modelar a pessoa que sou hoje, assim como a sua esposa Asmaa Bahlaouane e família por serem incríveis anfitriões e dos quais aprendi importantes lições.

Quero também agradecer ao meu amigo Tekin Bursal cuja bondade nunca poderá ser retribuída, assim como ao meu amigo Karel Hauk cuja personalidade sofisticada causou um grande impacto na minha forma de ver as coisas.

Por último, mas não menos importante, estou em dívida para com a minha família: o meu pai Albino Vieira, a minha mãe Elisabete Barbosa, o meu avô Albino Vieira e o meu irmão Emanuel Vieira por todos os seus aconselhamentos, apoio, e sacrifícios feitos para que eu pudesse atingir este marco na minha vida. A todos vocês: o meu muito obrigado.

First of all, I would like to thank my academic advisors: Prof. A. Manuel de Oliveira Duarte and Prof. Pedro Fonseca, for their advice and suggestions made during the development of this document. I would also like to thank my research colleague Eng. António Alves for his continuous support through these last years we have been working together on the various projects related with logistic solutions.

I am especially thankful to my friend Jesse Brandão, who has given me countless advice and support, helping shape the way I am today, as well as his wife Asmaa Bahlaouane and her family for being amazing hosts and from whom I have learned many important lessons.

I would also like to thank my friend Tekin Bursal and his family whose kindness can't ever be repaid, as well as my friend Karel Hauk whose sophisticated personality has made a huge impact on my way of seeing things.

Last but not least, I am indebted to my family: my father Albino Vieira, my mother Elisabete Barbosa, my grandfather Albino Vieira and my brother Emanuel Vieira for all their counseling, support and sacrifices made so that I could achieve this milestone in my life.

To all of you: thank you.

Palavras-chave

Logística, Carregamento de Mercadoria, Posicionamento 3D de Volumes, Varrimento por Laser, Visão estereoscópica por Computador

Resumo

Com a crescente necessidade de transporte de mercadoria como resultado da globalização económica, é importante o melhoramento dos processos e procedimentos das operações logísticas como o carregamento e descarregamento de mercadoria, por forma às corporações aumentarem a sua vantagem competitiva e rentabilidade.

Esta dissertação explora e apresenta dois temas relacionados com processos logísticos: Posicionamento de volumes e tecnologias de deteção para monitorização de mercadorias. Uma heurística foi desenvolvida para atribuição e posicionamento 3D de volumes dentro de contentores seguindo uma estratégia de colocação que produz soluções verticalmente estáveis e com um alto grau de compactação. Desenvolveram-se dois protótipos usando tecnologia de deteção capaz de medição de volumes, varrimento por laser e visão estereoscópica por computador, como fundação para um sistema para monitorizar o carregamento e descarregamento de mercadorias. Estes sistemas pretendem providenciar assistência para os operadores logísticos na aplicação de normas e identificação de potenciais problemas.

Os resultados obtidos pela heurística são promissores, mostrando que um conjunto de simples regras de posicionamento de caixas consegue obter uma boa percentagem de ocupação de volume do contentor. Ambos os protótipos de deteção apresentaram bons resultados nos testes de laboratório, com o protótipo de visão estereoscópica provando ser o mais preciso e fiável para potencial desenvolvimento em uma aplicação a ser instalada em contentores.

Esta dissertação conclui com observações acerca de futuros melhoramentos e desenvolvimentos para o trabalho implementado.

Keywords

Logistics, Cargo Loading, 3D Volume Placement, Laser Rangefinder, Computer Stereo Vision

Abstract

With the ever necessity of cargo transportation as the results of economic globalization, it is important to improve the processes and procedures of logistic operations such as cargo loading and unloading, in order for corporations to increase their competitive advantage and profitability.

This dissertation explores and presents two subjects related with logistic processes: Volume placement and sensing technologies for cargo monitoring. A heuristic was developed for 3D volume assignment and placement inside containers following a positioning strategy that produces vertically stable solutions with a high degree of compactness. Two prototypes using sensing technology capable of volume measurement, laser range finding and computer stereo vision, were developed as a foundation for a system for monitoring the loading and unloading of cargo. These systems aim to provide assistance to logistic operators on the application of standards and identification of potential issues.

The obtained results on the heuristic are promising, showing that a simple set of rules for placement of boxes can achieve a good occupation percentage of the container's volume. Both sensing prototypes showed good results on lab tests with the stereo vision prototype proving to be the most accurate and reliable for potential further developments into an application to be installed in containers.

This dissertation concludes with remarks regarding future improvements and developments for the implemented work.

Table of Contents

1. Introduction.....	1
1.1. Framework.....	2
1.2. Motivation.....	2
1.3. Problem Statement.....	4
1.4. Objectives	4
1.5. Methodology.....	5
1.6. Structure.....	6
2. State of the Art.....	7
2.1. Literature Review	7
2.1.1. Sensing Methods	7
2.1.1.1. Laser Range Finding Approaches	8
2.1.1.2. Stereo Vision.....	10
2.1.1.3. Other Approaches	11
2.1.1.4. Volume Measurement and 3D Reconstruction Methods	13
2.1.1.5. Comparison of 3D ranging techniques.....	14
2.1.2. Container Loading Problem	15
2.2. Market Solutions.....	17
2.2.1. Range Finding for 3D Reconstruction	17
2.2.2. Container Loading and Planning.....	20
3. Concepts and Design	23
3.1. Cargo Planning	23
3.1.1. Considered Scenarios	24
3.1.2. Definitions and Geometric Model.....	26
3.1.2.1. Pallet Assembling	26
3.1.2.2. Cargo Loading.....	31
3.1.2.3. Data Structure	35
3.1.3. Logistic Importance (Cost Function)	37
3.1.4. Constraints	38
3.1.5. Dimensions and Loading Layout	39
3.1.6. 0/1 Knapsack Problem	40

3.2. Sensing Methods.....	42
3.2.1. Requirements.....	42
3.2.2. Laser Rangefinder	42
3.2.3. Computer Stereo Vision.....	47
4. Sensing Systems for Cargo Management	53
4.1. Laser Rangefinder Prototype	53
4.1.1. Assumptions.....	53
4.1.2. Software Implementation	53
4.2. Computer Stereo Vision Prototype	58
4.2.1. Assumptions.....	58
4.2.2. Software Implementation	59
5. Heuristic for Volume Loading	65
5.1. Assumptions	65
5.2. Implementation	65
5.2.1. Pallet Assembling	65
5.2.1.1. Tower and Layer Building	68
5.2.2. Cargo Loading.....	72
5.2.2.1. Weight Distribution.....	74
6. Results and Discussion	77
6.1. Heuristic.....	77
6.1.1. Data Structure Example	77
6.1.2. Pallet Assembling	80
6.1.3. Cargo Loading.....	86
6.2. Sensing Systems	93
6.2.1. Laser Rangefinder Prototype.....	93
6.2.2. Computer Stereo Vision Prototype	94
7. Conclusions and Future Work	97
7.1. Future Work.....	98
References	101
Annexes	107
A. Triangulation Error Graphs.....	107
A.i. Laser rangefinder	107

A.ii. Stereo Vision.....	110
B. 0/1 Knapsack Problem Algorithm	115
B.i. Application Example.....	115
C. Generated Data for the Heuristic and Examples.....	117
C.i. Pallet Assembling	117
C.i.1. Other examples	120
C.ii. Cargo Loading.....	121
D. Practical Case Study	125

Table of Figures

Figure 1.1: European logistics costs ratio and transport tonnages (Adapted from: [2]).....	3
Figure 1.2: Development cycle (modified waterfall model)	5
Figure 2.1: Schematic of a laser system with focal plane scanning (Source: [5]).....	8
Figure 2.2: Laser phase-shift rangefinder schematic (Source: [6])	9
Figure 2.3: Example geometry for triangulation with color information (Source: [5])	9
Figure 2.4: Stereo triangulation (Source: [8])	10
Figure 2.5: Structured light method (Source: [12])	11
Figure 2.6: Time-of-flight camera measurement principle (Source: [14]).....	12
Figure 2.7: Velodyne HDL-64E S3 (Source: [27])	17
Figure 2.8: Sick LMS200 (Source: [29]).....	18
Figure 2.9: Kinect 1 (Source: [32])	18
Figure 2.10: Kinect 2 (Source: [34])	19
Figure 2.11: Bumblebee2 (Source: [37]).....	19
Figure 2.12: ZED Stereo Camera (Source: [38]).....	19
Figure 2.13: Intel RealSense R200 (Source: [39])	20
Figure 3.1: General planning process	24
Figure 3.2: Pallet assembling and Cargo loading formulations	25
Figure 3.3: 3D problem representation.....	26
Figure 3.4: Box type	26
Figure 3.5: Container type	27
Figure 3.6: Box pose.....	28
Figure 3.7: Payload.....	29
Figure 3.8: Pallet	30
Figure 3.9: Cargo pose	32
Figure 3.10: Bundle	32
Figure 3.11: Cargo payload	33
Figure 3.12: Cargo container	34
Figure 3.13: Class diagram.....	36
Figure 3.14: Common “euro truck” layouts - (a) 33 pallets, (b) 33 pallets, (c) 32 pallets ..	40
Figure 3.15: The Knapsack Problem (Source: [51]).....	40

Figure 3.16: LRF system overview	43
Figure 3.17: LRF with triangulation (based on [53])	43
Figure 3.18: LRF error variation for values below 10% of the distance and $\Delta h = 0$	45
Figure 3.19: LRF error variation for values below 10% of the distance with $\Delta h \neq 0$	46
Figure 3.20: SV system overview	47
Figure 3.21: Stereo vision with two cameras (based on [8])	48
Figure 3.22: SV error variation for values below 10% of the distance and $\Delta B = 0$	49
Figure 3.23: SV error variation for values below 10% of the distance and $\Delta B \neq 0$	50
Figure 4.1: LRF system activity diagram	55
Figure 4.2: Grid Generation and Measurement Procedure	56
Figure 4.3: Original frame (left) and Frame with Regions of Interest (right)	56
Figure 4.4: LRF detection algorithm	57
Figure 4.5: Example output with input data (red dots)	58
Figure 4.6: Stereo System Activity Diagram	60
Figure 4.7: Frame with object (left) and background frame (right)	61
Figure 4.8: Result after subtraction and smoothing	61
Figure 4.9: After threshold (left) and after morphological operations (right) results	62
Figure 4.10: Detected object (green rectangle) and detected corners (red)	62
Figure 4.11: Geometric shape as viewed by the system	63
Figure 4.12: Stereo system output	64
Figure 5.1: Volume assignment implementation flow	66
Figure 5.2: Volume placement implementation	67
Figure 5.3: Placement strategy - 1	68
Figure 5.4: Placement strategy - 2	69
Figure 5.5: Placement strategy - 3	70
Figure 5.6: Placement strategy - 4	71
Figure 5.7: Placement strategy - 5	72
Figure 5.8: Bundle assignment implementation flow	73
Figure 5.9: Chessboard weight matrix	75
Figure 5.10: Chessboard weight matrix with matching cases	75
Figure 6.1: Box type	77
Figure 6.2: Box placement into a pallet	78

Figure 6.3: 2D cargo placement	79
Figure 6.4: 3D cargo placement	80
Figure 6.5: 1-box type	81
Figure 6.6: Packing pattern (1-box type).....	81
Figure 6.7: Packing pattern (full 1-box type)	82
Figure 6.8: 3-box types.....	83
Figure 6.9: Packing pattern (3-box types)	84
Figure 6.10: Packing pattern (30-box types)	85
Figure 6.11: 6-box types.....	86
Figure 6.12: Loading pattern and weight distribution	88
Figure 6.13: Longitudinal and transversal weight distribution (efficient)	89
Figure 6.14: 3D loading pattern (efficient).....	90
Figure 6.15: Packing pattern (pallet 31)	90
Figure 6.16: Loading pattern with uniform weight distribution.....	91
Figure 6.17: Longitudinal and transversal weight distribution (uniform)	91
Figure 6.18: 3D loading pattern (uniform)	92
Figure 0.1: LFR error variation - 1	107
Figure 0.2: LFR error variation - 2	108
Figure 0.3: LFR error variation - 3	109
Figure 0.4: LFR error variation - 4	110
Figure 0.5: SV error variation - 1	111
Figure 0.6: SV error variation - 2	112
Figure 0.7: SV error variation - 3	113
Figure 0.8: Example - 1	120
Figure 0.9: Example - 2	120
Figure 0.10: Example - 3	121
Figure 0.11: Assembled pallet from client 8	126
Figure 0.12: Loading pattern (efficient)	128
Figure 0.13: 3D loading pattern (efficient).....	129
Figure 0.14: Weight distribution (efficient)	129
Figure 0.15: Loading pattern (uniform).....	130
Figure 0.16: 3D Loading pattern (uniform).....	131

Figure 0.17: Weight distribution (uniform)..... 131

Table of Tables

Table 2.1: Classification of ranging methods (Source: [4])	7
Table 2.2: Comparison of ranging methods (Source: [4], slightly adapted)	14
Table 6.1: Box properties	77
Table 6.2: Placement list (payload)	78
Table 6.3: Bundle	78
Table 6.4: Layout placement list (cargo payload)	79
Table 6.5: Execution parameters (input data).....	81
Table 6.6: Box types characteristics	83
Table 6.7: Packing pattern statistics (3-box-types)	84
Table 6.8: Packing pattern statistic (30-box types)	85
Table 6.9: 6-box types characteristics	86
Table 6.10: Generated bundles	87
Table 6.11: Selected bundles	87
Table 6.12: Loading pattern output statistics	88
Table 6.13: LRF prototype results	93
Table 6.14: Stereo vision prototype results	95
Table B.1: Input values.....	116
Table B.2: 0/1 KSP results table.....	116
Table C.3: Generated input data	117
Table C.4: Boxes left out of the placement	118
Table C.5: Placement list generated by the heuristic.....	119
Table C.6: Statistics - 1	120
Table C.7: Statistics - 2	121
Table C.8: Statistics - 3	121
Table C.9: Bundle layout placement (loading priorities)	121
Table C.10: Bundle layout placement (uniform weight distribution)	123
Table D.11: Boxes per client	125
Table D.12: Bundles	127
Table D.13: Selected bundles (clients)	127
Table D.14: Selection statistics	128

Acronyms

0/1 Knapsack Problem	0/1 KSP
Container Loading Problem	CLP
Laser Rangefinder	LRF
Stereo Vision	SV

Glossary

box type	A geometric entity that defines a box.
container type	A geometric entity defining a container that holds smaller entities.
box	An instantiation of a box type with an associated weight.
box pose	Positional information of a box inside a container (pallet).
payload	A set of boxes and their poses.
pallet	An instantiation of a container type that holds a payload.
cargo pose	Positional information of a pallet inside a container (cargo container).
bundle	A group of pallets belonging to the same client.
cargo payload	A set of pallets (belonging to assigned bundles) and their cargo poses.
cargo container	An instantiation of a container type that holds a cargo payload.

Dictionary of Symbols

box type	g
container type	g^*
box	B
box pose	p
payload	ψ
pallet	P
cargo pose	p^*
bundle	D
cargo payload	ψ^*
cargo container	C

1. Introduction

It comes as no surprise that logistics has played an important role throughout history. From sailing the sea to distant lands, waging war or commercial transactions, good logistics was often a key factor in the success of the outcome. Cultural expressions such as “those who go to the sea, get ready on land” (Portuguese proverb) illustrate the impact that logistics had on shaping cultures and civilizations.

Nowadays, logistics plays a significant role in creating competitive advantage for business organizations, and it is by recognizing its importance as a differentiating factor in the market, that companies can provide a better value to their customers than their competitors, which in turn result in an increase of business profitability.

More specifically, logistics management is the process of planning the procurement, movement and storage of equipment, materials or supplies and its associated information through the company in a way that maximizes the current and future profitability by fulfilling cost-effective orders [1]. In other words, logistics management is the integrator and coordinator of all logistic activities (fleet management, inventory management, warehousing, order fulfillment, etc.) from the point of origin (raw supplies) to the point of consumption (customer) meeting the customer’s requirements.

Cases of successful logistics management include IKEA, a multinational designer of furniture, that specialized in ready-to-assemble furniture that is designed so that the customer can assemble the final product at home. This means that transportation costs can go lower since items take less space during transportation, which in turn allows for a higher volume of goods to be transported. Parallel to these operations are powerful information systems that manage the inventory of each warehouse through sale projections, making sure that stocks are refilled with the exact amount of supplies. This cuts on lost sales and at the same time makes sure that excessive items are not ordered from the suppliers.

Another example is Walmart, an American multinational retail corporation, that instituted cross-docking at their warehouses, i.e., inventory moves directly from an arriving truck to a departing one. This allows products to spend less time at the warehouse and instead be readily available at the stores, cutting on potential lost sales. All these operations rely on information systems with a centralized database, the information is then shared with

suppliers so that stock is easily refilled. Palletization also improves the process by allowing the cargo capacity of the trucks to be utilized more efficiently.

Therefore, as seen by these examples, increasing the effectiveness of logistics operations can lead to a better execution of activities by the company, which in turn, directly translates into an increase of profits.

1.1. Framework

A key element in logistics is the transportation system. Transportation is what connects the different activities and directly influences the whole performance of the logistic system such as having an impact on reducing the operation's cost and promoting the service quality.

Palletization has been used reduce transportation and warehousing costs by packing the cargo inside pallets (which are easier to handle) providing a better integration with the other logistic activities. However, cargo loading and space optimization (being a real time problem) is a task of higher complexity since cargo can have different dimensions and be of different types. This has led companies to rely on trial and error methods or rules of thumb, based on their cumulative practical years of experience. Methods like these, which are not standard or even not well defined, often require time to reallocate or replace cargo within the container space until it is organized in a secure and stable position. These kind of processes are of utmost importance to an industry which is heavily influenced by fuel prices, since the time spent on loading the cargo and its efficiency, dictates how many transports can be done per working day, directly impacting the overall costs. Therefore, the implementation of well-defined methods that solve this problem is of critical importance to transport carriers looking to improve their services.

1.2. Motivation

In Europe during 2014, logistics costs accounted for 960 billion euros where transportation costs occupy an astonishing 44%, with a total transport of 18.6 billion tons being done mainly by road (Figure 1.1) [2].

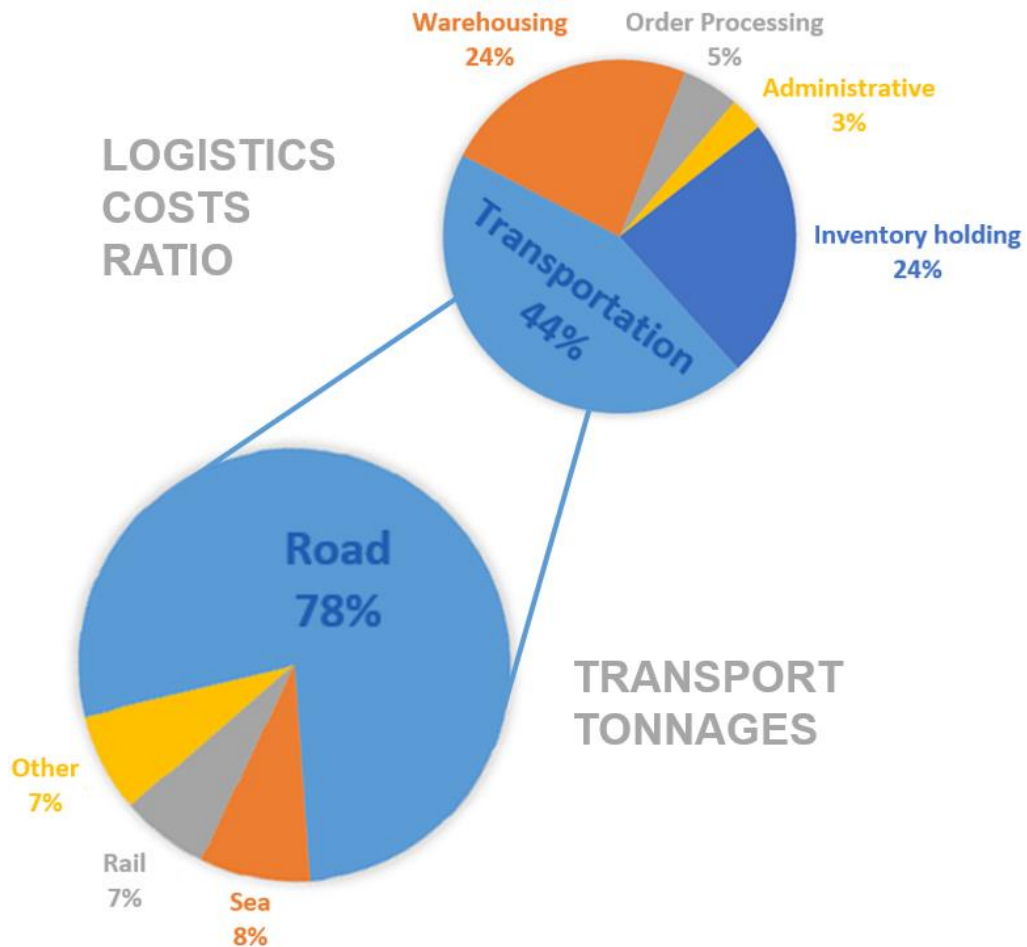


Figure 1.1: European logistics costs ratio and transport tonnages (Adapted from: [2])

Allied to these statistics is the increasing concern with carbon dioxide emissions. More than ever before, countries are required to reduce and control their pollution rates to develop and protect a sustainable environment. The broad public is more aware of these environmental concerns and a company that provides solutions with this in mind might capture that advantage.

On the other hand, these statistics highlight the importance of development and investment in transportation where the majority of the costs are concentrated. The improvement of processes executed in this field, more specifically the cargo loading and unloading process, can lead to the increase of efficiency and therefore reduce the costs involved.

1.3. Problem Statement

This dissertation focus on two subjects:

- Development of sensing methods to monitor a cargo container.
Logistic operators often require information on the available space inside a cargo container to better evaluate the efficiency of operations (cargo loading and transportation) or detect potential problems. Thus, sensing technology capable of measuring volumes and retrieve the cargo geometry can fulfill such task.
- Improve the loading and volume utilization of a cargo container.
Traditional cargo loading processes don't often provide the best usage of the container's volume. Therefore, a heuristic for cargo loading can play a role in improving the process.

Two systems designed for the measurement of volumes are presented and a heuristic is proposed to improve the process of loading/unloading a container.

1.4. Objectives

The central objective of this dissertation is to contribute to a better efficiency of cargo loading and unloading processes in transportation – crucial operations in the logistic chain – by resorting to the incorporation of information systems and sensing technology.

The specific objectives are the following:

- To formulate a 3D representation model for the loading of goods into a confined space as per the Container Loading Problem [3] definition.
- To develop a heuristic capable of solving a predefined subset of the Container Loading Problem and evaluate its results.
- To present the concept and design of 3D cargo volume representation techniques focusing on two frequently used approaches:
 - Stereo vision;
 - Laser range finding.

This work can be of relevance for logistic operators that seek a better efficiency for their operational processes by: on one hand, an improvement of the loading and unloading

of cargo from trucks or containers; on the other hand, an application of a measurement system that can monitor the cargo loading/unloading during critical processes, so that the operator has an extended control over their operations.

1.5. Methodology

Research was conducted on the available academic literature and market solutions in order to select the potential technologies and approaches that fit the necessary requirements.

Empirical and simulation approaches were used to experimentally acquire data to validate the feasibility of the designed solutions.

Development was performed using a sequential model (as shown in Figure 1.2).

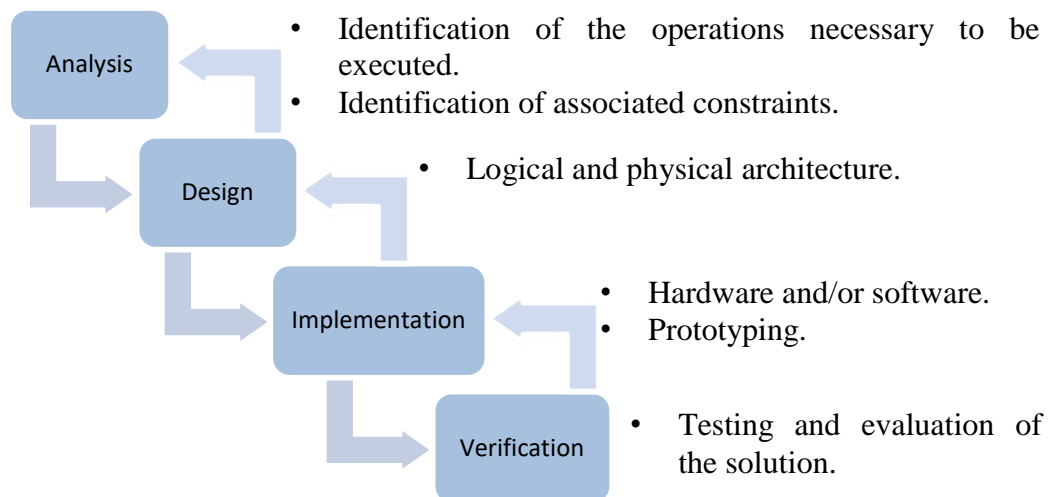


Figure 1.2: Development cycle (modified waterfall model)

This model provided a well-defined structure of steps for the development cycle, allowing to backtrack one phase if irregularities were detected, or additional features were required to be implemented.

During the design phase, simple models were constructed based on real environments and then simulated in order to test the application of the identified operations along with the associated constraints.

A prototyping approach was used during the implementation phase. Prototypes were build and improved through an iterative process of trial and error using the experimentally acquired data in controlled lab conditions, constructed to approximate real life scenarios.

Iterative prototyping allowed to detect problems and correct or minimize their effect, making the implementation more resilient at each step.

1.6. Structure

The organization of this dissertation goes as follows: in section 2, a state of the art review of the literature and available market solutions will be conducted focusing on describing the principles and characteristics of the solutions; section 3 describes the concepts and designing aspects of the implemented solutions; section 4 describes the software implementation of the developed volume measuring sensing systems; section 5 presents the implementation of an heuristic for cargo loading and volume assignment; section 6 presents and discusses the results obtained by the developed solutions; finally, section 7 concludes this dissertation and includes potential remarks for potential future work under the explored subject.

2. State of the Art

This section will present an overview of the principles behind the methods that provide measurement of volumes and 3D reconstruction. Heuristics and approaches available on volume assignment (solving the Container Loading Problem) are also reviewed. An overview of some available market sensors and cargo optimizing software (loading and packing) is also presented.

2.1. Literature Review

Relevant academic literature on sensing technology and volume assignment approaches were reviewed. Only literature written in English or Portuguese and publicly available or accessed through the University of Aveiro's library was considered.

2.1.1. Sensing Methods

There is a wide variety of 3D range finding methods. A possible classification for some of the most relevant in the industry, is proposed by Sansoni et al. [4] according to their measuring characteristics and is shown in Table 2.1.

Table 2.1: Classification of ranging methods (Source: [4])

	Triangulation	Time delay	Active	Passive
Laser triangulation	X		X	
Structured light	X		X	
Stereo Vision	X			X
Time-of-flight		X	X	
Interferometry		X	X	

Methods are classified as active if a controlled light source (such as a laser) is used as the method to retrieve the ranging information, and as passive if a direct light source is not used for the measurement procedure, i.e., the method infers the distance using indirect light sources (such as ambient light).

These methods are also considered direct, i.e., the output of their operation is raw range data (distances) with no extra information required about the target being measured.

2.1.1.1. Laser Range Finding Approaches

Distance measurement using laser methods can be divided into three categories: triangulation, time-of-flight and interferometry. Amann et al. [5] do a review of the common laser range finding techniques used in the industry.

Laser pulse time-of-flight distance measuring systems measures the round trip time of a light pulse emission (commonly a laser) and the return from the reflection upon hitting the target object which is then translated to a distance. An implementation of such a system uses a focal plane scanning (Figure 2.1) which allows to 3D map an entire surface (without mechanically drive the system to measure different points of the surface) by using a matrix of different detectors where each covers a small fraction of the field of view illuminated by the laser.

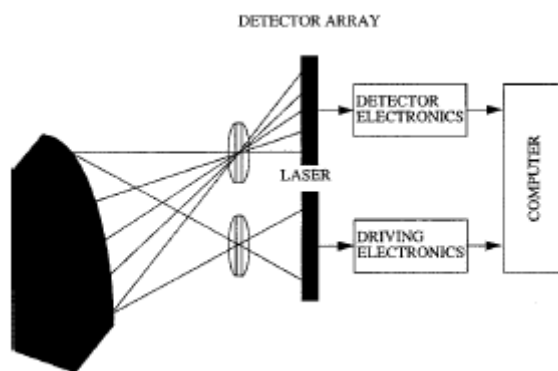


Figure 2.1: Schematic of a laser system with focal plane scanning (Source: [5])

Error in these systems can arise from noise sources, such as the electronic circuitry or background radiation, which induce time jitter which ultimately settles the maximum capable precision of the system. However, the advantage of a time-of-flight system comes from its high precision and the fact that it uses the same direct path from the target and back to the sensor, which allows for a more compact design when compared with other solutions such as the optical triangulation method.

Laser phase-shift range finding works by modulating the optical power (laser) with a constant reference frequency and then measure the difference in phase obtained by the received reflection from the target, which is translated into the distance. This method allows measurements of 1 millimeter of resolution or less.

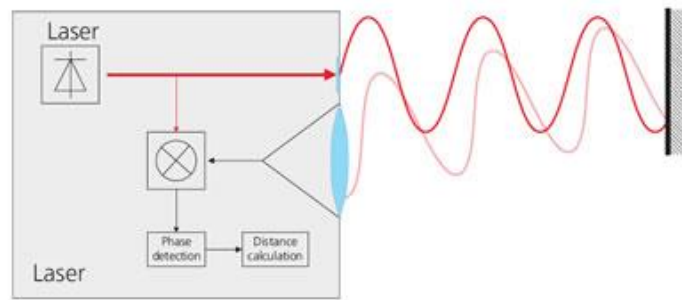


Figure 2.2: Laser phase-shift rangefinder schematic (Source: [6])

Electrical crosstalk between the transmitter and receiver is one of the major sources of error with these systems. However, this influence cannot be totally removed with conventional shielding techniques, which is why this is the factor that determines the accuracy of the system at a given modulation frequency.

Distance measuring with phase-shifting is, however, a high accurate method for ranges up to 20 meters. A photodiode with a large area can increase the field of view of the sensor, allowing the laser to be scanned by micro mirrors making the system more compact and potentially reduce its costs of implementation.

Optical triangulation is done using a setup of a light source (an emitter such as a laser), a photo sensor (receptor) and an appropriate geometry (as exemplified in Figure 2.3) where the receptor can detect the light reflected by the target off the light source in order to 3D map the object. Based on the position of the detected pixels and the geometry of the system, distances can be measured to target objects. Danko [7] also proposes a similar solution where in order to improve the pixel detection conversion to distances, experimental data is taken and fitted through a proper function.

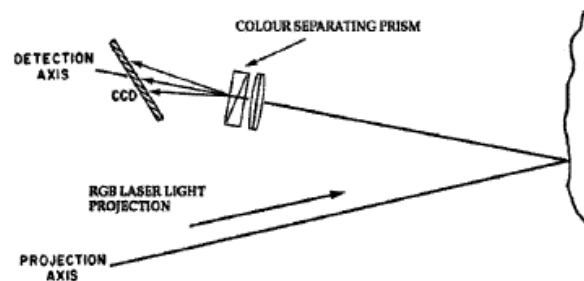


Figure 2.3: Example geometry for triangulation with color information (Source: [5])

The main source of error on this system is due to speckle noise. This phenomenon happens when diffusion occurs at the surface of the target object, which in turn results in the

scattering of the laser reflections. The system then needs to calculate the centroid of the detected pixels from the various reflections which results in uncertainty in the detected position. This method relies on a good geometric design in order to obtain the best resolution. It should also be noted that this method is low cost and easy to implement.

2.1.1.2. Stereo Vision

Stereo vision is a method used to create an illusion of depth by using two images taken from different positions and triangulating distances to targets, making it possible to reconstruct 3D environments. For this operation, the following condition must hold: pictures should be taken at the same instant, i.e., the target object/scenery must not move or change between shots.

Mrovlje and Vrančić [8] use two pictures from two horizontally aligned cameras to determine the distance between the camera set and a target object (Figure 2.4), by calculating the horizontal pixel disparity (of the object) between both pictures. The process begins by selecting an object to be identified and then the algorithm detects the same object on the second picture (searching for minimum pixel difference) to determine the pixel disparity. The distance is then calculated using triangulation, proving to be highly accurate. Mahammed et al. [9] use the same principle, however, to improve the distance calculation, experimental data is used to determine a fitting function that better relates the pixel disparity and distance to the target object (in the same way as proposed by Danko [7] for the triangulation using a laser). Both authors require absolute parallelism of the camera's optical axis.

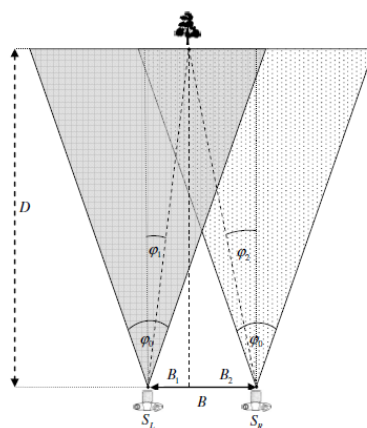


Figure 2.4: Stereo triangulation (Source: [8])

Masuda et al. [10] address the parallelism condition by converting the taken images into parallel images using image rectification. This process is done through calibration of the cameras using their intrinsic and extrinsic parameters. The distance is then calculated using the pixel disparity principle and triangulation. The same process of stereo rectification is employed by Miura et al. [11].

The biggest challenge in stereo vision is solving the correspondence problem, i.e., matching a feature in one picture to the same feature in the second picture. This requires using complex and computationally intensive algorithms to extract features and do the matching. Feature extraction also requires sufficient variation of intensity, otherwise potential features of interest might not be detected and correctly extracted for disparity calculation. Nonetheless, stereo vision provides highly accurate measurements and only requires 2 cameras which makes it physically easy to implement.

2.1.1.3. Other Approaches

Aside from the traditional depth measurement methods, there are other approaches such as the **structured light** method. Jia et al. [12] describe the method: a laser emits a known pattern (typically using infrared light) into a scene where the target object resides, a camera then captures the distortion pattern and correlates it against a reference pattern to infer the distance through triangulation (Figure 2.5).

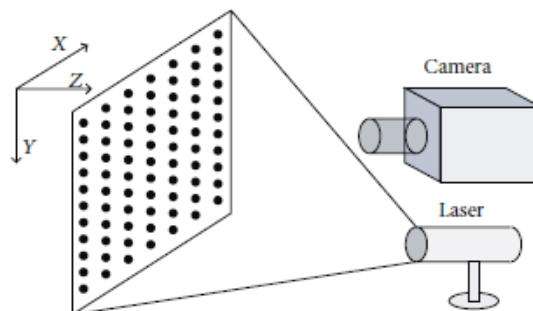


Figure 2.5: Structured light method (Source: [12])

Ambient light (or other external sources) and improper calibration of the system are the two main sources of error for this system. Even though the laser emits in the infrared spectrum, ambient light is a source of infrared noise that needs to be filtered. On the other hand, calibration errors are due to improper calculation of the camera intrinsic parameters

that are not always possible to properly determine. These sensors can typically only measure on a limited range.

Alhwarin et al. [13] propose an improvement to the method by combining two sensors in order to apply the stereoscopic principle. This works by calculating the disparity between the depth data obtained from both sensors (after stereo rectification), allowing for triangulation methods to determine objects whose properties made them hard to detect (such as transparent, reflective or absorptive materials).

Another relevant technology is **time-of-flight cameras**. Hansard et al. [14] describe the principle of depth measurement. An infrared wave is emitted to a target object and a sensor detects the reflected infrared component. The distance is determined by measuring the phase difference between the emitted signal and the reflected response.

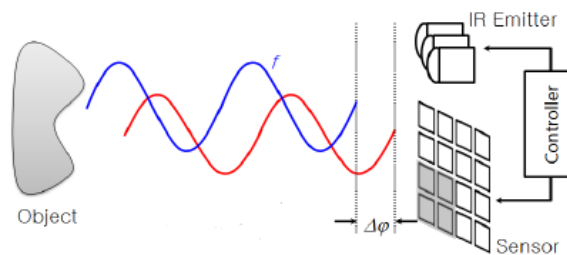


Figure 2.6: Time-of-flight camera measurement principle (Source: [14])

There are two methods to detect phase shifts as described by Li [15]:

- Pulsed modulation
- Continuous-wave modulation

In **pulsed modulation** the object is illuminated for a brief period of time and the reflected signal is sampled (at every pixel) by 2 out-of-phase samples. The accumulated electric charges during each of the 2 samples are then measured and used to determine the distance to the object.

For **continuous-wave modulation** the system takes 4 samples each one with a 90-degree phase delay from each other. The distance is then calculated from the phase angle between the illumination and reflection.

The measurement accuracy is limited by the emitter's power capability which might also suffer from added noise from ambient light. Another problem is motion blur for cases where either the target or the system is moving.

2.1.1.4. Volume Measurement and 3D Reconstruction Methods

From the distance points generated by the sensor system, volume and 3D reconstruction can be obtained through different methods. Zhang et al. [16] show two methods of simple implementation:

- **Meshed surface** is a technique that reconstructs the surface of the target object using interpolation methods such as the Delaunay triangulation. The volume is then calculated by summing the individual volumes of each projected triangle of the reconstructed surface to a defined plane.
- **Isolated points** method relies on the having a large amount of measurement points with no middle gaps (points where the measurement was inconclusive). Zhang et al. [16] showed that in this case each point can be regarded as the top of a truncated 3D pyramid, and the volume can be obtained by calculating each top to a reference plane.

With enough measurement points, both methods can achieve a good estimation of the target volume, however, the isolated points method is computationally less intensive and trivial to implement than the meshed surface, which might be a conclusive factor when deciding which method to implement.

Xiang and Zhou [17] describe an algorithm for 3D reconstruction to be used by indoor mobile robots. This algorithm takes the range measurements from the laser system and does the initial processing in 2D, comparing each measurement line in order to match them into surfaces. This surface reconstruction technique could then be applied in the meshed surface method for calculating the volume of a target scan.

Morris et al. [18] and Matos [19] propose 3D reconstruction for navigational purposes by using an occupancy grid. A grid of cells is generated from the available space where measurements were taken, then the measurement properties are calculated and assigned to each cell in the grid, resulting in a 3D or 2D (depending on the application) representation of the field of view of the sensor system. This solution requires a balance between the number of generated cells and computational power required to process all the cells, leading to a compromise in the precision the system can get.

Dias et al. [20] [21] show an approach to create 3D models by combining the range information (for the geometry) with digital photographs (for the textures). This process is

done by computing a texture map for the model by linking the texture of the photos with the range data (which has been previously processed by using a 2D Delaunay triangulation to create the surfaces), allowing to achieve photo-realistic 3D models of real scenes.

Miura et al. [11] do 3D reconstruction using the information from a stereoscopic system by doing an area-based correspondence matching from the points measured by both cameras, the camera parameters are then used to calculate the 3D points of the reconstruction.

2.1.1.5. Comparison of 3D ranging techniques

Each ranging method has strengths and weaknesses as identified by Sansoni et al. [4] and shown in Table 2.2.

Table 2.2: Comparison of ranging methods (Source: [4], slightly adapted)

Methods	Strength	Weakness
Laser triangulation	Simple implementation; High tolerance to ambient noise; High acquisition of range data; Low cost.	Limited measurement range; Loss of range information due to dead zones due to positional (system and target object) constraints.
Structured light	High acquisition of range data.	Limited measurement range; Computationally complex; Loss of range information due to dead zones due to positional (system and target object) constraints; Sensible to ambient noise.
Stereo vision	Simple implementation; High accuracy; Low cost.	Computationally intensive; Sensible to ambient noise.
Time-of-flight	Large measurement range; High tolerance to ambient noise.	Low accuracy at close ranges; High cost.
Interferometry	High accuracy.	Range data dependent on targets with good positional characteristics (flat surfaces); High cost.

The volume measurement and 3D reconstruction methods can be applied on the range data obtained from these ranging methods, with running times being generally associated with the quantity of necessary data to reconstruct the surface of the object.

Simplicity of implementation and cost are two factors that have high importance on the development of systems to be used on logistic applications (such as cargo management), therefore, laser triangulation and stereo vision were the two selected methods for two prototype systems detailed under this dissertation.

2.1.2. Container Loading Problem

In the literature, the Container Loading Problem (CLP) [3] can be interpreted as a geometric assignment problem, where small objects (boxes) are put inside a bigger container. This can be expressed by considering that cuboid (boxes and containers) objects are defined by a geometry comprising a length, a width and a height:

$$\bigcup_{i=1}^N \text{Geometry}_{\text{boxes}_i} \subseteq \text{Geometry}_{\text{Container}} \quad (2.1)$$

This problem is considered NP-hard which means that it can't be solved in polynomial time, therefore, heuristics are often the only viable option to produce solutions within reasonable running times [22].

When the set of cargo to be loaded in the container is comprised of only identical boxes, it is said to be homogeneous, if it has many different types of boxes it is strongly heterogeneous, and if it has only a limited set of boxes it is weakly heterogeneous [23]. When enough containers are available to accommodate all cargo, the loading problem is defined as of the input minimization type, since the objective is to minimize the number of required containers. On the other hand, problems where there are a limited number of containers that can only pack a subset of the available cargo, are defined as of the output maximization type, since the objective is to maximize the volume usage of the available containers and pack them with as much relevant cargo as possible [3].

Wang et al. [22] show a heuristic using a tertiary tree structure to dynamically decompose the available space into subspaces that are then filled with the best calculated type of boxes. At each box placement, the remaining space is divided into three regions, then

each region is filled with a new box, repeating the division process until the whole container is filled (this being a recursive process, it is stopped at a certain amount of steps due to the known dimensions of the boxes). This algorithm was designed to work with no constraints and a set of weekly heterogeneous cargo.

Moura and Oliveira [24] propose a solution using a constructive algorithm along with the Greedy Randomized Adaptive Search Procedure meta-heuristic to improve the solution performance, where it first builds a solution and then uses local search to improve the output. This algorithm works by building layers of boxes (transversal walls where the depth is defined by the first box put on the layer). These boxes are selected by computing all types of boxes considering all possible orientations and then selecting the best box and respective orientation that better fills the space. Free spaces are then filled by selecting the best box type. The algorithm was developed considering only orientation, cargo stability and volume constraints, obtaining results by using a set of weakly heterogeneous cargo with small runtimes, as reported during the computational tests done by the authors. Parreño et al. [25] also developed a similar solution, with the first set being done by a constructive heuristic algorithm that places boxes and calculates the maximal-spaces that are left after the placement (akin to the Wang et al. [22] method of volume region division).

Gehring and Bortfeldt [23] developed a genetic algorithm that works by generating a set of disjunctive towers of boxes which are then arranged on the container's floor in order to better occupy the available volume. The towers are first generated by a greedy algorithm that tends to minimize the free space on each tower, then the genetic algorithm calculates which tower bases provide the best cover of the area of the container's floor, maximizing the total value of the packed boxes. This cycle is repeated several times with various combinations being generated, at the end of the execution the best generated solution is used. This algorithm considers the orientation, stacking, total weight, stability and balance constraints, being suitable for problems where simple stability requirements are sufficient, and stronger when working with a strongly heterogeneous set of cargo. Araújo and Pinheiro [26] also propose a solution using a genetic algorithm. First it uses a wall building algorithm to maximize the used space, then the genetic algorithm calculates the best weight distribution of the selected cargo.

Bortfeldt and Wäscher [3] identified that most of the available literature is focused on a few subsets of the CLP, such as having few or no constraints on the loading of cargo, as in

minimizing the number of used containers when in packing a weakly heterogeneous set of cargo, or packing a set of cargo with one or more variable dimensions while minimizing the available volume in the container. This reveals that there is still room for more research to be done especially with multi-constraint systems.

2.2. Market Solutions

There is a wide variety of sensors for 3D reconstruction, as well as cargo planning software. Some of the most relevant and popular sensors are presented along with their hardware characteristics. Cargo planning software available in the market are also presented and their features described.

2.2.1. Range Finding for 3D Reconstruction

For laser range finding solutions there are some sensors worthy of mentioning due to their application, such as the **Velodyne HDL-64E LiDAR** (shown in Figure 2.7) used on the Google Self-Driving Car Project. This time-of-flight sensor was designed for obstacle detection and navigation of vehicles, it has 64 laser channels that allow for a 360° horizontal and 26.8° vertical field of view, it can generate up to 2.2 million points per second achieving an accuracy below 2 centimeters and has a range of 120 meters [27]. The price of a unit is around \$75000 [28].

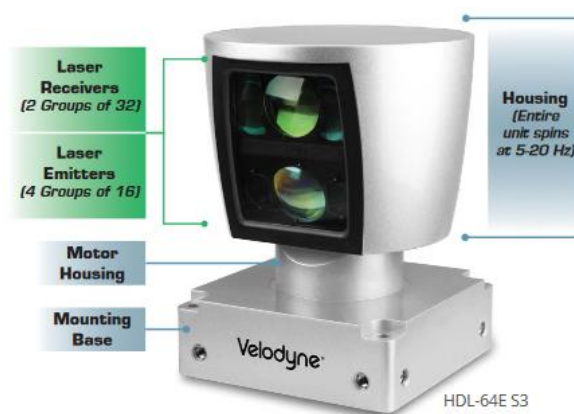


Figure 2.7: Velodyne HDL-64E S3 (Source: [27])

Another time-of-flight sensor is the **Sick LMS 200** (Figure 2.8) used by the ATLAS Project on the ATLASCAR. It has a wide range of applications from determining volumes

to vehicle navigation, a scanning angle of 180° and a range of 10 meters [29]. The price of this sensor rounds about \$6000 [30].



Figure 2.8: Sick LMS200 (Source: [29])

A highly popular sensor is the **Kinect 1st generation** (Figure 2.9) by Microsoft, a structured light sensor developed to be used on consoles but also adopted by the robotics community for various projects. This sensor has a measuring range up to 3.5 meters with 57° horizontal and 43° vertical field of view, and an error inferior to 4 centimeters [31]. This model is out of production after the launch of a new version.

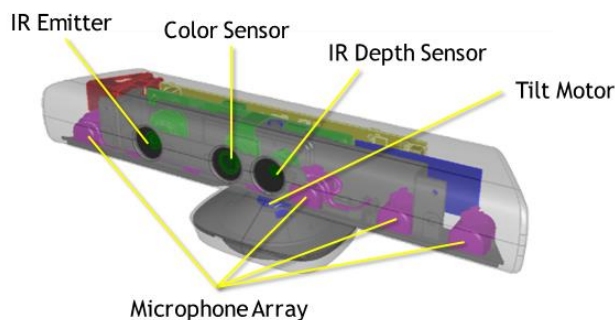


Figure 2.9: Kinect 1 (Source: [32])

The new iteration, **Kinect 2nd generation** (Figure 2.10), is a time-of-flight camera with a similar range as the first generation for the same applications, however it has 70° horizontal and 60° vertical field of view [31]. This sensor is priced at \$100 [33].

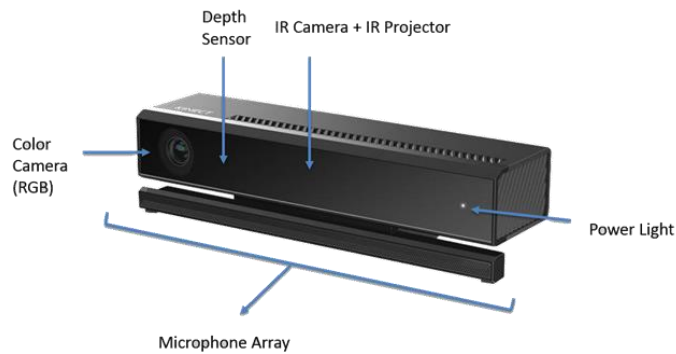


Figure 2.10: Kinect 2 (Source: [34])

Bumblebee2 (Figure 2.11) by Point Grey, is a stereo camera sensor that was used by Princeton University in the DARPA Urban Challenge. This sensor has a range up to 60 meters, outputting 640x480 images at 48 frames per second and having a 100° of field of view [35]. A unit costs \$1895 [36].



Figure 2.11: Bumblebee2 (Source: [37])

Another interesting stereo camera sensor is the **ZED Stereo Camera** (Figure 2.12) developed by Stereolabs (and currently partnering with NVIDIA for inclusion of its stereo sensing technology into vehicles). This sensor was designed to work both indoors and outdoors, having a range up to 20 meters with configurable depth image output, an angle of 110° field of view and a price of \$449 per unit [38].



Figure 2.12: ZED Stereo Camera (Source: [38])

Intel RealSense R200 (Figure 2.13) is also a stereo camera sensor developed by Intel. Built for 3D measurement of environments both outdoors and indoors, this camera has a range up to 5 meters and a field of view of 59° (for depth perception) [39] [40]. This sensor uses infrared cameras and an infrared projector to illuminate the scene to facilitate the image capture. It is currently available in the market for \$99 [41].



Figure 2.13: Intel RealSense R200 (Source: [39])

2.2.2. Container Loading and Planning

There is an extensive offer of cargo planners and packers in the market, with most of them offering the same features. However, given their commercial nature, the principles behind their algorithms are not revealed which makes it harder to compare the different software. Nonetheless, a list of some of these commercially available software will be presented focusing on some of the differentiating factors between solutions. This list is partially based on the selection done by Sarwar [42].

Single Container Packing by 3DBinPacking is an online service that provides packing simulations for boxes into containers. The software takes the dimensions of the boxes and the container, and creates a 3D representation of the optimal assignment of the boxes and a step by step loading diagram. It also provides a list of the boxes that can't be loaded (after the container reaches its volume capacity) and the packing statistics. This service is available from \$15 to \$1500 per month depending on the number of server requests required [43].

CargoWiz by Softtruck provides the standard loading optimization as well as more features such as allowing the user to set various loading constraints; orientation, stacking limits, loading priorities and total weight limits. It also provides cost calculation per package being shipped. A single license is available for \$747 [44].

Cargo Optimizer by Dreamsofts Optimization allows to optimize cargo loading providing support for interlock patterns (loading patterns that promote cargo and container

stability) as well as the weight distribution graph of the loaded container. The software is available as a license or full version with the cheapest being a €39 yearly license [45].

CubeMaster by Logen Solutions provides load optimization with constraints such as stacking, orientation and weight, with the option of providing loading/unloading sequences. It is available as an online service or standalone software, with the online version being priced as \$49 per month [46].

Packer3d by Packer 3d has interesting features such as the ability to load container from the side (instead of from the bottom, this is useful for containers whose door is on the side) and along with the common constraints (orientation, stacking, etc.) it adds the ability to define goods as temperature sensitive which makes the algorithm pack them the furthest away possible from the doors. The software is available as a standalone license or online service, with the service costing €120 per month [47].

3. Concepts and Design

Providing full cargo management control is a complex endeavor due to the many factors and constraints associated with the logistic environments. Two important logistic subsets of activities were selected as the focus of this dissertation:

- **Cargo Planning**

This process deals with the cargo assignment to containers (that can also be pallets) in order to improve the efficiency of the logistic operations by providing the operators with a system to help making a decision of which cargo to transport.

- **Sensing Methods**

Used during loading or unloading of the cargo to monitor and verify that the process is being conducted according to the planned assignment (executed during the cargo planning phase), standards or laws are being applied, as well as monitoring the status of the cargo for potential issues.

The necessity of incorporation of both activities is explicit when considering that the logistic operator after being tasked with the selection of cargo that should be processed into transportation (usually aided by decision support systems), is often blind to the execution processes occurring during transportation (while the cargo being loaded or with the vehicles already on the road). Therefore, sensing systems can play role in providing up to date information on the state of the transportation processes making it easier for the operator to evaluate the performance of the chain of activates as well as reacting faster to new opportunities or hazards that might occur during these processes.

This section will cover the concepts and designing aspects of solutions developed under these two components.

3.1. Cargo Planning

Cargo shipment planning (Figure 3.1) is necessary in order to perform the assignment of goods to the available methods of transportation. To perform a subset of this task, a heuristic for volume assignment of containers was developed and implemented.

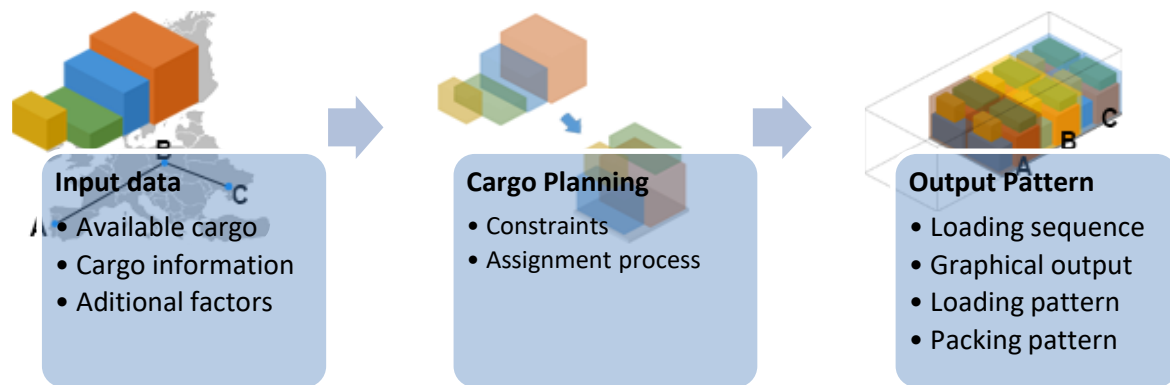


Figure 3.1: General planning process

The developed heuristic functions as a decision support system to aid the logistic operator improving the performance of cargo loading processes.

3.1.1. Considered Scenarios

In a typical logistic scenario, a logistic operator is faced with the task of selecting which cargo (boxes or pallets) need to be processed into transportation (containers or trucks). Therefore, the developed heuristic considers two main cases based on challenges faced by the operators. These cases are (as exemplified by Figure 3.2):

- **Pallet assembling:** boxes are assigned to pallets (in this case a pallet is a container) producing a packing pattern.
- **Cargo loading:** pallets (viewed as cargo and grouped as bundles) need to be assigned to a container producing a loading pattern.

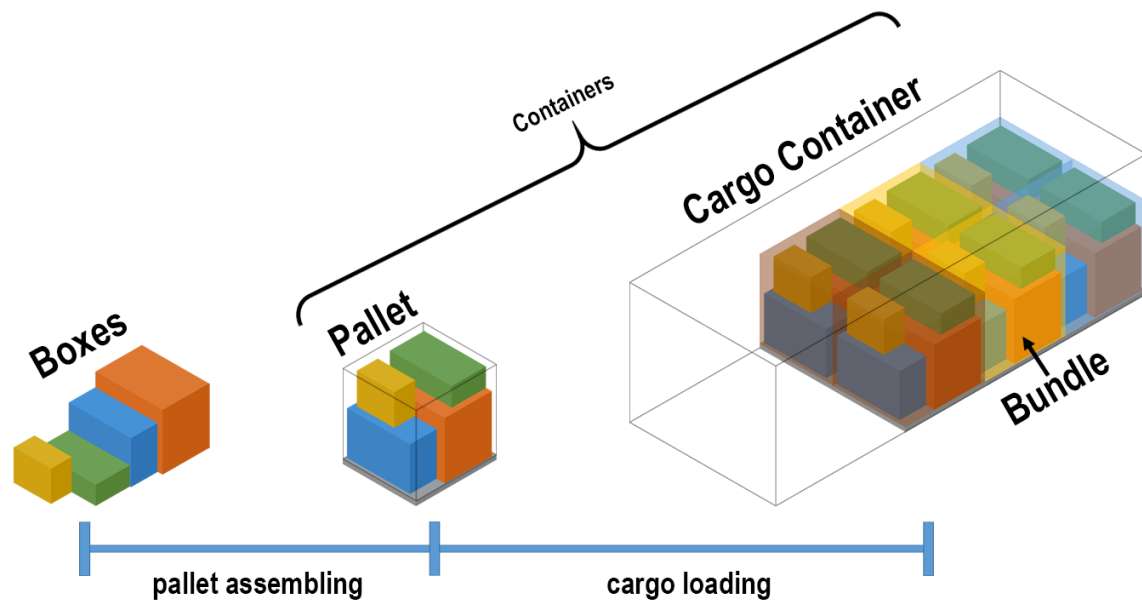


Figure 3.2: Pallet assembling and Cargo loading formulations

A **packing pattern** is a geometric assignment that is constructed when the focus is to maximize the occupied space in smaller volumes such as a pallet, while at the same time holding the physical constraints associated with such problems (stability, weight, etc.).

A **loading pattern** are obtained by trying to maximize the value (such as profit, priority, etc.) of transporting goods of larger scale (such as pallets being loaded into a cargo area of a truck or a container). Only non-stackable cargo units are considered for the generation of these patterns (this is the typical case of standard pallets, stackable solutions might however be applied in certain logistic environments).

3.1.2. Definitions and Geometric Model

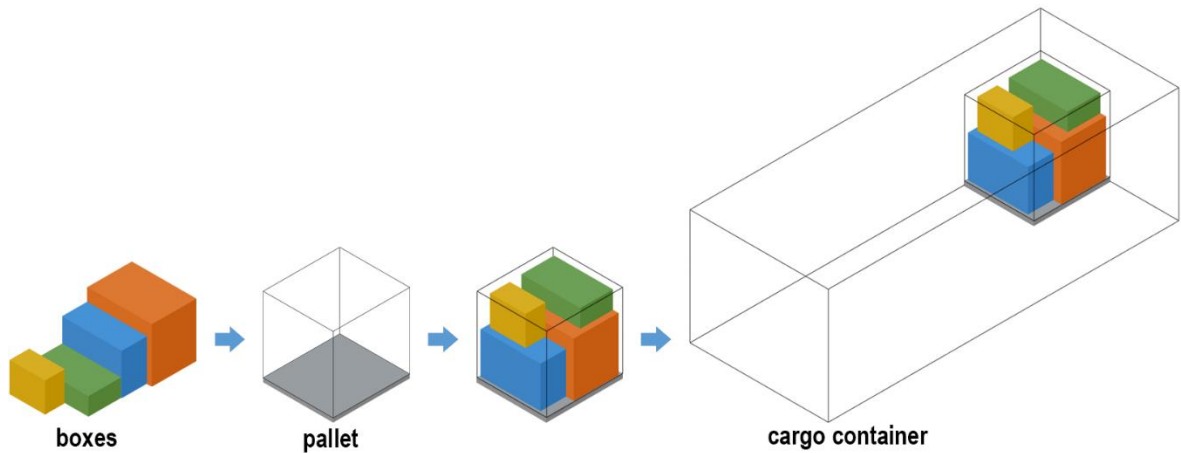


Figure 3.3: 3D problem representation

The representation model (Figure 3.3) consists of the following elements:

- **Boxes** (also referred to as **cargo**): three dimensional objects that fill containers.
- **Containers**: three dimensional objects that contain cargo. Containers can be of two types:
 1. **Pallets**: hold boxes.
 2. **Cargo Containers**: hold pallets (viewed as cargo).

Throughout this dissertation both boxes and containers are always assumed to be cuboid, i.e., closed three dimensional objects composed of six rectangular faces joined at right angles [48].

3.1.2.1. Pallet Assembling

Formally, the t -th box type, g_t , exemplified by Figure 3.4, is an entity characterized by the 3D geometry information of a box:

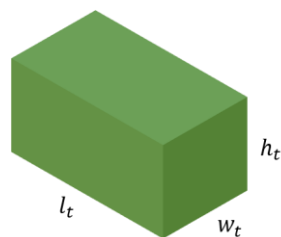


Figure 3.4: Box type

$$g_t = (l_t, w_t, h_t) \quad (3.1)$$

With

$$t \in \{1, 2, \dots\} \subset \mathbb{N}$$

$$l_t, w_t, h_t \in \mathbb{R}_{>0}$$

Where

l_t – length

w_t – width

h_t – height

The n -th container type entity, g_n^* , exemplified by Figure 3.5, characterizes the geometry and maximum weight supported by the container.

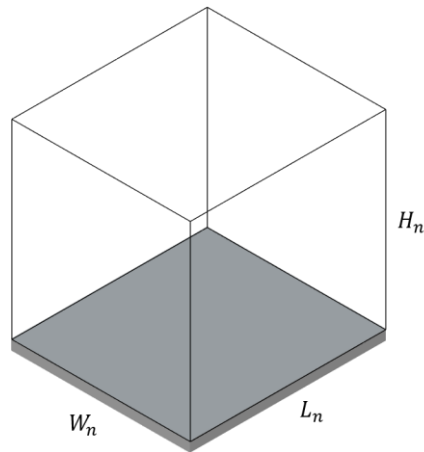


Figure 3.5: Container type

$$g_n^* = (L_n, W_n, H_n, \omega_{max,n}) \quad (3.2)$$

With

$$n \in \{1, 2, \dots\} \subset \mathbb{N}$$

$$L_n, W_n, H_n, \omega_{max,n} \in \mathbb{R}_{>0}$$

Where

L_n – length

W_n – width

H_n – height

$\omega_{max,n}$ – maximum weight supported

The instantiation of a box type produces a particular i -th box, B_i , that can be represented as follows:

$$B_i = (g_t, \omega_i) \quad (3.3)$$

With

$$i \in \{1, 2, \dots\} \subset \mathbb{N}$$

$$\omega_i \in \mathbb{R}_{>0}$$

Where

g_t – geometry of the box (box type)

ω_i – weight

Each i -th instance has an associated pose, p_i , in relation to the container (in this case a pallet as exemplified by Figure 3.6) where it is placed.

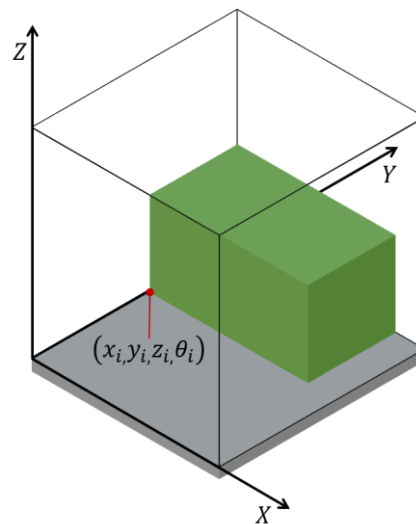


Figure 3.6: Box pose

$$p_i = (x_i, y_i, z_i, \theta_i) \quad (3.4)$$

With

$$i \in \{1, 2, \dots\} \subset \mathbb{N}$$

$$x_i, y_i, z_i \in \mathbb{R}_{>0}$$

$$\theta_i \in \{0^\circ, 90^\circ\}$$

Where

x_i – position along X axis

y_i – position along Y axis

z_i – position along Z axis

θ_i – azimuthal angle

In order to store the information of different boxes and respective poses, a payload entity, ψ_j , is defined as set of pairs of boxes and their poses inside the pallet.

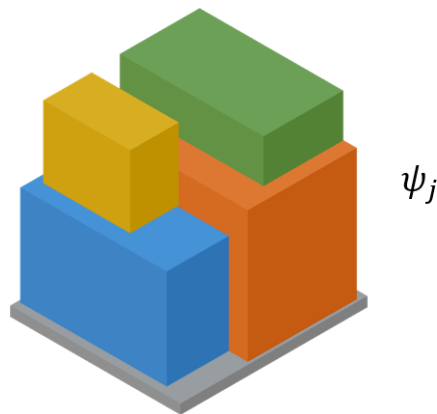


Figure 3.7: Payload

$$\psi_j = \{(B_i, p_i)\} \quad (3.5)$$

With

$$j \in \{1, 2, \dots\} \subset \mathbb{N}$$

Where

B_i – instantiation of a box

p_i – pose of the box

From the universe of possible pairs of boxes and poses, only those that are contiguous constitute elements of suitable payloads to be assigned to pallets.

Pallets, P_j , are containers that hold boxes (in their poses, i.e., the payload) and are formally described as:

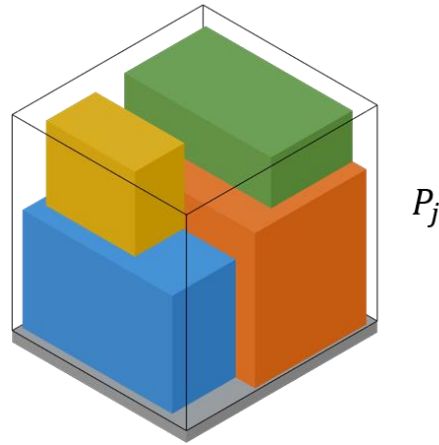


Figure 3.8: Pallet

$$P_j = (g_n^*, \psi_j) \quad (3.6)$$

With

$$j \in \{1, 2, \dots\} \subset \mathbb{N}$$

Where

g_n^* – geometry of the container (container type)

ψ_j – payload assigned to the pallet

Each pallet has its own individual payload, therefore, the assignment of the j -th payload to the j -th pallet requires the payload to follow two geometric conditions:

1. Boxes are placed entirely within the container limits.

Considering a payload ψ_j with boxes B_i whose box type is g_t , assigned to a pallet P_j whose container type is g_n^* :

$\forall_{(B_i, p_i)} \in \psi_j \wedge i \in [1, 2, \dots, N] \subset \mathbb{N}$:

$$\begin{aligned} x_i + l_t &\leq L_n \\ y_i + w_t &\leq W_n \\ z_i + h_t &\leq H_n \end{aligned} \quad (3.7)$$

2. Boxes don't overlap.

Considering a payload ψ_j with boxes B_i whose box type is g_t . The Region

R_{ρ_i} occupied by $\rho_i = (B_i, p_i) \in \psi_j$ is:

$$R_{\rho_i} = \{(x, y, z): x_i \leq x \leq x_i + l_t \wedge y_i \leq y \leq y_i + w_t \wedge z_i \leq z \leq z_i + h_t\}$$

Therefore, $\forall \rho_i, \rho_s \in \rho_j \wedge \rho_i \neq \rho_s$:

$$R_{\rho_i} \cap R_{\rho_s} = \emptyset \quad (3.8)$$

As a consequence of these two conditions:

- The sum of the volume of the boxes inside the pallet is less or equal to the volume of the pallet.

Considering a payload ψ_j with boxes B_i whose box type is g_t , assigned

to a pallet P_j whose container type is g_n^* :

$$\sum_{t=1}^N l_t w_t h_t = \sum_{t=1}^N v_t \leq L_n W_n H_n = V_n \quad (3.9)$$

Where

v_t – volume of a box

V_n – volume of the container (pallet)

N – number of assigned boxes

3.1.2.2. Cargo Loading

A cargo pose entity, p_j^* , exemplified by Figure 3.9, contains the positional information of pallets (viewed as cargo) when assigned to a cargo container. Since pallets are assumed to be non-stackable, the model only needs to work in two dimensions.

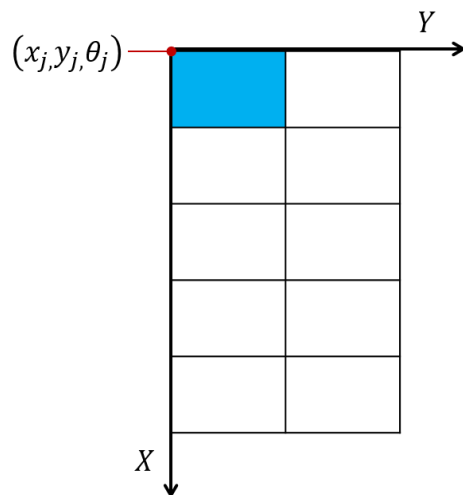


Figure 3.9: Cargo pose

$$p_j^* = (x_j, y_j, \theta_j) \tag{3.10}$$

With

$$j \in \{1, 2, \dots\} \subset \mathbb{N}$$

Where

x_j – position along X axis

y_j – position along Y axis

θ_j – azimuthal angle

A bundle, D_r , is a group of pallets bound together and is defined as:

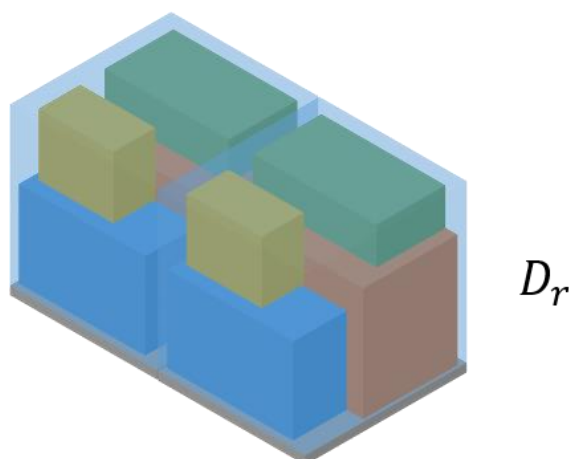


Figure 3.10: Bundle

$$D_r = (\{P_j\}, \mu_r, \lambda_r) \quad (3.11)$$

With

$$r \in \{1, 2, \dots\} \subset \mathbb{N}$$

$$\mu_r \in [0, 1]$$

$$\lambda_r \in \mathbb{R}_{\geq 0}$$

Where

P_j – pallet from the set of assigned pallets

μ_r – bundle individual priority

λ_r – value from the cost function (logistic importance)

A cargo payload, ψ^* , is the entity that associates pallets from bundles and their pose inside the cargo container.

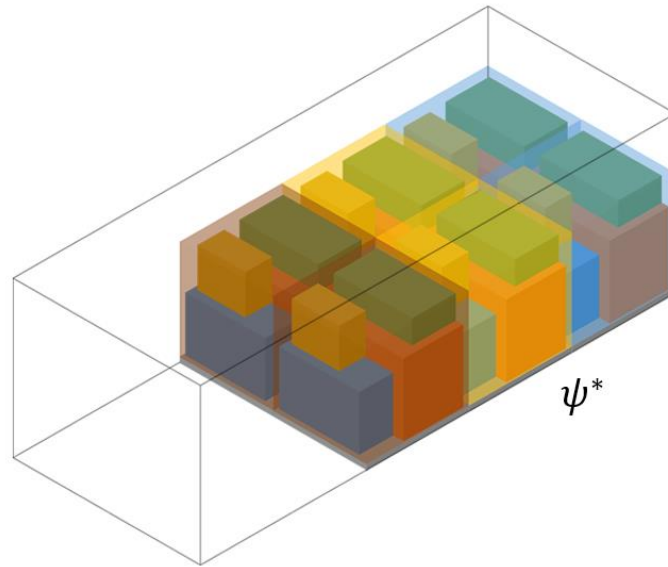


Figure 3.11: Cargo payload

$$\psi^* = \{(P_j, p_j^*)\}, P_j \in D_r \quad (3.12)$$

Where

P_j – pallet assigned to the bundle

p_j^* – pose of the pallet

The cargo container, C , holds the assigned pallets from the selected bundles according to a specific layout given by the cargo payload pose.

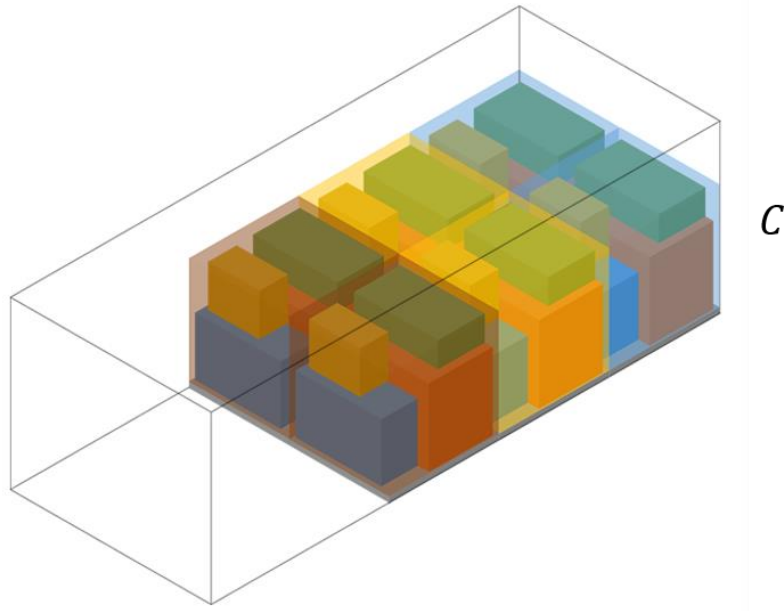


Figure 3.12: Cargo container

$$C = (g_n^*, \{D_r\}, \psi^*, c) \quad (3.13)$$

With

$$c \in \{1, 2, \dots\} \subset \mathbb{N}$$

Where

g_n^* – geometry of the container (container type)

D_r – bundle from the set of assigned bundles

ψ^* – cargo pose

c – capacity (number of pallets allowed into the container)

Like the previous case of assignment of boxes into pallets, assigning pallets into cargo containers needs to fulfill two geometric conditions. However, given the simplification made to the model (due to pallets being non-stackable) these conditions can be rewritten as:

1. Pallets are placed entirely within the cargo container limits.

Considering a cargo payload ψ^* with pallets P_j (belonging to the assigned bundles D_r) whose container type is g_k^* , assigned to a cargo container C whose container type is g_n^* :

$$\forall_{(P_j, P_j^*)} \in \psi^* \wedge j \in [1, 2, \dots, c] \subset \mathbb{N}: \quad (3.14)$$

$$x_j + L_j \leq L_n$$

$$y_j + W_j \leq W_n$$

2. Pallets don't overlap.

Considering a cargo payload ψ^* with pallets P_j (belonging to the assigned bundles D_r) whose container type is g_k^* . The Region $R_{\varrho_j^*}$ occupied by

$\varrho_j^* = (P_j, p_j^*) \in \psi^*$ is:

$$R_{\varrho_j^*} = \{(x, y): x_j \leq x \leq x_j + L_k \wedge y_j \leq y \leq y_j + W_k\}$$

Therefore, $\forall \varrho_j^*, \varrho_f^* \in \psi^* \wedge \varrho_j^* \neq \varrho_f^*$:

$$R_{\varrho_j^*} \cap R_{\varrho_f^*} = \emptyset \quad (3.15)$$

As a consequence of these two conditions:

- The sum of area of the pallets is less or equal to the total area of the container.

Considering a cargo payload ψ^* with pallets P_j (belonging to the assigned bundles D_r) whose container type is g_k^* , assigned to a cargo container C whose container type is g_n^* :

$$\sum_{k=1}^c l_t w_t = \sum_{k=1}^c a_k \leq L_n W_n = A \quad (3.16)$$

Where

a_k – area of each pallet

A – total area of the cargo container

3.1.2.3. Data Structure

These entities and their relationships are implemented into a data structure whose class diagram is shown in Figure 3.13.

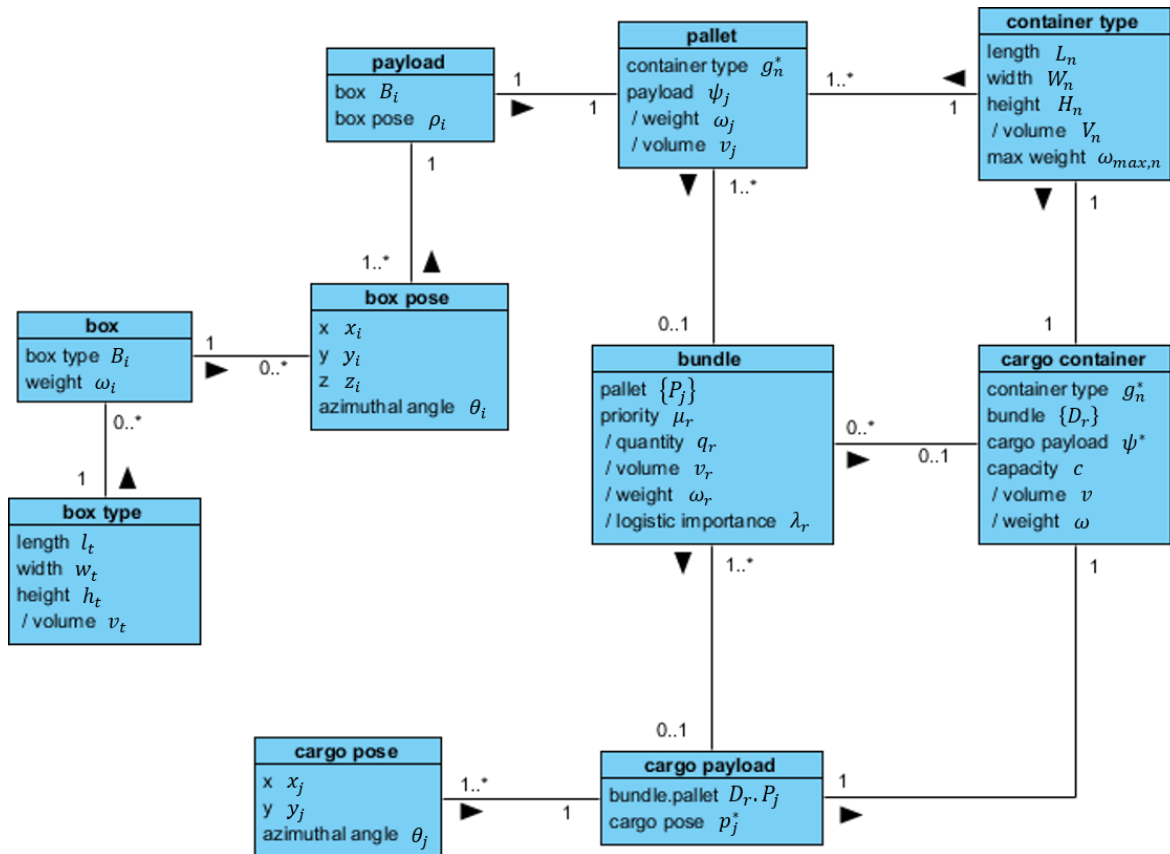


Figure 3.13: Class diagram

The following list describes the additional derived elements necessary for the software implementation.

box type

/volume (v_t) – calculated from the geometry

container type

/volume (V_n) – calculated from the geometry

pallet

/weight (ω_j) – calculated from all assigned weights in the payload

/volume (v_t) – calculated from all the assigned volumes

bundle

- /quantity (q_r) – number of assigned pallets to the bundle
- /volume (v_r) – sum of volumes from the assigned pallets
- /weight (ω_r) – Sum of weights from the assigned pallets
- /logistic importance (λ_r) – Calculated from ratios (described in 3.1.3)

cargo container

- /volume (v) – sum of volumes from the assigned bundles
- /weight (ω) – Sum of weights from the assigned bundles

3.1.3. Logistic Importance (Cost Function)

In order to give meaning to the optimization, bundles have an associated variable that can be interpreted as a priority, i.e., a higher value dictates a higher priority in the positioning of the bundle inside the container (e. g. in trucks or containers there is a sequence for loading/unloading where a high priority bundle will be closer to the door). This variable is given by a function that translates the relevant factors involved in the system that is being optimized into a specific numerical value.

This function takes into consideration three major ratios that were considered to have the biggest impact in the selection criteria. These ratios are:

- Percentage of container capacity used by a bundle D_r :

$$Q_{D_r} = \frac{\text{number of pallets in bundle } D_r}{\text{pallet capacity of the container}} = \frac{q_r}{c}$$

This gives relevance to bundles that have the highest number of assigned pallets. It is assumed that the number of pallets assigned to a bundle is never greater than the capacity of the container.

This ratio is used for maximizing the cargo container capacity usage.

- Individual priority of a bundle D_r :

$$\mu_r \in [0,1]$$

Given the different nature of each bundle (which is related to time constraints, client affiliation, etc.), it is often the case that some bundles have higher

priority than others. This is a normalized variable that translates this individual priority.

This factor is used for maximizing the individual bundle priority.

- Percentage of occupied volume by a bundle D_r :

$$V_{D_r} = \frac{\text{sum of volume of the bundle } D_r \text{ pallets}}{\text{volume of the container}} = \frac{v_r}{V_n}$$

This gives relevance to bundles whose assigned pallets have the higher ratio of occupied volume, providing the best overall volume occupation of the cargo container.

This ratio is used for maximizing the occupied volume of the container.

Therefore, the logistic importance function is represented as the following equation:

$$\text{logistic importance} = \lambda_r = \alpha Q_{D_r} + \beta \mu_r + \gamma V_{D_r} + \delta x \quad (3.17)$$

Where α , β , γ and δ are relevance coefficients and x is an open normalized variable that can translate additional ratios or factors that are not declared explicitly (e.g. fuel spent, route's parameters, etc.).

It should be noted that Equation (3.17) is only an example constructed in order to provide realistic scenarios for the executions, however given the modularity of the implementation, this function can be replaced with any other function that better applies to other simulated environments.

3.1.4. Constraints

Constraints are physical or logistical limitations that influence the assignment of boxes to containers in addition to the geometric conditions of the model. They can be considered as hard or soft constraints. Hard constraints must always be respected where soft constraints can be hold but there is room for flexibility in cases where they can't be maintained [3].

For the **Pallet assembling** scenario, the following constraints are considered:

- Weight limits: containers (pallets in this case) have a limit of physically supported weight.

- **Stability:** whenever stacking boxes the structure must be able to hold without boxes falling from the assigned position (due to gravity or external forces), i.e., the vertical center of gravity of the pallet must be in the middle or below the vertical geometric center.
- **Orientation constraints:** boxes will require to have a specific orientation, i.e., a determined face will always face downwards and only rotate along the vertical axis of the container.

For the **Cargo loading** scenario, the following constraints are considered:

- **Weight limits:** containers have a limit of possible weight that can be loaded.
- **Weight distribution:** cargo distribution is important to maintain stability and should be as uniform as possible along the container.
- **Loading priorities:** established by the Last In First Out nature of the container loading process and often related with vehicle routing, cargo have loading priorities to meet the routing criteria.
- **Complete-shipment constraints:** cargo is organized in bundles that require to be shipped together or not at all, which means that these bundles must be considered as a whole when assigning them to containers.

3.1.5. Dimensions and Loading Layout

A loading pattern is represented according to a loading layout (Figure 3.14). Each possible layout depends on factors such as the container's dimensions and type of cargo loaded.

A standard 'euro truck' has 13.6m of length, 2.45m of width and 2.5m of height [49]. These dimensions allow a total of 32 or 33 standard euro pallets loaded depending on the type of cargo. A euro pallet has 1.2m of length and 0.8m of width [50] with variable height limited only by the height of the container where it is placed.

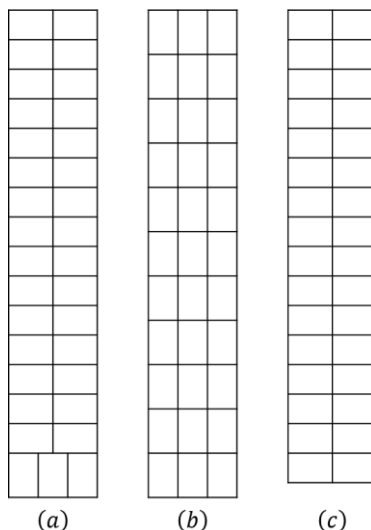


Figure 3.14: Common “euro truck” layouts - (a) 33 pallets, (b) 33 pallets, (c) 32 pallets

The implementation in this dissertation will consider only the layout (a) given its popularity in the logistics bundling sector. The layout is always filled from top to bottom, left to right.

3.1.6. 0/1 Knapsack Problem

The Knapsack Problem is allegorically presented as the thief problem. A thief breaks into a house holding a knapsack that has limited weight capacity. Inside the house there are various objects each having different weights and associated values (all known by the thief and represented in Figure 3.15). The problem is then set: which items should the thief take so that value (profit) can be maximized while not exceeding the weight capacity of the knapsack?

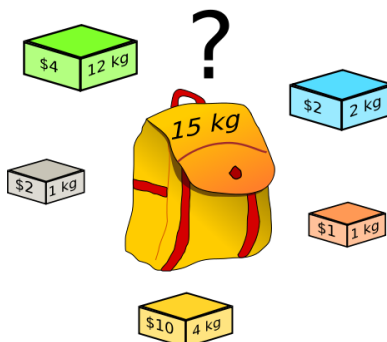


Figure 3.15: The Knapsack Problem (Source: [51])

Formally, this problem is described as a combinatorial optimization problem that seeks to find the best solution to a given container with a fixed capacity (maximum allowed weight) and a set of items with an associated weight and value, which items should be picked in order to maximize the sum of the value without exceeding the given capacity of the container.

A common variation of this problem is the 0/1 Knapsack Problem (**0/1 KSP**) which can be defined as [51] [52]:

$$\begin{aligned}
 & \text{maximize } \sum_{i=1}^n v_i x_i \\
 & \text{subject to } \sum_{i=1}^n w_i x_i \leq W \text{ and } x_i \in \{0,1\}.
 \end{aligned} \tag{3.18}$$

Each item i (from 1 to n) has a weight w_i and a value v_i , and x_i is a binary variable that is equal to 1 if the item is selected to be loaded into the container or 0 otherwise. The container has a maximum weight capacity of W .

A dynamic programming approach can be implemented for the 0/1 KSP assuming that all w_i and W are positive integers. Defining $m[i, w]$ as the maximum value that can be obtained with the first i items that fit into the container with capacity w , the algorithm starts with the following conditions:

- $m[0, w] = 0$ (no items are included in the container)
- $m[i, 0] = 0$ (the container has capacity 0 therefore no items fit in it)

Then, the solution is calculated recursively [52]:

- $m[i, w] = m[i - 1, w]$ if $w_i > w$ (else can't fit the item)
- $m[i, w] = \max \left(\begin{array}{l} m[i - 1, w], \\ m[i - 1, w - w_i] + v_i \end{array} \right)$ if $w_i \leq w$ (don't use the item / use the item)

The goal is to find the maximal value $m[n, W]$ of the items allowed into the container.

The implemented approach has a complexity of $O(nW)$. Further implementation details, including the pseudocode, are available in Annex B.

The 0/1 KSP algorithm excels at providing optimization in one dimension, which makes it a viable choice to be used as a mechanism to select optimized combinations of items from a list of all potential candidates.

3.2. Sensing Methods

Even though there are various methods that can be used to implement a volume measurement system, there are two that are the focus of this dissertation:

- Laser Rangefinder (**LRF**) with optical triangulation.
- Computer Stereo Vision (**SV**).

Two (prototype) systems using these methods were implemented at the Department of Electronics, Telecommunications and Informatics of the University of Aveiro.

3.2.1. Requirements

The sensing systems developed under this work are seen as proof of concept and had to fulfill specific requirements. Functional requirements:

- The system shall calculate the distance from a reference point (itself) to a target object (point in space, usually a box or any of its features).
- The system shall calculate the occupied/available volume.
- The software shall output a 3D visualization of the occupied/free volume.

Non-functional requirements include:

- The software shall be multi-platform.
- The system shall be low-cost.
- The physical dimensions of the sensors shall be reduced (to minimize interference with the physical loading/unloading of cargo).
- The process execution time shall be reduced or within a reasonable time.

3.2.2. Laser Rangefinder

The first approach considered was a laser rangefinder system using optical triangulation. This is a low-cost and easy to implement approach that provides a reliable distance reading that is then used to estimate the volume of target objects.

The developed system (Figure 3.16) is composed of a camera and a laser mounted in a servo mechanism able to provide pan and tilt rotations.

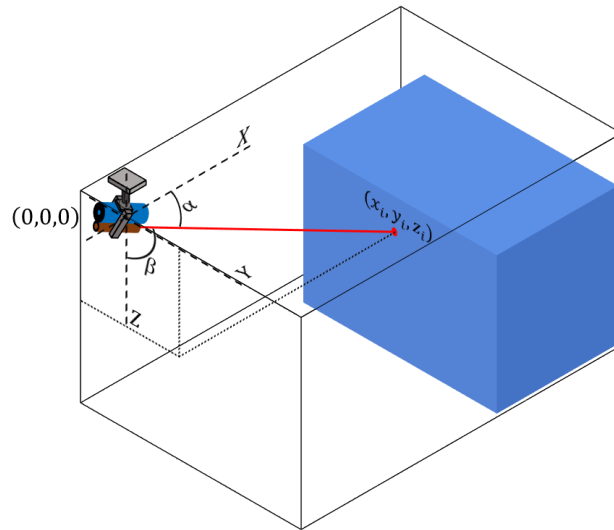


Figure 3.16: LRF system overview

This system uses mono-computer vision and triangulation to calculate the distance between the system to the target point. The measuring approach was based on Danko [53].

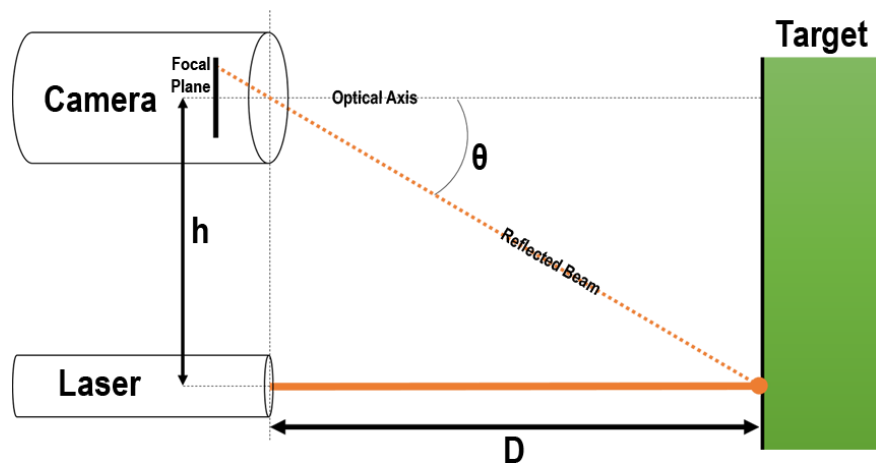


Figure 3.17: LRF with triangulation (based on [53])

A laser at a distance h (also called baseline) and aligned with the optical axis of the camera is projected to a target (Figure 3.17), the resulting reflected beam is then captured in the focal plane of camera which is translated into a representation of the laser dot along the vertical axis of the image. This vertical distance is then used to calculate the distance D to the target, as given by the following equation:

$$D = f(h, \theta) = \frac{h}{\tan(\theta)} \quad (3.19)$$

With θ being the optical axis angle which is given by the distance in pixels from the laser dot to the center of the focal plane.

A frame is a two dimensional entity captured from the camera, with a width (w_f) and height (h_f) which are measured along the horizontal and vertical axis, respectively. Both dimensions represent a number of pixels. Along the vertical axis there is also the maximum vertical field of view (θ_0), therefore each pixel corresponds to a fraction of this angle:

$$\Delta\theta = \frac{\theta_0}{h_f} \quad (3.20)$$

The error of measurement of this system can be calculated using the propagation of error method [54].

$$\Delta D = \left| \frac{\partial f}{\partial \theta} \right| \Delta\theta + \left| \frac{\partial f}{\partial h} \right| \Delta h = \frac{h}{\sin(\theta)^2} \Delta\theta + \frac{1}{\tan(\theta)} \Delta h \quad (3.21)$$

Equation (3.21) can then be represented in a graph (as shown in Figure 3.18) to better understand the variation of the measurement error.

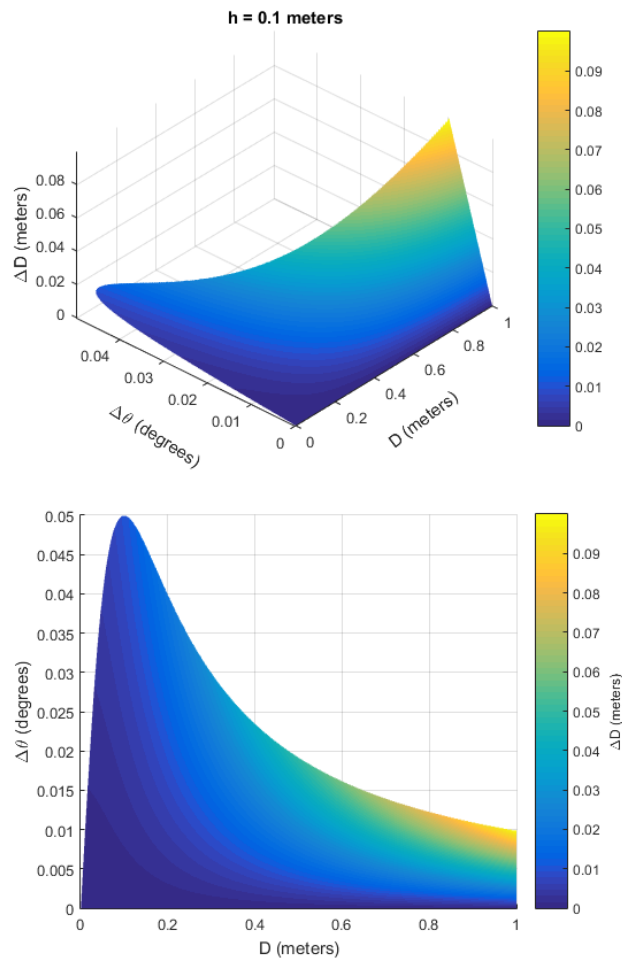


Figure 3.18: LRF error variation for values below 10% of the distance and $\Delta h = 0$

The error increases with the distance if there are angle mismatches (i.e. incorrect recognition of the pixels that represent the laser dot), with the highest variation permitted being at shorter ranges. It is possible to have an increased range of permitted variation by increasing the value of h . This can be seen in the additional graphs available in Annex A.i.

Most cameras have a restricted vertical field of view which affects the ability of the system to take measurements at short ranges, creating a dead zone where measurements can't be made. Having a larger h aggravates this condition since it also increases the area of the dead zone assuming that the field of view of the camera is kept the same.

Figure 3.18 considers $\Delta h = 0$ since h is a constant, i.e., the camera and the laser are always static in relation to each other, even when the system moves to measure different points in space. Ideally, measurements are supposed to take place in stable environments (a stopped truck or non-moving container), however, unpredictable events such as metal

dilatation due to heat exposure or destructive vibrations can introduce an additional error in the measurement as seen in Figure 3.19.

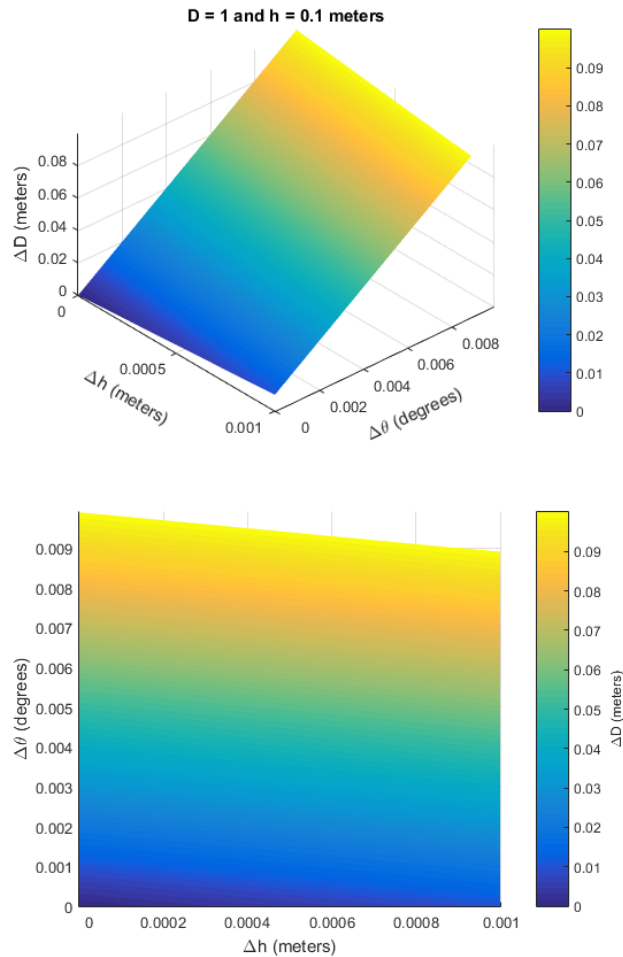


Figure 3.19: LFR error variation for values below 10% of the distance with $\Delta h \neq 0$

A small variation of h can severely impact the measurement reliability of the system. This is even more noticeable at greater distances as it can be seen in the graphs available in Annex A.i. To be able to reduce this impact, cameras would be required to have a higher resolution, which in turn makes them more expensive.

It should be noted that the actual error might be higher due to optical aberrations, the most frequent ones being radial distortions such as barrel or pincushion distortions. Many of these effects can be corrected through software at the expense of additional processing and potential cropping of the frames, which will result in a narrower field of view along both the vertical and horizontal dimensions.

In order to reduce the angle mismatches due to incorrect pixel recognition, curve fitting is done on a set of experimentally acquired data within the range of the distance of the measurements. In this case the data can be fitted with a simple linear regression (for small angles as shown by Danko [53]) with θ being given by the following equation:

$$\theta = \Delta\theta * h_i + \delta \quad (3.22)$$

With h_i being the number of vertical pixels from the center of the image to the center of mass of the visible target (laser dot) and δ being an offset to compensate for eventual alignment errors.

3.2.3. Computer Stereo Vision

The second system uses stereo vision to triangulate the distance to the object as an alternative that allows to have a bigger baseline.

This system (Figure 3.20) uses two aligned parallel cameras that take dual captures from the environment and then calculate the distance to common points of interest (detected features) which are then displayed in a three dimensional representation.

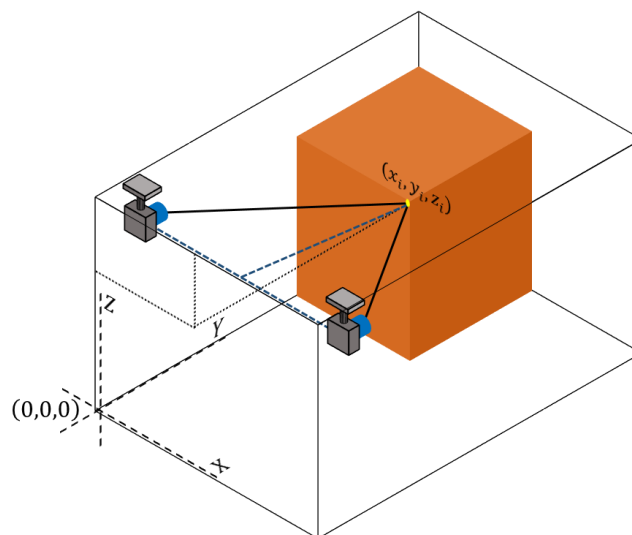


Figure 3.20: SV system overview

From the two acquired images, the system matches the same feature (box corner) calculating its horizontal disparity (in reference to the optical axis of each camera) which is

the key element in determining the distance from the set to the object. This measuring approach was based on Mrovlje and Vrančić [8].

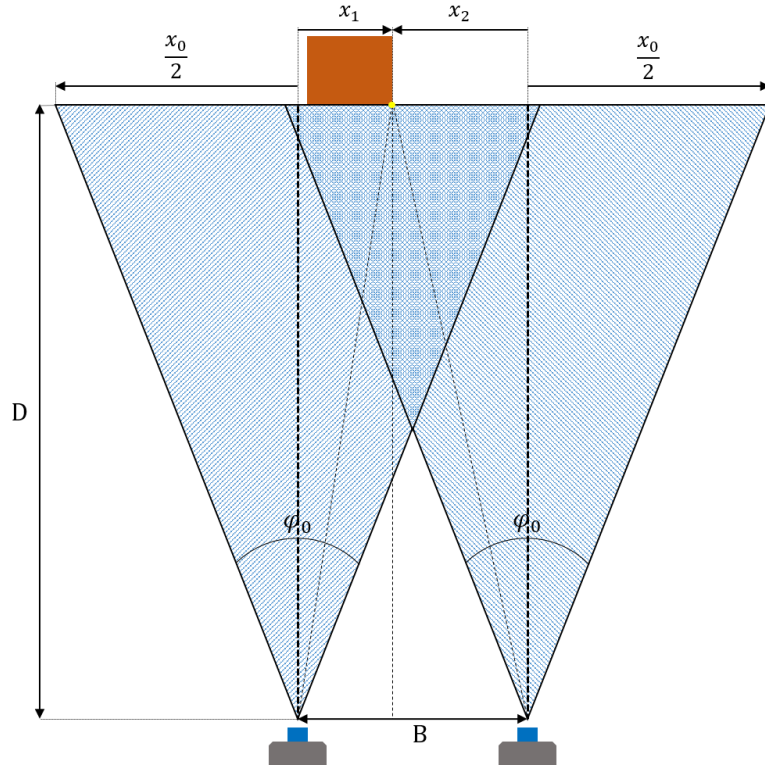


Figure 3.21: Stereo vision with two cameras (based on [8])

Assuming the two cameras are identical (same intrinsic parameters) and the point of interest is between the two optical axis of the set as shown in Figure 3.21, the distance D between the set and the target object is given by the following equation [8]:

$$D = f(B, x) = \frac{Bx_0}{2 \tan\left(\frac{\varphi_0}{2}\right)(x_1 + x_2)} \quad (3.23)$$

Where B (baseline) is the distance between the two cameras, φ_0 is the horizontal view angle, x_0 is the width in pixels of the captured image, x_1 and x_2 are the horizontal distances in pixels from the optical axis to the same relevant feature in both the left and right camera frames, respectively. It is worth to note that the horizontal disparity between the captured images to the feature is given by the component $x = (x_1 + x_2)$, also referred as pixel disparity, which is the element that allows to calculate the distance to the object.

If these geometric assumptions are not met, then the mathematical formulations would be different, and were not considered in this dissertation for this system's design.

The measurement error of this system can then be calculated in a similar fashion as the previous system by using the propagation of error method.

$$\Delta D = \left| \frac{\partial f}{\partial B} \right| \Delta B + \left| \frac{\partial f}{\partial x} \right| \Delta x = \frac{B x_0}{2 \tan\left(\frac{\varphi_0}{2}\right) x^2} \Delta x + \frac{x_0}{2 \tan\left(\frac{\varphi_0}{2}\right) x} \Delta B \quad (3.24)$$

Equation (3.24) can be used to plot a graph showing the evolution of the error along with the distance and horizontal disparity.

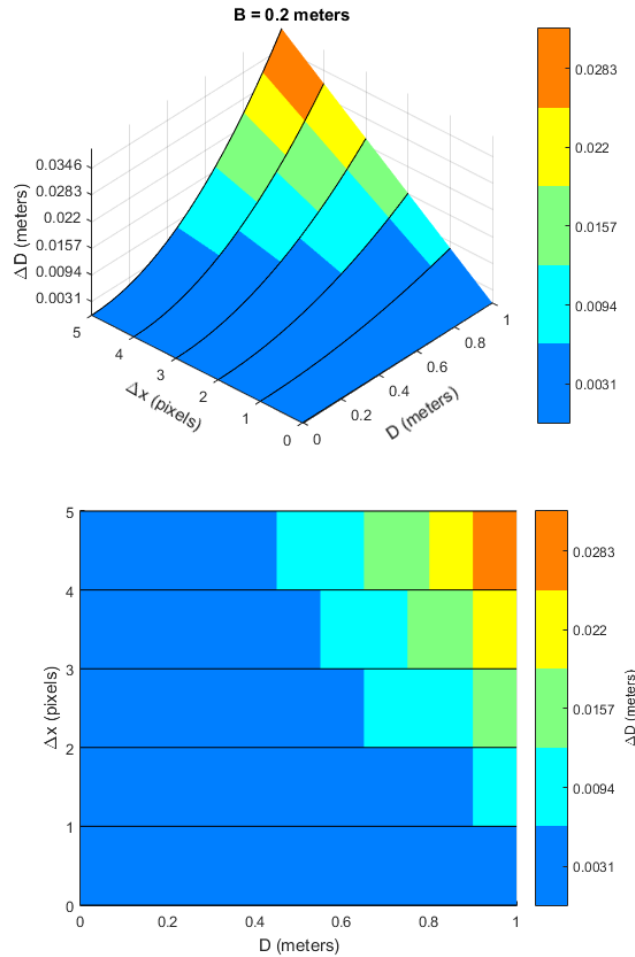


Figure 3.22: SV error variation for values below 10% of the distance and $\Delta B = 0$

Unlike the previous system, this case allows B to take a higher value since both cameras are always in a fixed position (no rotations required), this translates in a lower error variation as shown in Figure 3.22 when compared to the LRF sensor. Annex A.ii contains further

graphs with different baselines showing how bigger baselines provide lower errors with greater distances.

Figure 3.22 considers $\Delta B = 0$ since B is a constant, i.e., cameras are fixed to the measurement locations inside the container. However, for the same reasons as the previous system, an additional source of error can be introduced if the baseline suffers any change.

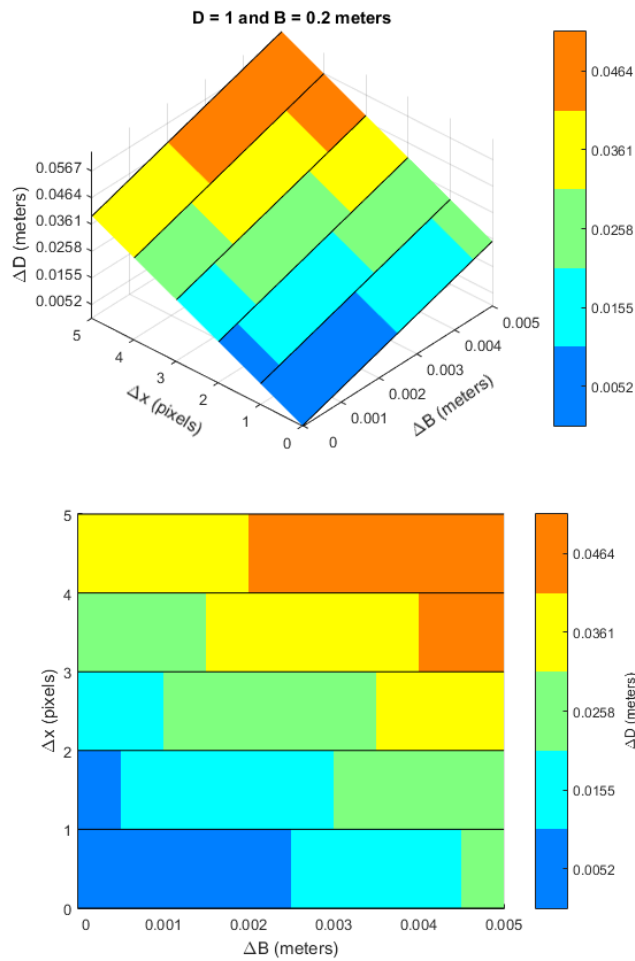


Figure 3.23: SV error variation for values below 10% of the distance and $\Delta B \neq 0$

The stereo vision method is more tolerant of small baseline variations (unlike the LRF system), which is more noticeable with higher baselines even for greater distances as seen in the additional graphs available in Annex A.ii.

Equation (3.23) can likewise be approximated using curve fitting (using data obtained experimentally) which allows the system to provide better results, permitting some tolerance to differences of alignment between the cameras or other potential detection errors. In this

case the power law is the best fit for the input data, with the equation taking the following form [9]:

$$D = kx^z \quad (3.25)$$

Where $x = (x_1 + x_2)$ with k and z being constants that are obtained from the experimental data.

By converting this distance to Cartesian coordinates (using simple trigonometry) and knowing that each distance is related to a corner of the box and the container's dimensions, the system can produce a three-dimensional representation of the visual input.

4. Sensing Systems for Cargo Management

This section will present the assumptions and software implemented on each of the two designed systems. These systems were implemented in a smaller scale than the target environment, which is a truck cargo space or a container. Both systems were designed to measure non-stackable large volumes (such as pallets).

4.1. Laser Rangefinder Prototype

The first prototype (developed under the supervision of Professor A. Manuel de Oliveira Duarte and Professor Pedro Fonseca) is based on a laser triangulation method, and provides a 3D scan of the inside of the container along with an estimation of the occupied volume.

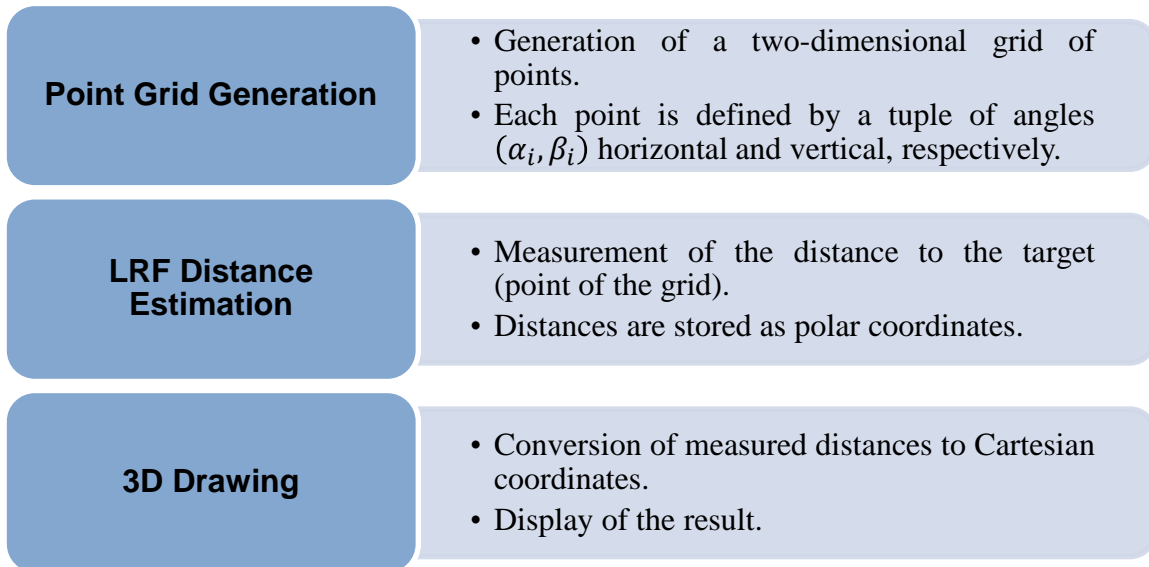
4.1.1. Assumptions

This system works under the following assumptions:

- The parallelism between the camera optical axis and the laser must hold, although as previously shown, there is room for some tolerance of alignment errors.
- The laser is always at a known position in relation to the camera (for this prototype the laser is below the camera and aligned with the center of the optical axis).
- The laser dot is always expected to be the brightest element detected.
- Boxes (used as a representation of pallets in the target environment) are loaded sequentially.
- There are no gaps between boxes.

4.1.2. Software Implementation

The implementation of this system is modularized and consists of three main components:



These modules cooperate to execute the following activity diagram (Figure 4.1) which shows the overall functionality of the system.

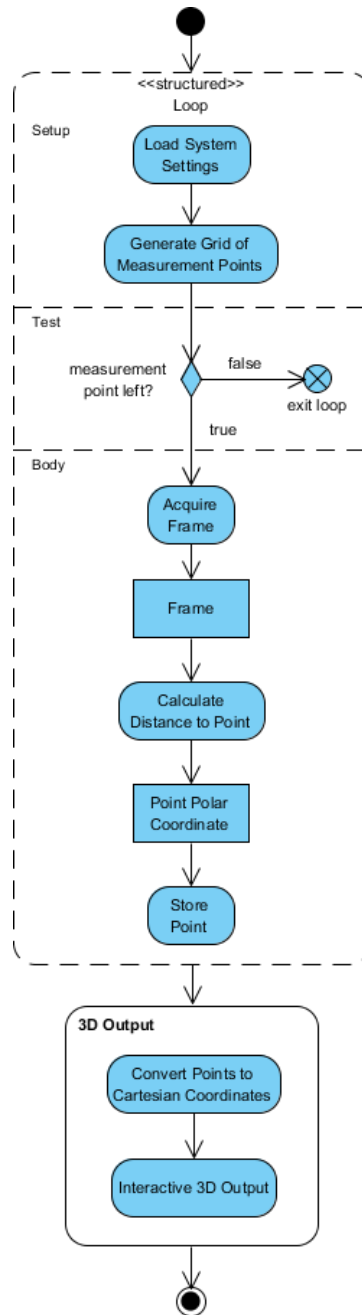


Figure 4.1: LRF system activity diagram

Each module was designed and implemented independently which means that they can operate in different machines (in cases where distributed computing is more advantageous) using a communication protocol that properly provides the required data input for each module.

A grid of points (α_i, β_i) is generated in order to cover the measured space. Each point represents a position where the system will move and make a measurement.

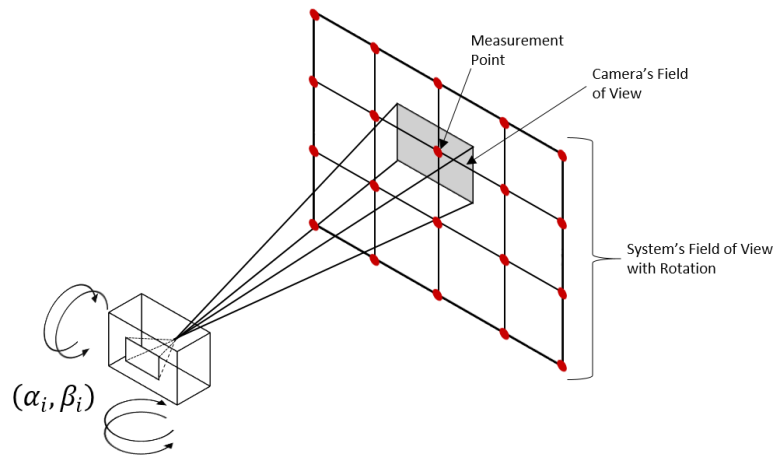


Figure 4.2: Grid Generation and Measurement Procedure

This grid is generated by dividing the total system's field of view (considering the rotation) by a number of user defined columns and rows, with the measurement points being the intersection between both.

Given the predictability of the position of the laser due to the geometry of the system (as previously shown in Figure 3.17), it is possible to infer the region in the frame that the laser dot will be. This allows the frame to be processed only on the region of interest.

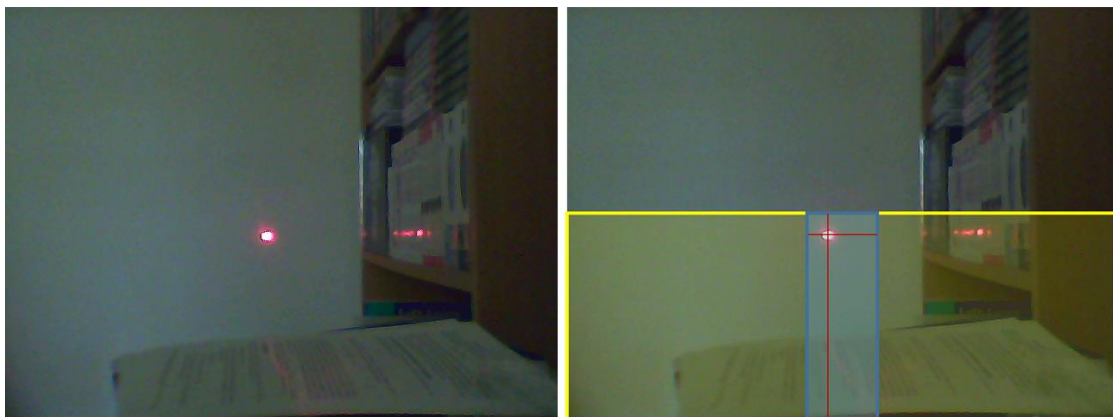


Figure 4.3: Original frame (left) and Frame with Regions of Interest (right)

These regions of interest vary with the position of the laser in relation to the camera. For this prototype, the laser is below the camera which means that it is expected that the laser dot will always be detected in the bottom half of the frame. This can be seen by the yellow region which is the first being selected as seen in Figure 4.3, after this selection a smaller region can be extrapolated since the laser is aligned with the center of the optical axis, therefore the laser will be located in a section of the middle of the frame (blue region).

This region can vary in width if there is an alignment error, however it is expected that this region is much smaller than the whole half frame (yellow region).

After this process, the smaller selected region is processed in order to find the brightest pixel, which corresponds to the laser dot, and retrieve its vertical distance to the center h_i (measured in pixels) as necessary for the Equation (3.22). This detection process is exemplified by the activity diagram in Figure 4.4.

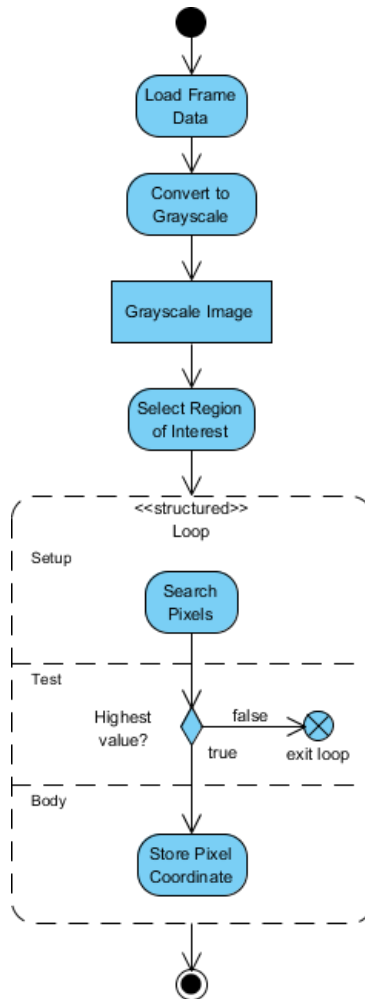


Figure 4.4: LRF detection algorithm

Based on the measurement angles (α_i, β_i) from the grid and their associated measured distances (in polar coordinates), the drawing module converts the input to Cartesian coordinates (correcting the center of the axis, otherwise the representation would be upside down since the original axis has its center on the camera system). The drawing is done from left to right with each point's closest neighbors forming a rectangle section that is drawn, and at the same time its volume is calculated. Each section is compared with the next one

and if their position (along one or more axis planes) is within a small configurable threshold value, then the algorithm interprets both as part of the same surface, making adjustments so that their positions match while drawing. The total volume is the sum of each section's volume and the result is then displayed to the user (as shown in Figure 4.5).

This volume estimation method is similar to the isolated points method presented by Zhang et al. [16], but assuming each point as the top of a parallelepiped given the cubic nature of the measured boxes. However, this system 3D reconstruction functionality was initially developed without being aware of previous approaches.

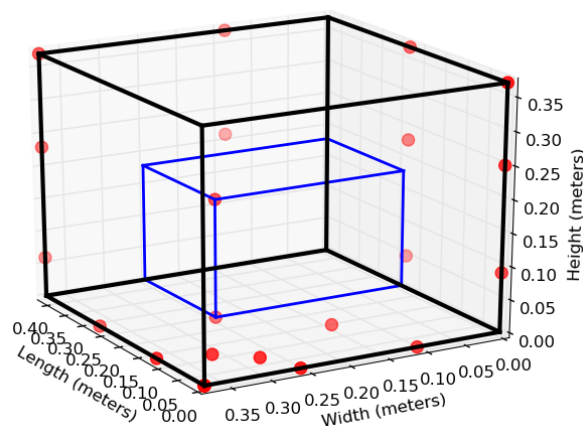


Figure 4.5: Example output with input data (red dots)

4.2. Computer Stereo Vision Prototype

The second prototype (developed under the supervision of Professor A. Manuel de Oliveira Duarte) uses stereo vision for distance measurements.

4.2.1. Assumptions

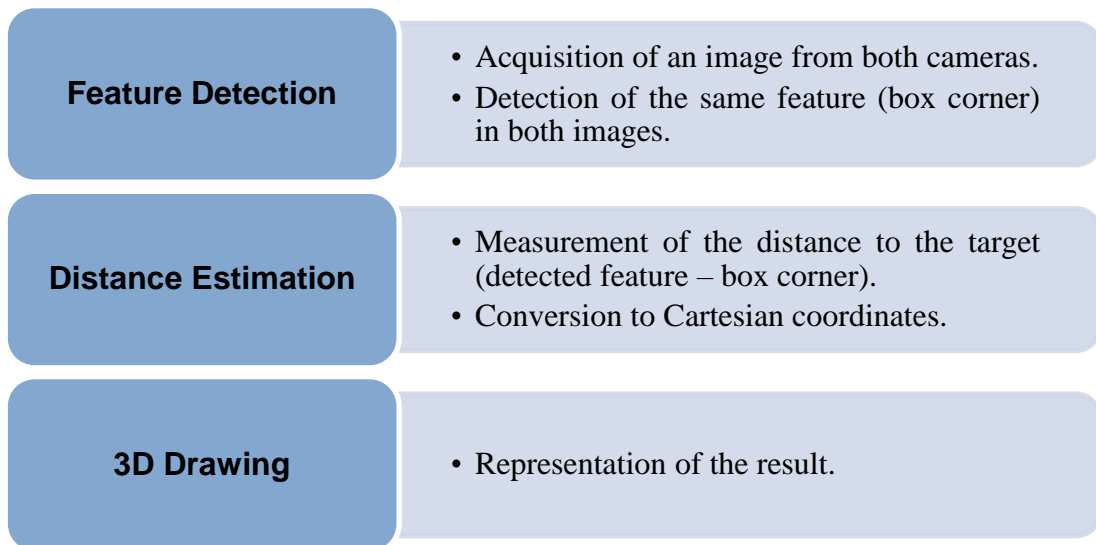
This system works under the following assumptions:

- The parallelism between both camera's optical axis must be maintained (with some tolerance if curve fitting is used).
- Boxes must have a certain degree of contrasting colors to the background (this is a software implementation limitation that will be detailed further along).
- The visible face of the box is always normal to the camera's optical axis.

- Boxes are non-stackable (this is a limitation to simulate the fact that in the target environment pallets are considered to be non-stackable) and are loaded sequentially and packed next to each other (no gaps along the container's length).

4.2.2. Software Implementation

The implementation of this system consists of three main modules:



These modules execute the following activity diagram shown in Figure 4.6.

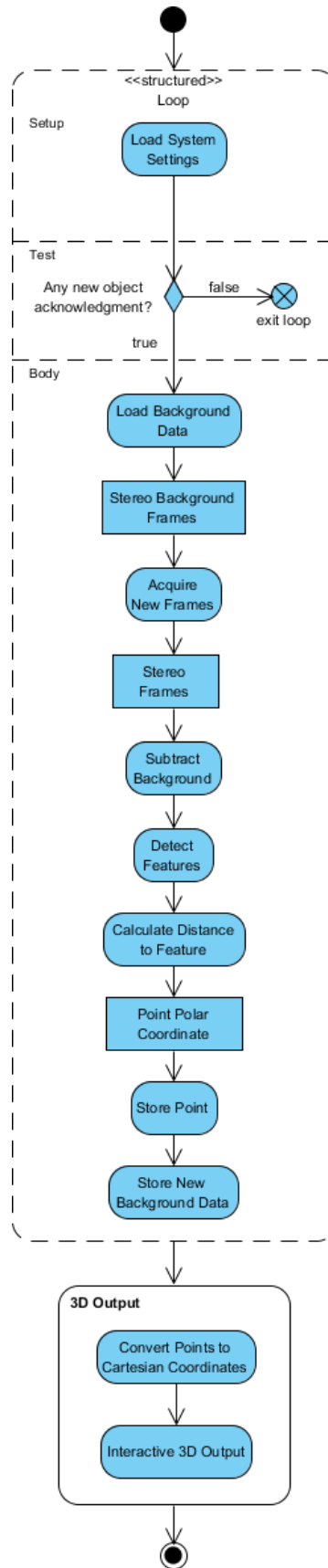


Figure 4.6: Stereo System Activity Diagram

The process begins by subtracting the background (frame without a box) from the active frame (frame with a box), which isolates the target object (box) making it easier to detect its relevant features (corners). This starts by taking a frame of the background and then a frame with the object as exemplified in Figure 4.7.



Figure 4.7: Frame with object (left) and background frame (right)

The subtraction of both frames is then smoothed with a filter to further eliminate rogue elements that might still be visible (Figure 4.8).

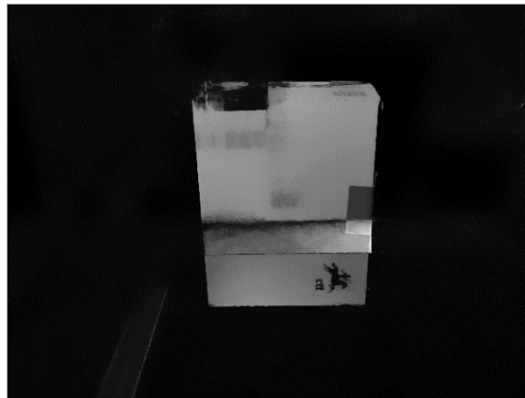


Figure 4.8: Result after subtraction and smoothing

With this result a threshold is applied in order to create a binary mask with the object's shape. Morphological operations of closing and opening are applied to filter smaller isolated concentrations of pixels (Figure 4.9).

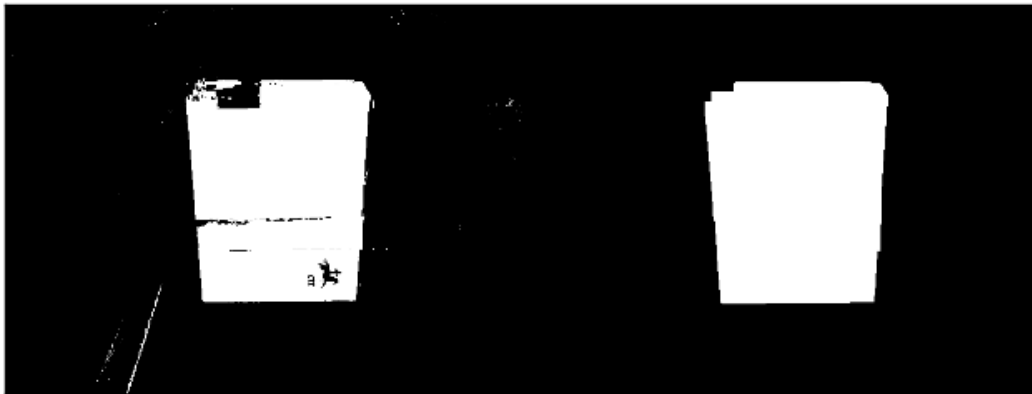


Figure 4.9: After threshold (left) and after morphological operations (right) results

Finally, this mask is multiplied to the original frame with the object, and the corners are determined (Figure 4.10).

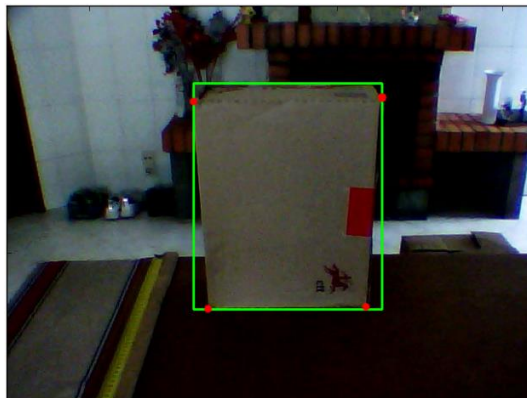


Figure 4.10: Detected object (green rectangle) and detected corners (red)

Detecting the corners of the box can be done using known corner detectors algorithms (such as Harris Corner Detection), however such algorithms are often complex and take considerable amount of computational resources. To mitigate this issue, a simple method was implemented based on the geometry of the system and the foreseeable shape of the box as seen by the camera.

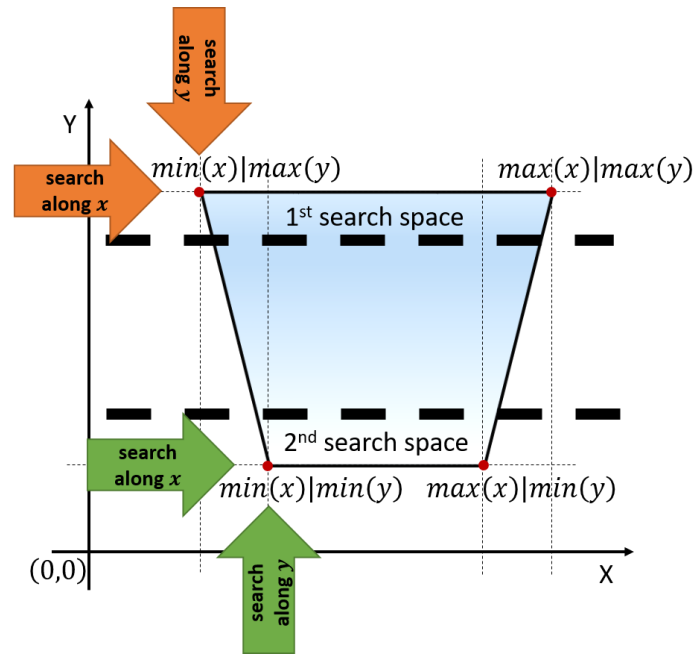


Figure 4.11: Geometric shape as viewed by the system

The system's optical axis must be parallel, however, they don't need to be parallel to the container's *Length x Width* plane, in this prototype they were positioned with an angle that allows to capture as much space possible of the container's space. Due to this geometry, the cameras capture the frontal face of the box as a trapezoid (as seen in Figure 4.11). The resulting image (which can be interpreted as a 2-dimensional array since it is an intensity image when in grayscale) after the application of the mask (as previously shown in Figure 4.9) assumes the value 0 for regions where the background was removed, and intensity values correspondent to the isolated object. This image is searched first starting on the top (represented in Figure 4.11 by 1st search space): for each x , y is searched for the first value that is different than 0 ($min(x)$), and the last value different from 0 ($max(x)$), these are the positions of the top corners. The same procedure is executed for the 2nd search space.

This searching procedure is easy to implement and requires very little running time, however it is not as robust as formal corner detector algorithms, since it requires the masks to be properly detected, therefore this solution is only useful for systems that have limited processing power or specific task problems like the ones being tested with this prototype. Either way, given the modularity of the system, a formal corner detector algorithm can be integrated into the system for more complex detection problems.

After the detection of the corners on both frames (left and right), disparity can be calculated and the distance to the object estimated using Equation (3.25). After converting each distance to Cartesian coordinates (using the geometry of the system as shown in Figure 3.20) a 3D representation is generated (Figure 4.12).

The volume estimation is still inspired by the isolated points method presented by Zhang et al. [16] using parallelepipeds, which makes it faster since it is only processing the corner's coordinates and the container's dimensions.

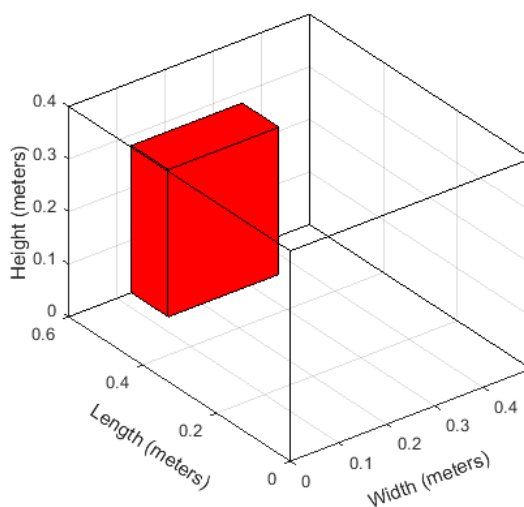


Figure 4.12: Stereo system output

It is worth to note that feature detection can be simplified by reducing the number of corners needed to reconstruct the cargo. This is due to the target environment where this system is designed to operate (typically a truck's cargo space or a container) being usually loaded with non-stackable cargo according to a specific layout. Given that the system knows the container dimensions and the shape of the layout being used, it is only necessary to detect the interior corners of the cargo, i.e., the ones not adjacent to the container's walls, since those can be inferred from the geometry. The developed system uses this approach to reduce the processing time required to produce a 3D output.

5. Heuristic for Volume Loading

This section will detail the implementation of the developed heuristic for volume assignment under the two considered scenarios (previously detailed in 3.1.1):

- Pallet assembling
- Cargo loading

5.1. Assumptions

The model implemented by this heuristic works under the following assumptions:

- Boxes, pallets and cargo containers are cuboid.
- Boxes have similar form factors to pallets.
- Boxes are placed in a parallel horizontal plane to the pallet.
- Boxes and pallets have only two possible rotation degrees (0° or 90°). In the case of boxes, one of the faces with the biggest area needs to always be facing downwards. Pallets are non-stackable and rotation is dependent on their position on the loading layout.

5.2. Implementation

The implemented heuristic is composed of two complementary parts. The first part deals with volume assignment and placement in a 3D environment (pallet); The second part deals with non-stackable volume assignment into containers (truck's cargo space).

5.2.1. Pallet Assembling

Packing patterns are bound by the weight limit and orientation constraints (only two horizontal rotation degrees are permitted).

Due to its heuristic nature, this process will find practical local solutions for the loading of boxes into pallets.

This heuristic was implemented to work with a weakly heterogeneous set of cargo, i.e., only a few cargo types are considered for the execution. This translates the fact that since

the palletization of goods is used in logistics activities, boxes tend to have similar dimensions (form factors) that better fit the pallet, allowing for a maximization of its volume usage.

Each box is assumed to always have one of the faces with the biggest area facing down, allowing only rotations along the vertical axis.

Figure 5.1 shows the flow of steps followed by the execution and assignment process.

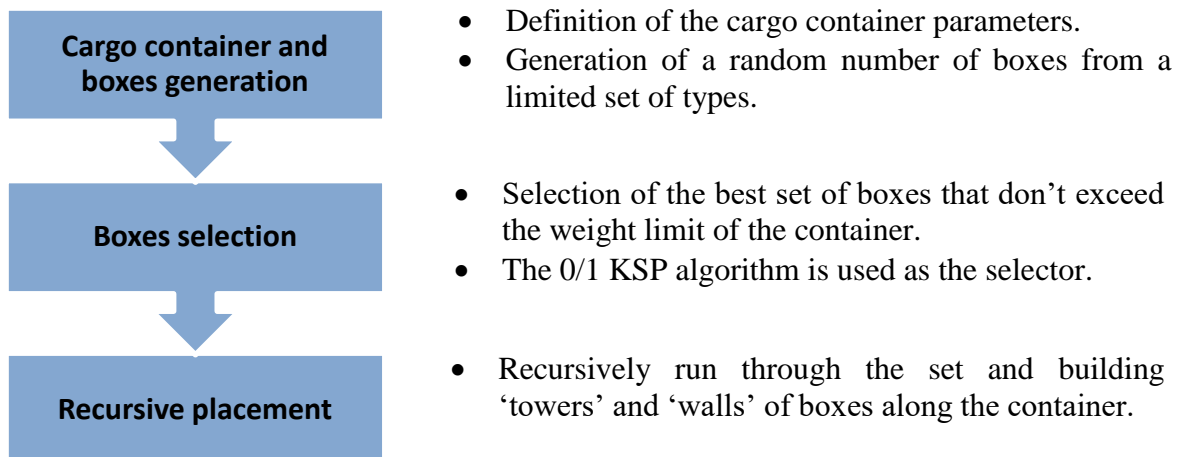


Figure 5.1: Volume assignment implementation flow

A random number of boxes are generated from a limited set of types with random weights associated to each individual box. Boxes with bigger contact area with the container floor are always heavier than smaller ones. This is important to create a gradient of weight during placement.

The generated set of boxes is then passed through a selection process using the 0/1 KSP algorithm that maximizes the sum of volume of the selected boxes while keeping the sum of weights equal or under the pallet container defined limit.

The selected boxes are placed according to the diagram represented by Figure 5.2.

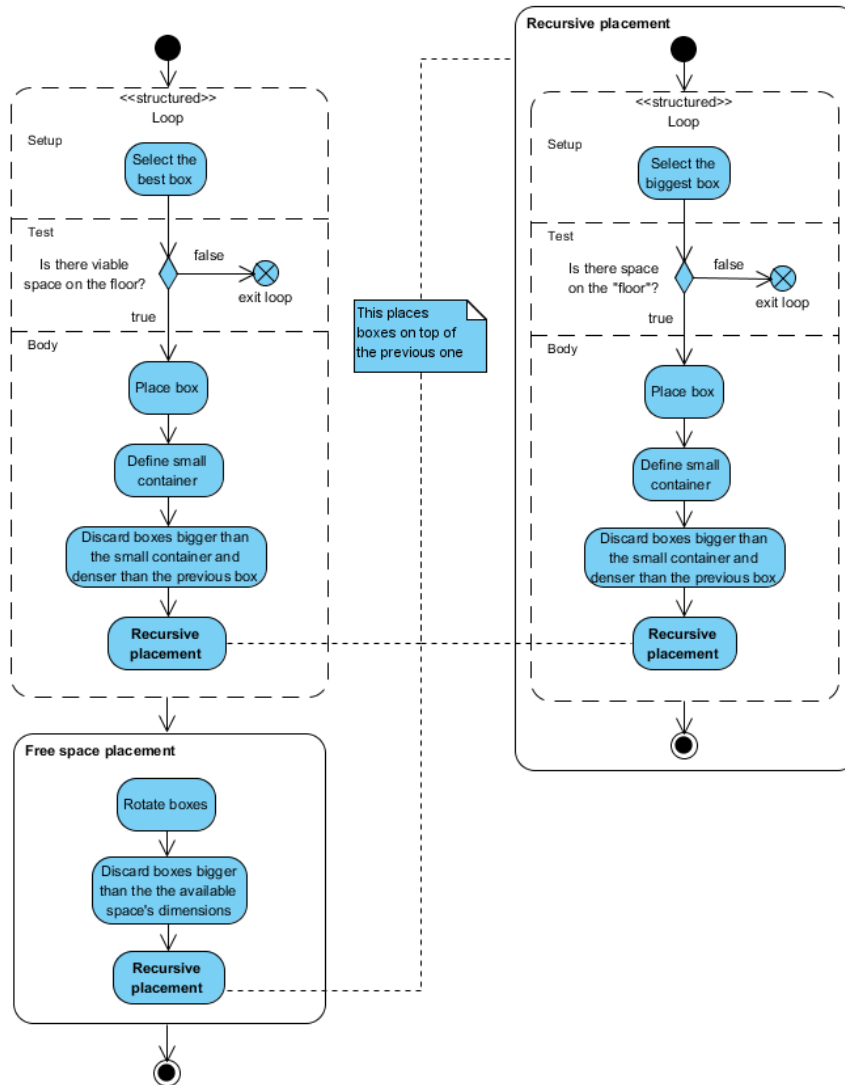


Figure 5.2: Volume placement implementation

The heuristic starts by selecting a box with the biggest volume and whose type has the most number of copies (this being the criteria for the selection of boxes to be placed on the container’s floor).

After each box placed on the floor, the heuristic defines the volume on top of the box as a smaller container and discard all boxes bigger than its the dimensions and have higher density than the previous box, creating a small subset of boxes that fit inside this container.

For vertical placement, the heuristic uses a recursive method that always selects the biggest box available (in terms of bottom contact area) from the previous subset of boxes, recalling itself for placing new boxes on top of the previously placed one and repeating the process for adjacent spaces, narrowing the subset of boxes even further by continuously discarding denser and bigger boxes than the dimensions of each smaller container created.

After the placement of all possible boxes, the remaining that haven't been placed are rotated 90° and used to fill free spaces left were they might fit by using the same recursive approach.

Boxes with higher weight are always placed first with the consequent boxes on top always being lighter (lower weight and density) than the previously placed ones. This guarantees the vertical stability of each ascending tower by having the vertical center of mass in the middle or below the tower's height. The vertical center of mass of the resulting packing pattern is approximated by calculating the average center of mass from all the towers.

5.2.1.1. Tower and Layer Building

The heuristic places boxes by making 'towers' that together form layers (or 'walls') of boxes. This placement strategy is inspired on Moura and Oliveira [24] and Gehring and Bortfeldt [23]. The following sequence of images illustrates the placement process.

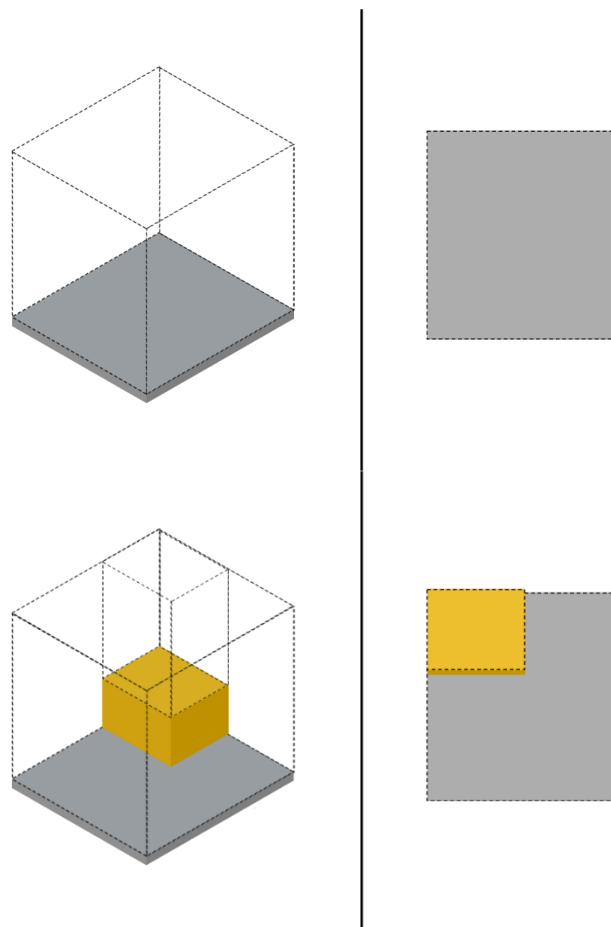


Figure 5.3: Placement strategy - 1

The first box placed defines the maximum width of the layer and initiates its construction. After the placement of a box (Figure 5.3), the heuristic regards the top volume as a smaller container and proceeds to place the biggest box available (in terms of contact surface area) in the space. This process is done recursively until there is no more viable vertical space.

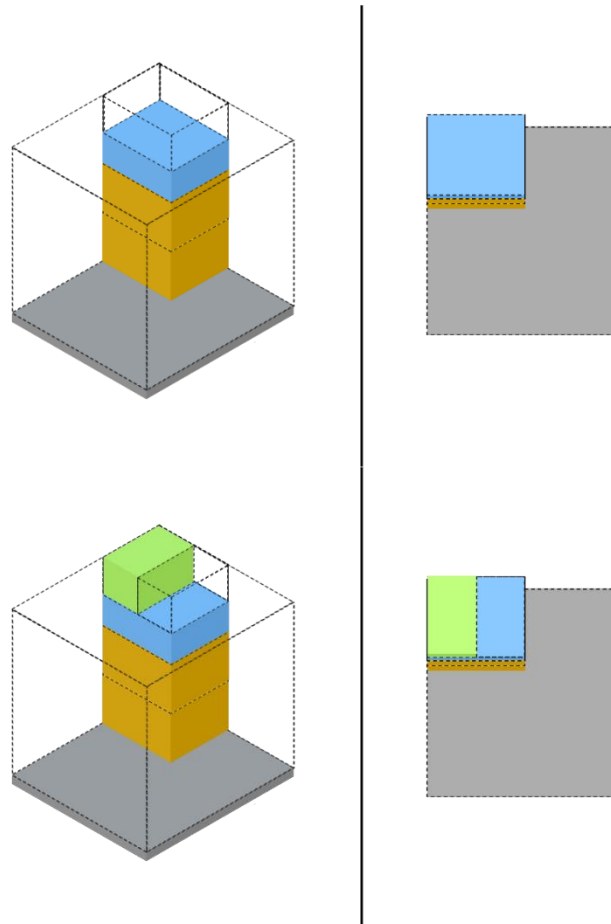


Figure 5.4: Placement strategy - 2

Since this is a recursive process, each placed box is a layer even if it is on top of another box, which in that case is referred to as an intern layer of the smaller container (Figure 5.4). After the box placement, the heuristic verifies if there is space for a new intern layer and places a new box.

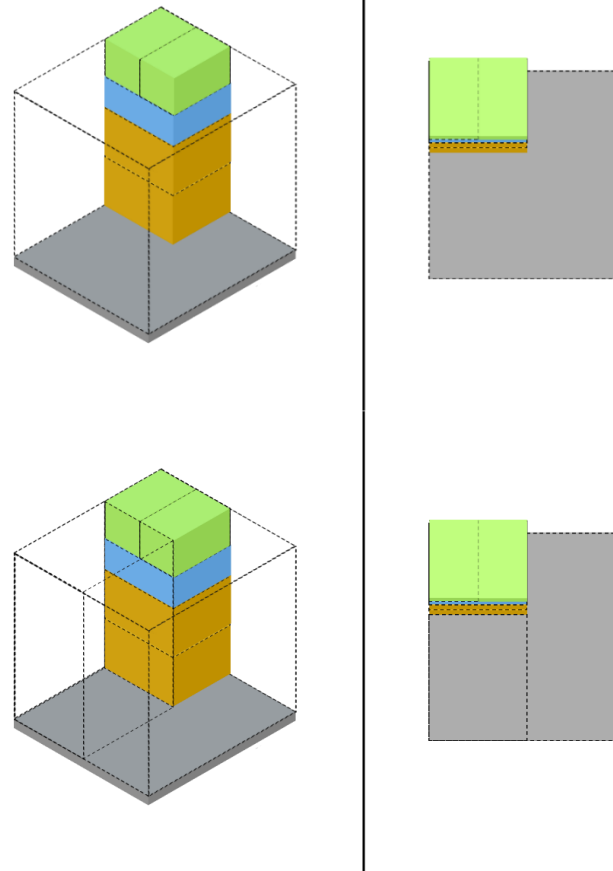


Figure 5.5: Placement strategy - 3

When the 'tower' is completed, the construction of the layer is resumed (Figure 5.5) by placing a new box. This process is repeated ('tower' building recursion) until no more boxes (with the current orientation) can be placed, tagging the remaining space as free (Figure 5.6).

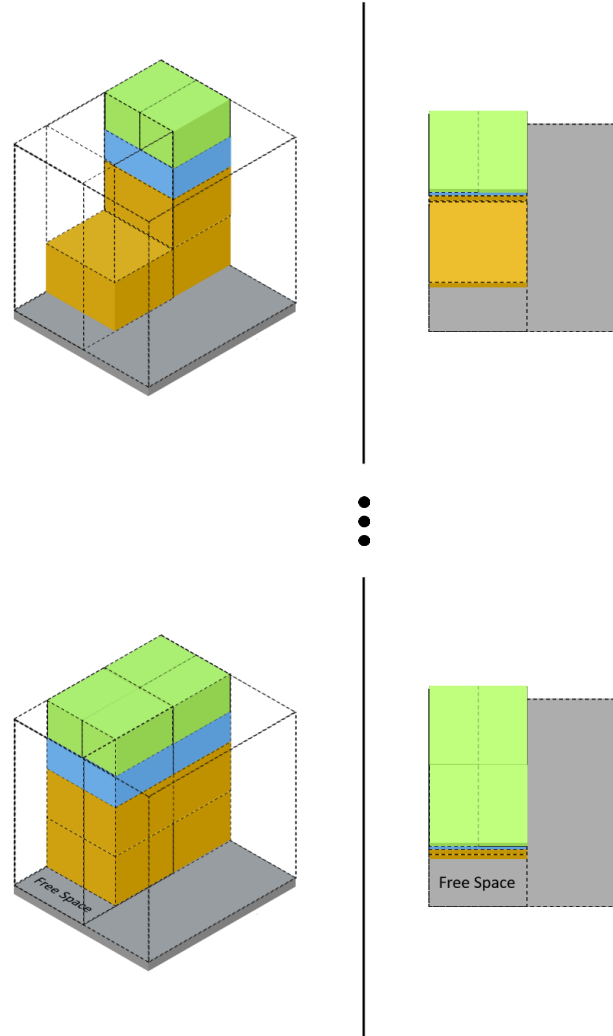


Figure 5.6: Placement strategy - 4

As it was the case with the intern layer on top of a box, if there is still space left in the container, a new layer is started by placing a new box (Figure 5.7) and repeating the whole process until the container is full or no more layers can be created due to boxes not fitting.

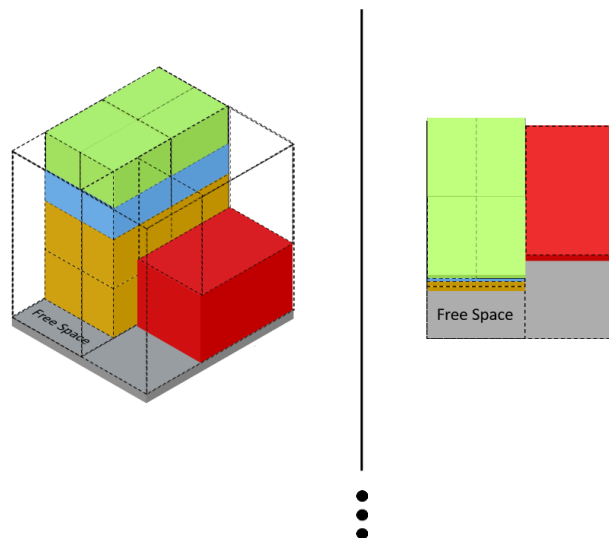


Figure 5.7: Placement strategy - 5

When no more layers can be created, the heuristic rotates by 90° all the boxes left and cycles through the tagged free spaces re-applying the recursive placement strategy as before.

This heuristic provides compact placement by prioritizing the selection of boxes that match the width of the layer, with the strongest point being the vertical placement ('tower' building).

The adoption of this strategy stems from the fact that it simplifies the placement by removing the vertical dimension and making it a 2D (area) assignment problem (as visualized by the top view on the previous figures). This means that boxes can be placed according to the area they occupy either on the floor or on top of another box and use the vertical dimension only for checking up if the box fits or not.

5.2.2. Cargo Loading

Loading patterns for pallets assigned to containers (such as a truck) are bounded by the complete-shipment (refer to constraints in 3.1.3), loading priorities (given by the logistic importance), weight limit and weight distribution constraints. This process is simplified to 2D since pallets are assumed to be non-stackable.

Two sub scenarios were considered for the generation of loading patterns:

1. The complete-shipment, loading priorities and weight limits constraints as hard constraints and the weight distribution as a soft constraint.

This translates a common scenario in transport and logistics companies where bundle's pallets often need to be loaded together to optimize the pickup/delivery route. In this case the weight is distributed the best as possible within each bundle. However, due to the uncertain nature of the weight of each bundle, this means that a uniform weight distribution is not always guaranteed.

2. The weight distribution and weight limits as hard constraints.

In this scenario the weight must be distributed uniformly across the container as best as possible. This means that bundles might need to be broken and pallets placed individually.

In order to simulate the process for the scenarios, bundles are generated with a random number pallets assigned and a logistic importance value calculated from the given function (previously detailed in 3.1.3). Since all bundles can be viewed as independent and unique, the 0/1 KSP algorithm is used to determine if a bundle should be selected or not. This allows to calculate which bundles maximize the sum of the value of the logistic importance while keeping the total number of pallets within the allowed capacity. This process is described by Figure 5.8.

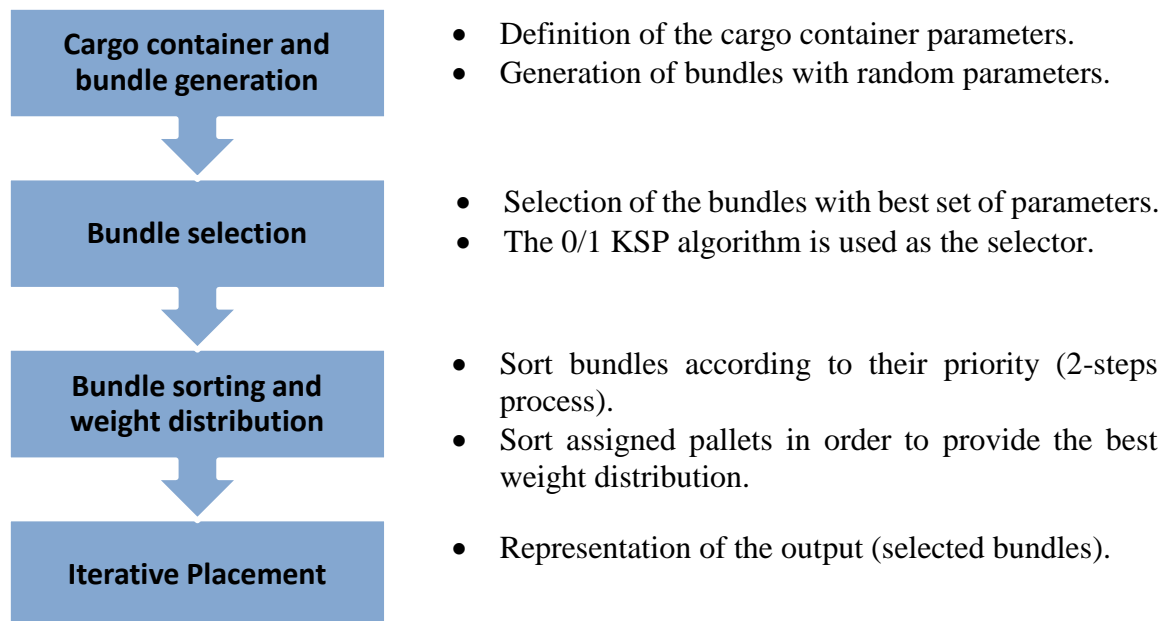


Figure 5.8: Bundle assignment implementation flow

The bundle selection is done in two steps:

1. Selection by weight (weight limit).

This step selects the bundles that ensure that the weight limit of the container is not violated.

2. Selection by logistic importance (priority).

This second step picks from the bundles previously selected, those that maximize the sum of value of logistic importance.

In cases where the container still has free space after the application of these two steps, the process is repeated using the bundles that were not selected along with the available weight and space left in the container from the first pass.

It is worth to note that the selection is done (using the 0/1 KSP algorithm) in the first step with the physical weight of each bundle as the *weight* and the logistic importance as the *value* (according to the nomenclature used in 3.1.6), with the maximum capacity of the container being its weight limit. In the second step, the number of pallets is the *weight* and the logistic importance is the *value*. In this case, the maximum capacity of the container is the maximum number of pallets permitted by the loading layout.

5.2.2.1. Weight Distribution

To ensure the container stability of the loading pattern, the cargo needs to be positioned in order to provide a weight distribution as uniform as possible along both the longitudinal and transversal dimensions.

A chessboard matrix (Figure 5.9) is implemented, for individual bundles, by positioning heavy and light weights alternatively. Each pair of pallets positioned transversally constitutes a container section.

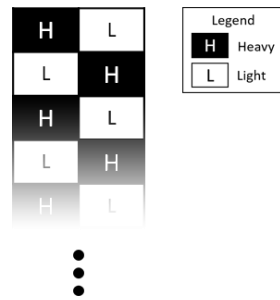


Figure 5.9: Chessboard weight matrix

This method of weight distribution guarantees the stability of the container, however, due to the unpredictable range of weights that can be part of a bundle, in larger distributions there is the possibility of multiple weights being very similar which would result in these weights being matched near the end of the distribution. These matching weights are instead positioned in the middle of the container (as exemplified by Figure 5.10), resulting in a better distribution.

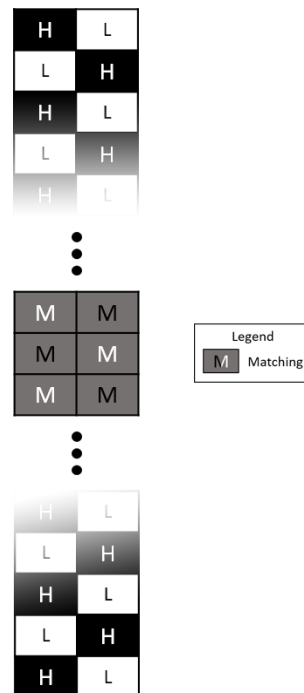


Figure 5.10: Chessboard weight matrix with matching cases

To evaluate the stability obtained by the chessboard weight matrix, the weight influence between adjacent sections is calculated using an approximation given by the following equation:

$$\Omega_i = \frac{w_{i-1}}{2} + w_i + \frac{w_{i+1}}{2} \quad (5.1)$$

For the transversal section i , its real weight Ω_i (weight considering the influence from the adjacent sections) is calculated with w_i as the weight of the current transversal section (with both pallet's weight summed together), w_{i-1} as the weight of the previous adjacent section and w_{i+1} as the weight of the next adjacent section. For the extremity sections, it is considered as if the container formed a ring, i.e., the value for the first section uses the influence of the next section and the last section, the same principle applies to the last section. This allows to consider the influence of the extremity sections without them introducing a constant value proportional to their weight.

This approximation tries to incorporate the influence of weight between sections in a simplistic way, in order to have a rough idea of the stability of the distribution.

The instability factor of the distribution is then given by:

$$I = \sum_{i=1}^N \left| \Omega_i - \frac{1}{N} \sum_{j=1}^N \Omega_j \right| \quad (5.2)$$

Where N is the number of sections of the container.

The value of the instability factor is used as a comparison tool between weight distributions, where a smaller I translates a better distribution.

6. Results and Discussion

In this section the results obtained with the heuristic for volume assignment and sensing systems will be presented and discussed.

6.1. Heuristic

Executions of this heuristic were conducted to evaluate the output solutions provided in both considered scenarios (pallet assembling and cargo loading). These tests reflect possible real world cases where the heuristic would operate.

6.1.1. Data Structure Example

To validate the representation capabilities of the data structure developed in 3.1.2, a simple example will be demonstrated.

A box type represented by Figure 6.1 and its properties in Table 6.1 is considered.

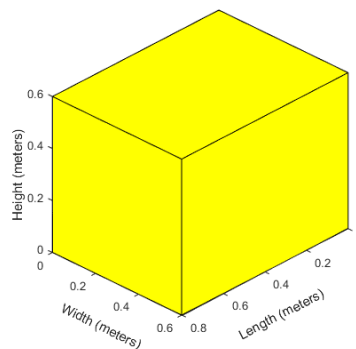


Figure 6.1: Box type

Table 6.1: Box properties

Dimensions (meters) ($l \times w \times h$)	Weight (weight units)
0.8 x 0.6 x 0.6	1

The box is then paced into a pallet, with the dimensions of 1.2 x 0.8 x 1 meters, as shown in Figure 6.2.

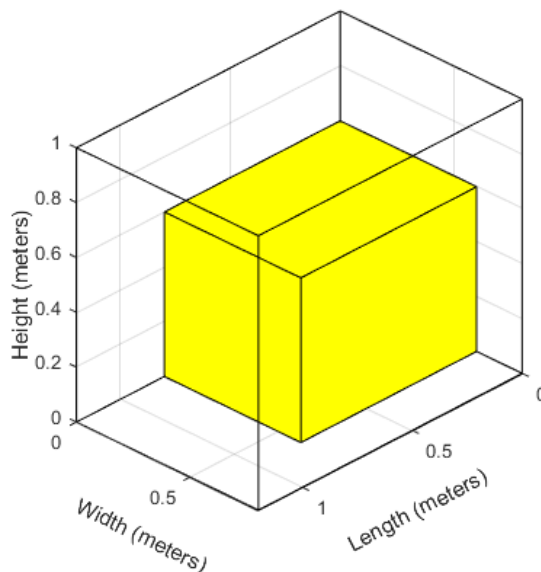


Figure 6.2: Box placement into a pallet

The heuristic produces a placement list (payload) for this pallet as shown in Table 6.2.

Table 6.2: Placement list (payload)

x (meters)	y (meters)	z (meters)	box type	weight (weight units)	azimuthal angle (binary value)
0	0	0	1	1	0

This pallet is then assigned to a bundle and results in the following data shown in Table 6.3.

Table 6.3: Bundle

pallets (*)	weight (weight units)	volume (m^3)	logistic importance (**)
1	1	0.288	0,0794

*structure array: the number represents the quantity of pallets on each bundle. The structure also contains information about the weight, volume, box types and a placement list like the one presented in Table 6.2 for each pallet.

**the bundle individual priority is not directly stored but it is used to calculate the value of logistic importance along with other parameters detailed in 3.1.3.

The bundle is then assigned to a cargo container and the pallet placed inside as shown in Figure 6.3. The cargo container has the dimensions of 3.2 x 2.4 x 1.5 meters and a capacity of 8 pallets.

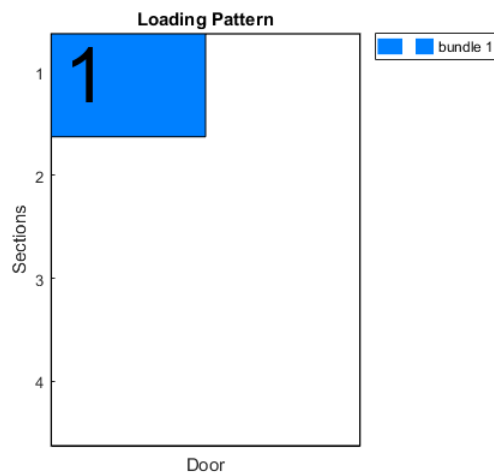


Figure 6.3: 2D cargo placement

Once again, the heuristic produces a layout placement list (cargo payload) as shown in Table 6.4.

Table 6.4: Layout placement list (cargo payload)

x (meters)	y (meters)	bundle number (*)	azimuthal angle (binary value)	pallet weight (weight units)
0	0	1	0	1

*the bundle number represents the bundle with the row number of the table being the pallet.

Using the information from Table 6.4 (cargo payload) and Table 6.2 (payload) along with the associated entities, it is possible to produce a 3D representation of the loaded cargo container as shown in Figure 6.4.

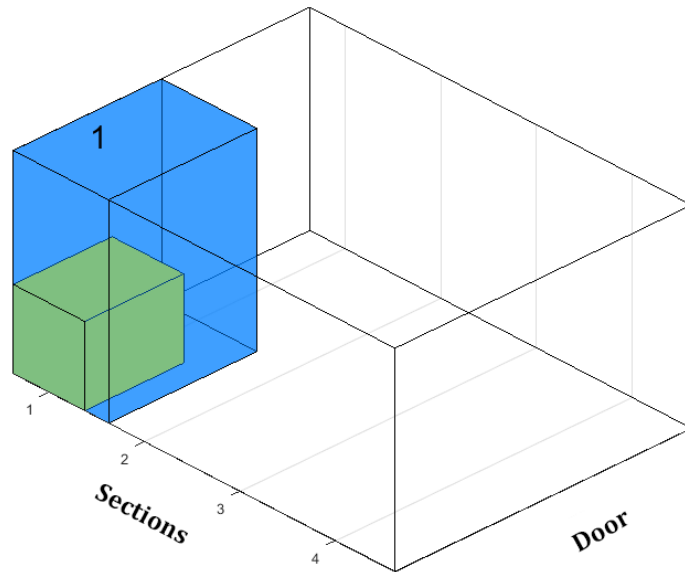


Figure 6.4: 3D cargo placement

6.1.2. Pallet Assembling

In this scenario, it is considered that the problem faced is of the output maximization type, where there is only one container (pallet) whose volume needs to be maximized while having a set of available boxes larger than the volume of the container.

To produce packing patterns, a container of 1.2m length, 0.8m width and 1m of height was defined, allowing a maximum of 20 weight units (these are abstract units that represent weight).

First, a simple scenario is constructed using only one type of box with a well-defined form factor as represented in Figure 6.5.

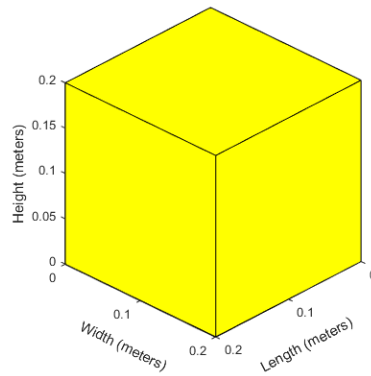


Figure 6.5: 1-box type

This box type has the following characteristics as shown in Table 6.5.

Table 6.5: Execution parameters (input data)

Number of boxes	Dimensions (meters) ($l \times w \times h$)	Weight range (weight units) ($min-max$)
10	0.2 x 0.2 x 0.2	0.15-0.5

Ten cubic boxes were generated with random weights picked from a range of 0.15 to 0.5 weight units.

Applying the developed heuristic to this input data produced the following packing pattern (Figure 6.6).

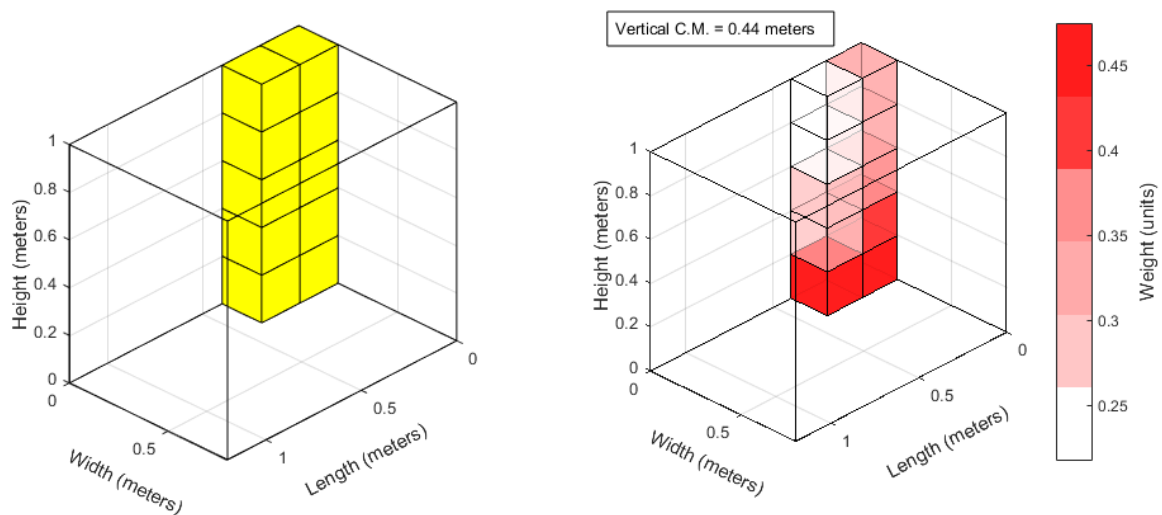


Figure 6.6: Packing pattern (1-box type)

According to the placement strategy implemented (explained in 5.2.1) the first box placed defined the layer width. Boxes are placed on top of the first one using a recursive

placement strategy until no more boxes can fit, at this point the heuristic continues with the construction of the layer by positioning the next box on the base and repeating the recursive top placement.

It is also worth to note that the vertical center of mass (0.44m) is below the middle of the pallet height, resulting in a vertically stable packing pattern.

Another test scenario considered is to generate enough boxes of this type to completely fill the container. The resulting packing pattern is shown in Figure 6.7.

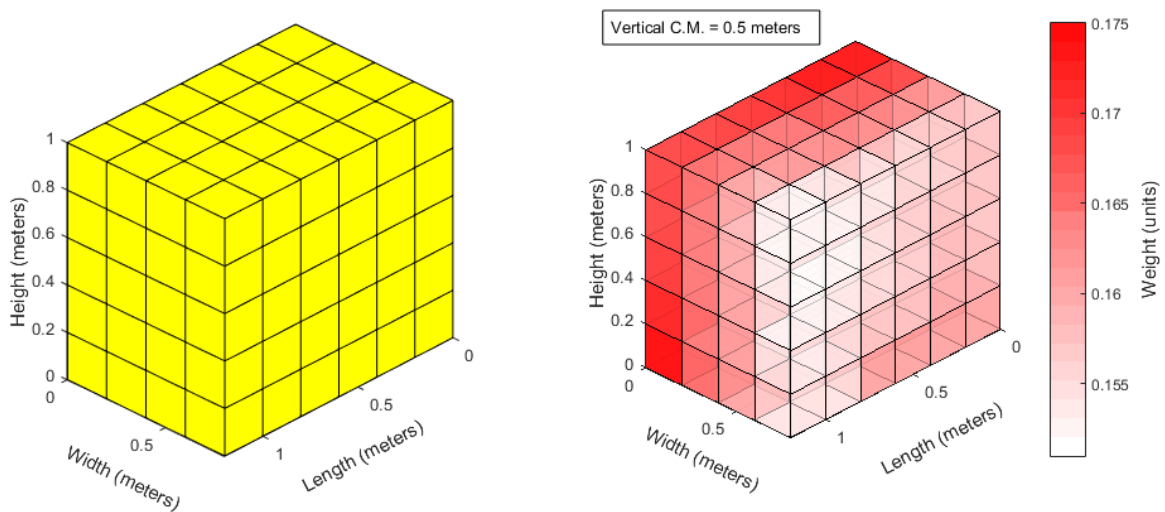


Figure 6.7: Packing pattern (full 1-box type)

This scenario now shows the multiple layers constructed until the container was completely full (or no more boxes could be fit in the available spaces, which might happen in cases where there are more boxes with different box types). The same conclusions about the vertical stability can be taken as in the previous test scenario.

A standard scenario is now considered using a weakly heterogeneous set of cargo. Three box types were defined and are represented as shown in Figure 6.8.

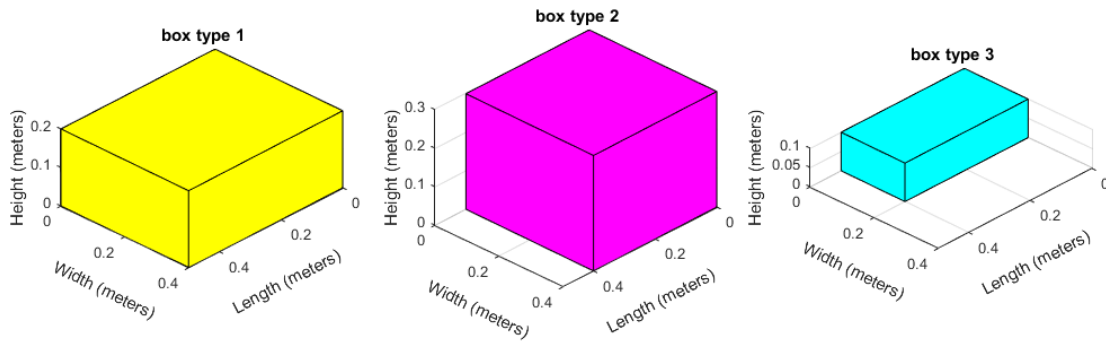


Figure 6.8: 3-box types

These boxes have the following characteristics as shown in Table 6.6.

Table 6.6: Box types characteristics

Box type	Dimensions (meters) ($l \times w \times h$)	Weight range (weight units) ($min-max$)
1	0.5 x 0.4 x 0.2	0.4-0.5
2	0.4 x 0.4 x 0.3	0.3-0.4
3	0.4 x 0.2 x 0.1	0.009-0.01

The number of boxes of each type is generated randomly (from a range of 6 to 15) as well as the weight associated (from within the defined weight range).

An example of a packing pattern obtained with this heuristic is shown in Figure 6.9. A total of 35 boxes were selected by the 0/1 KSP algorithm as having a sum of weight less or equal to the maximum supported by the pallet. Then the heuristic selected from these boxes according to the placement strategy to generate the packing pattern. The input data along with the outputs for this example can be seen at Annex C.i as well as further examples.

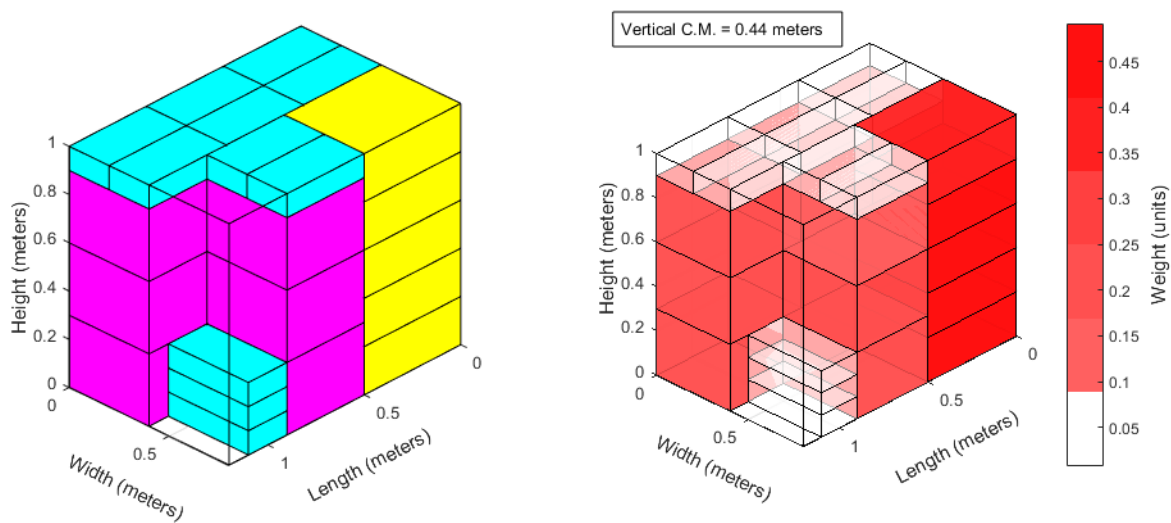


Figure 6.9: Packing pattern (3-box types)

This packing pattern obtained the following results shown in Table 6.7.

Table 6.7: Packing pattern statistics (3-box-types)

Container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)	Number of boxes used	Number of boxes left out	Box usage (%)
0.96	0.86	90	6.86	28	7	80

This is the typical scenario that the heuristic was developed for: having a weakly heterogeneous set of cargo with form factors similar to the pallet.

The high percentage of volume usage reflect the placement strategy employed by this heuristic that favors compact solutions, however, given the randomness associated with the generation of the number of boxes, the percentages can vary slightly.

Once again, since boxes with higher weight are always placed below, this packing pattern is vertically stable (vertical center of mass is located below the middle of the pallet’s height), i.e., boxes won’t fall after being placed.

To test the limits of this heuristic, a scenario with a strong heterogeneous set of cargo was constructed, generating 30 boxes with random dimensions ranging from 0.1 to 0.5 meters (each dimension) and weights equal to the bottom surface area of each box.

The heuristic produced the following solution (Figure 6.10).

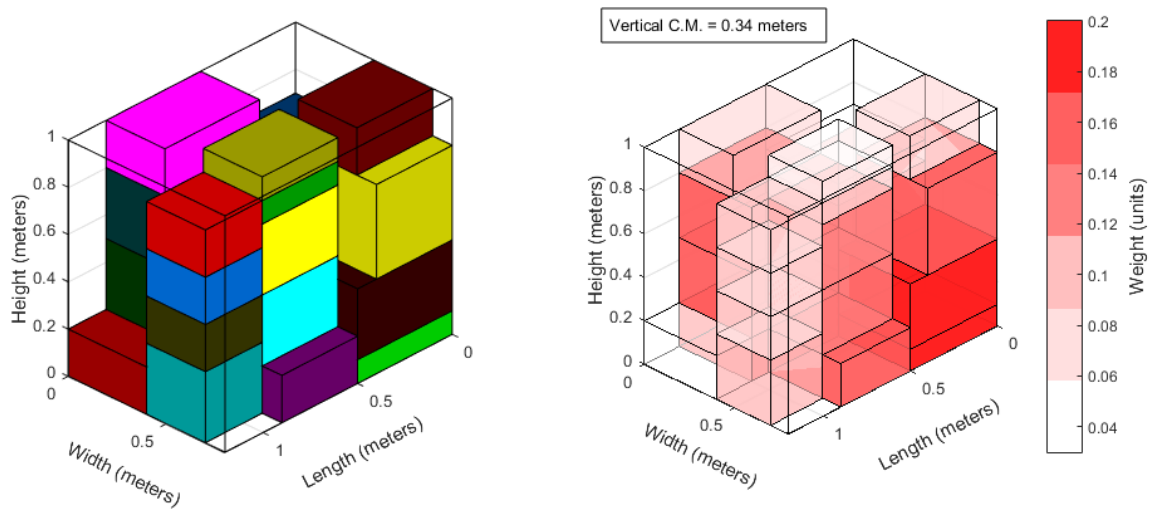


Figure 6.10: Packing pattern (30-box types)

This packing pattern obtained the results shown in Table 6.8.

Table 6.8: Packing pattern statistic (30-box types)

Container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)	Number of boxes used	Number of boxes left out	Box usage (%)
0.96	0.723	75.31	2.31	21	9	70

This extreme scenario shows that the placement strategy of the heuristic is still valid, which can be seen by the two layers and towers constructed and vertical stability, however, since boxes now have a wide variety of form factors, placement is not as compact as the previous cases.

As it is, the heuristic produces satisfactory results when the assumptions of the developed model are met, however, an improvement on the placement strategy would be to construct the layers along the horizontal plane while keeping the tower building process. This would allow to better fill the container floor while pushing the free spaces to the top of the container, making it also easier to implement horizontal weight distribution.

While these test scenarios were conducted only for one pallet, the heuristic is designed to work with multiple pallets constructing solutions sequentially while there are pallets and boxes available.

6.1.3. Cargo Loading

This scenario will translate a case where the operator has only one truck available, and the number of bundles to be loaded exceeds the capacity of the truck.

As an example execution, 10 bundles were generated with a random number of pallets assigned from a range of 1 to 10. Packing patterns for each pallet (which have the dimensions of 1.2 x 0.8 x 2.0 meters) use the box types described in Figure 6.11 and Table 6.9.

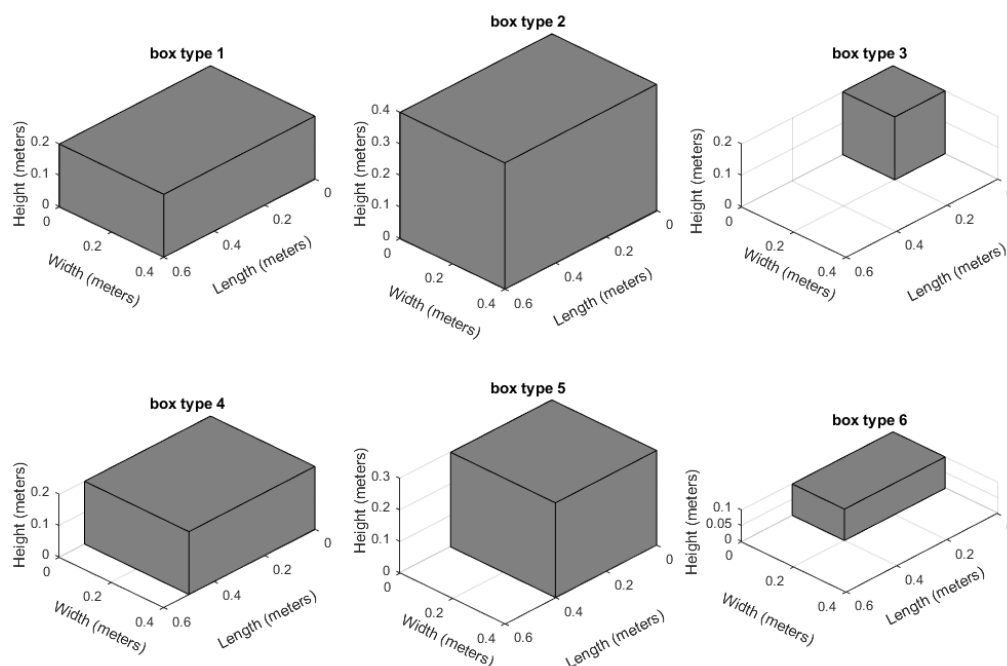


Figure 6.11: 6-box types

Table 6.9: 6-box types characteristics

Box type	Dimensions (meters) ($l \times w \times h$)	Weight range (weight units) ($min-max$)
1	0.6 x 0.4 x 0.2	0.09-0.3
2	0.6 x 0.4 x 0.4	0.35-0.5
3	0.2 x 0.2 x 0.2	0.005-0.01
4	0.5 x 0.4 x 0.2	0.08-0.29
5	0.4 x 0.4 x 0.3	0.09-0.3
6	0.4 x 0.2 x 0.1	0.005-0.01

For each pallet, the generation algorithm picks 3 of the 6 box types randomly (with the possibility of repetitions) and instantiates 20 boxes for each selected type. The instantiated boxes are then used by the heuristic to produce a packing pattern per pallet. This generation strategy is done in order to obtain potentially unique packing patterns.

The logistic importance of each bundle is calculated according to the Equation (3.17) (as detailed in 3.1.3) with the coefficients being assumed as $\alpha = 0.5$, $\beta = 0.1$, $\gamma = 0.4$ and $\delta = 0$. These coefficients favor the selection of bundles with a higher number of assigned pallets and higher potential occupation of the container volume. The bundle priority μ_r is generated randomly and the other elements are calculated per pallet according to the statistics of the generated packing pattern.

The container has dimensions of 13.6 x 2.45 x 2.5 meters, a capacity of 33 pallets (as defined by the chosen layout) and a maximum weight of 320 weight units.

Additional output data from this execution can be seen at Annex C.ii. A practical case study using this heuristic can also be seen at Annex D.

Table 6.10 shows the generated bundles and their characteristics.

Table 6.10: Generated bundles

bundle	pallets	weight (weight units)	volume (m^3)	logistic importance
1	2	15,94	3,84	0,0944
2	3	23,30	5,28	0,0732
3	5	37,39	9,46	0,1636
4	5	41,58	9,28	0,1538
5	6	50,79	11,36	0,2130
6	6	44,79	11,02	0,2425
7	7	56,12	13,12	0,2474
8	7	55,34	13,20	0,2499
9	7	56,28	12,86	0,1984
10	9	69,18	16,38	0,2292

The heuristic then selected the bundles which maximize the value of logistic importance while keeping the weight and capacity equal or below the maximums supported.

The selected bundles are shown in Table 6.11.

Table 6.11: Selected bundles

bundle (previous id)	pallets	weight (weight units)	volume (m^3)	logistic importance
---------------------------------	----------------	----------------------------------	--------------------------------------	--------------------------------

1 (1)	2	15,94	3,84	0,0944
2 (3)	5	37,39	9,46	0,1636
3 (5)	6	50,79	11,36	0,2130
4 (6)	6	44,79	11,02	0,2425
5 (7)	7	56,12	13,12	0,2474
6 (8)	7	55,34	13,20	0,2499

For the first scenario, the generated loading pattern based on loading priorities from the selected bundles is shown in Figure 6.12 and its statistics in Table 6.12.

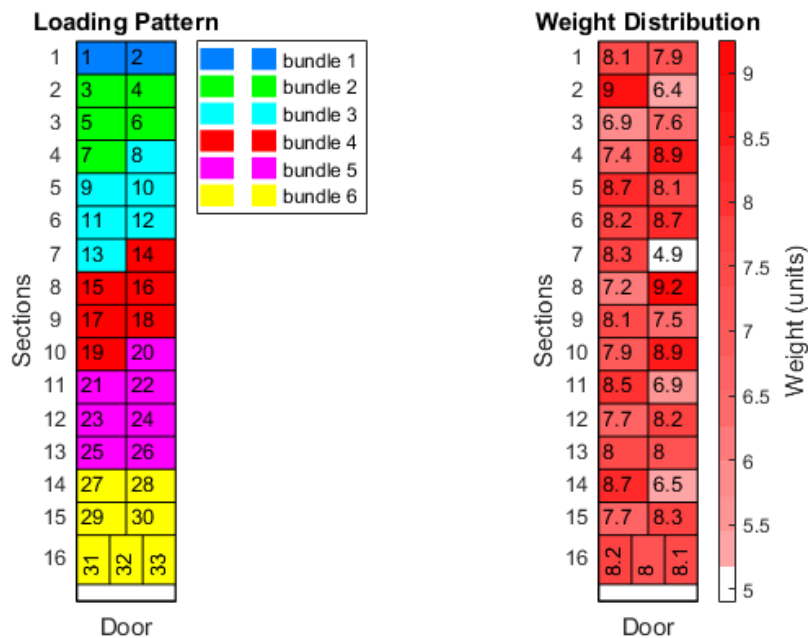


Figure 6.12: Loading pattern and weight distribution

Table 6.12: Loading pattern output statistics

Usable container volume (*) (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)
63.36	62	97.85	260.37

*usable volume that the pallets can occupy within the container’s volume due to the pallet’s dimensions.

The selected bundles are organized from the lowest priority (farthest from the door) to highest priority. This priority is given by the value of the logistic importance of each bundle.

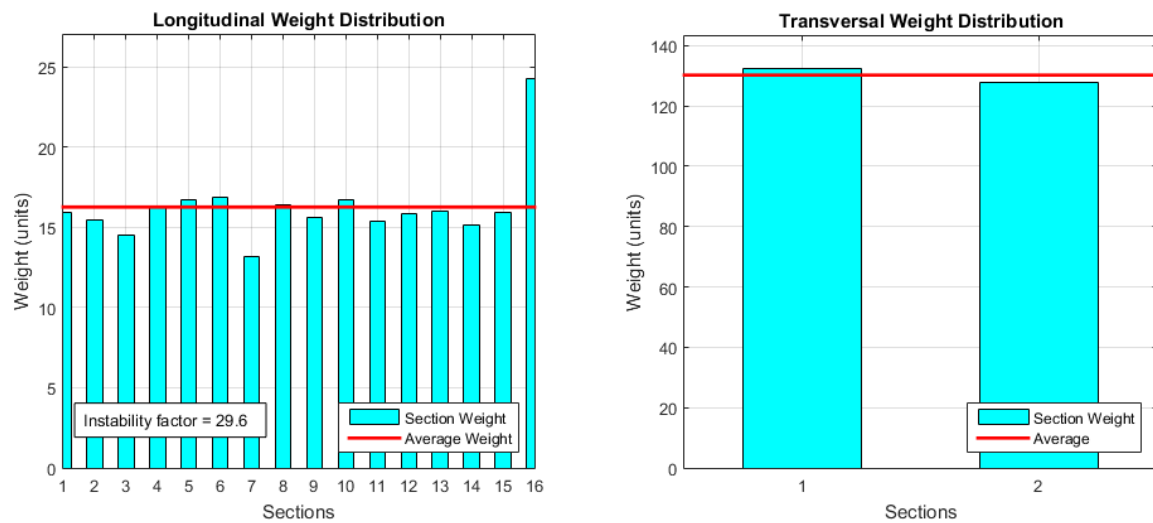


Figure 6.13: Longitudinal and transversal weight distribution (efficient)

This loading pattern guarantees the maximum efficiency according to the pickup/delivery route, sacrificing some of the container stability as seen by the oscillating weights around the average which is translated by value of the *Instability factor*. Nonetheless, the *Instability factor* value is still low which is a result of the distribution being executed according to the chessboard matrix strategy, allowing for a stable transversal distribution as shown by the graphs in Figure 6.13.

As expected from the coefficients used for the logistic importance function, the 0/1 KSP algorithm selected bundles that yielded a high percentage of occupied volume while keeping the weight under the maximum allowed (as seen in Table 6.12).

Using the data structure developed for this model (detailed in 3.1.2.3) and implemented in a linear database, it is possible to produce a 3D representation of the loading pattern (Figure 6.14) in order to give the operator a global perception of the loading pattern.

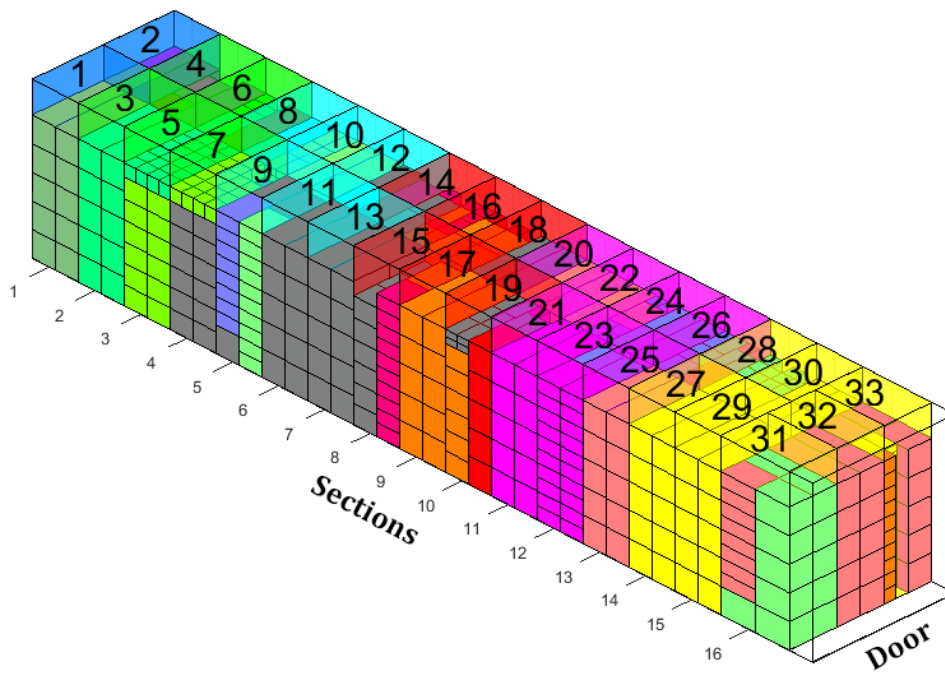


Figure 6.14: 3D loading pattern (efficient)

Additionally, pallets can be accessed and represented individually by using their number as search parameter. For example, Figure 6.15 shows pallet 31 whose packing pattern was retrieved from the database.

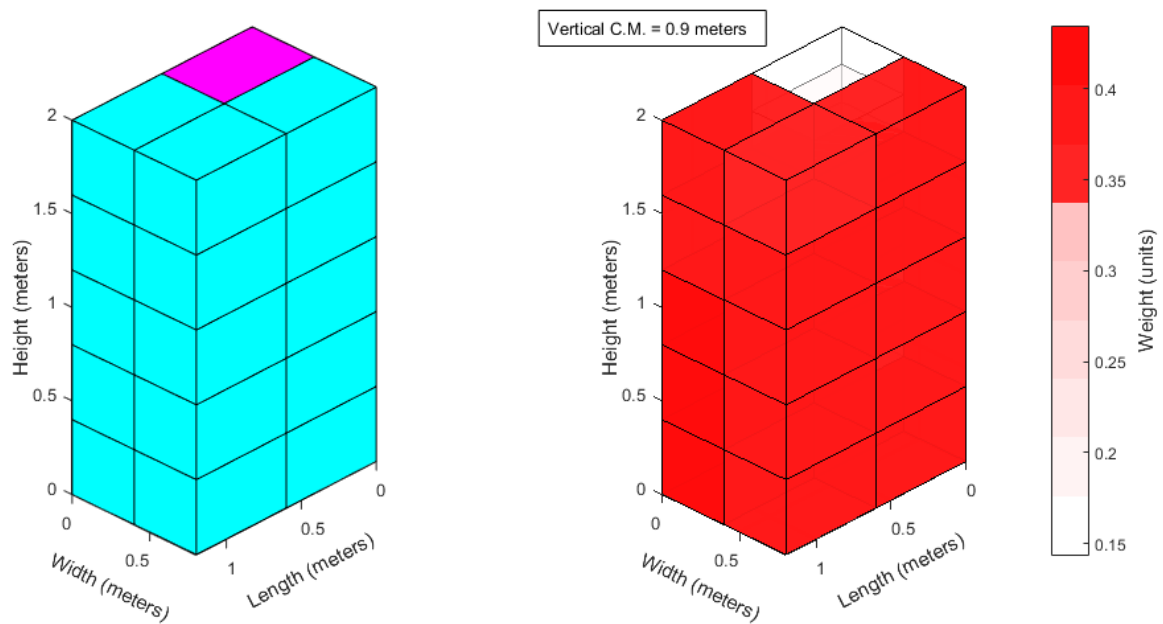


Figure 6.15: Packing pattern (pallet 31)

For the second scenario, the previously generated bundles are broken and pallets are placed individually as shown by Figure 6.16:

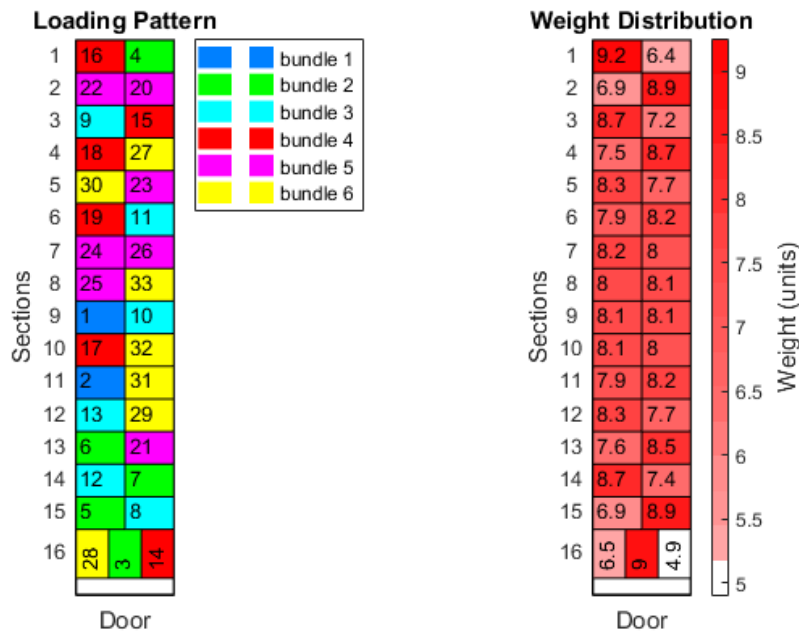


Figure 6.16: Loading pattern with uniform weight distribution

The bundle's pallets were reorganized in order to achieve a uniform weight distribution as best as possible (Figure 6.17).

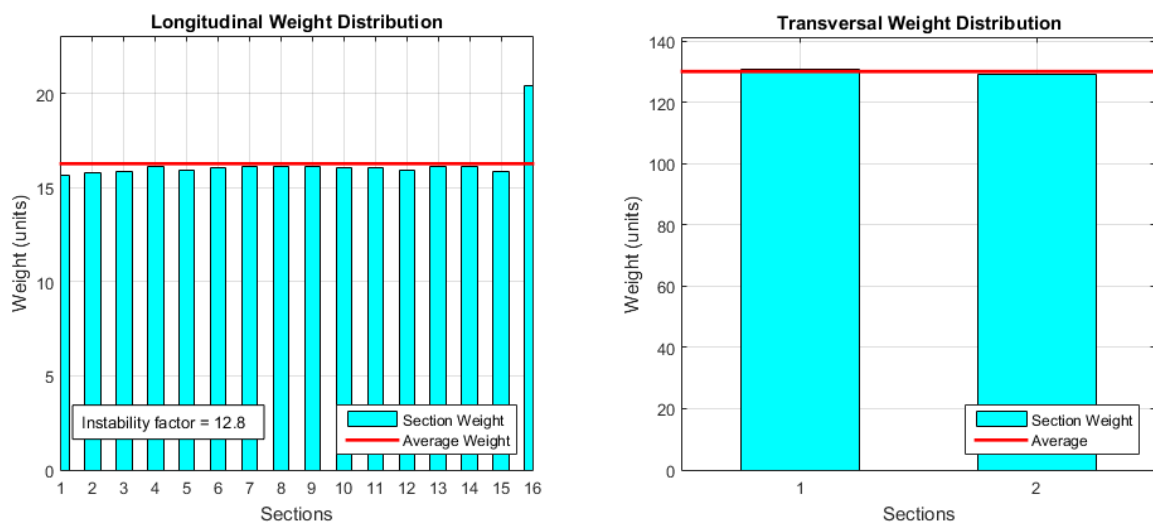


Figure 6.17: Longitudinal and transversal weight distribution (uniform)

This loading pattern guarantees maximum stability (of both dimensions) of the container (due to the chessboard matrix distribution with the similar weights reallocated to the center) as it can be seen by comparing the obtained *Instability factor* value with the previous distribution which is higher than the obtained by this one. However, this is achieved at the expense of the integrity of each bundle.

This scenario can be considered in cases where there is only one bundle that completely fills the container, or when the container stability needs to be maintained during a specific type journey such as long distance transportation (after which the cargo can be unloaded and reorganized back into the more efficient loading pattern).

Once again, using the data structure in the database it is also possible to represent this loading pattern in 3D as shown in Figure 6.18.

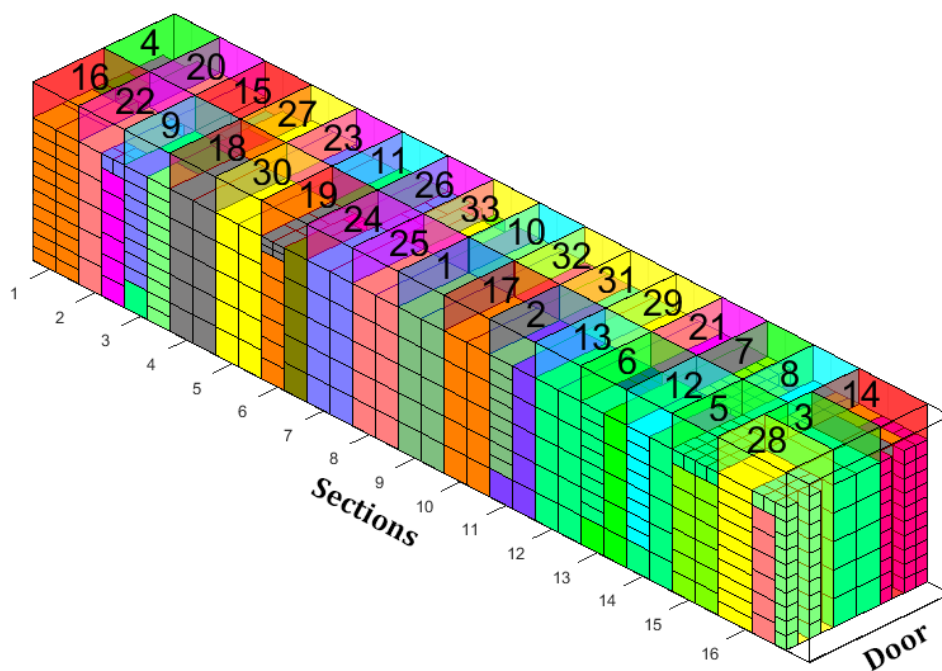


Figure 6.18: 3D loading pattern (uniform)

While these test scenarios considered only one cargo container, the heuristic is designed to operate with multiple trucks, providing loading patterns sequentially while there are trucks and pallets available.

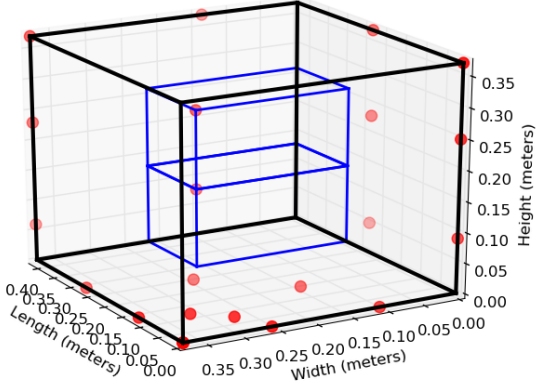
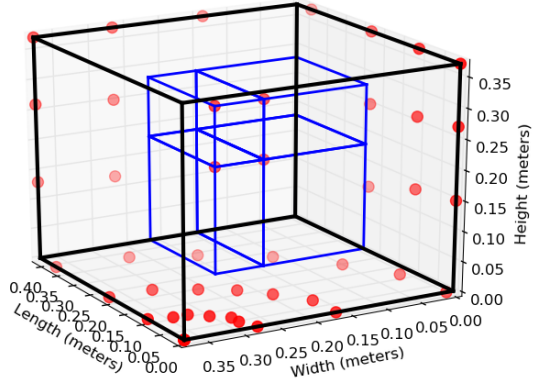
6.2. Sensing Systems

To evaluate the performance of the developed sensing systems, the prototypes were tested in lab conditions where the containers and boxes used were of smaller size compared to the target environment (truck's cargo space with pallets). These measurements were done as a proof of concept.

6.2.1. Laser Rangefinder Prototype

A small container with 0.36 meters of height, 0.37 meters of width and 0.42 meters of length was used as the environment for the tests. The following table shows the obtained results on the measurement of a box with 0.29 meters of height, 0.20 meters of width and 0.14 meters of length, for 2 different sets of generated grid points. The system was located in the top corner facing the reader (top left corner of the *Width x Height* plane) and had a baseline of 0.1 meters.

Table 6.13: LRF prototype results

3D Plot	Grid Dimensions (rows x columns)	Measured Volume (m^3)	Actual Volume (m^3)	Error (%)
	5 x 3 (24 points)	0.004	0.008	50
	6 x 6 (49 points)	0.007	0.008	12.5

As shown in Table 6.13, increasing the number of measurement points increases accuracy by reducing the area of dead zones between points. However, with less points it is still possible to have an idea of the position of the object, which might in some cases be a good tradeoff to have a higher measurement error (i.e. cases where only a rough shape is necessary to be visualized). This error is due to the reconstruction method being partial, i.e., the box volume is being calculated section by section as long as there are enough points in its surface.

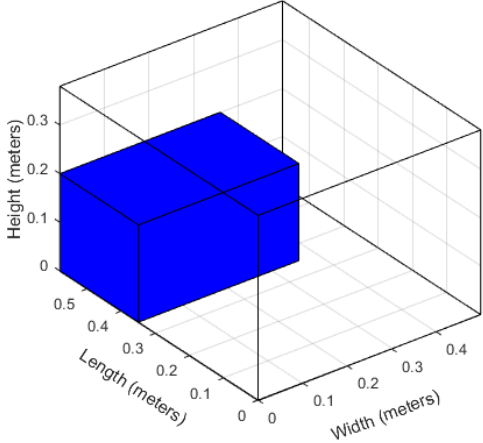
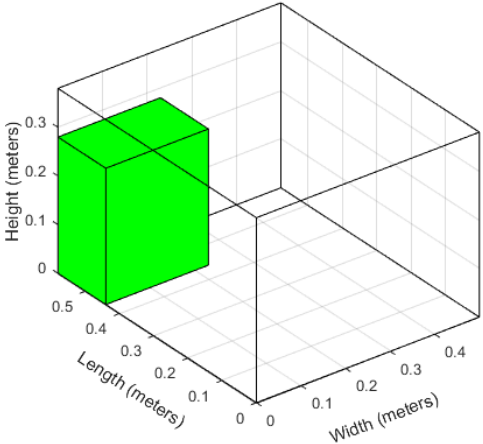
The number of frames taken is equal to the number of points in the grid, which means that a lower error comes at the cost of overall processing time. An improved solution would be to create a grid of point with variable density, where more measurement points would be concentrated on regions where a significant difference of measurement is present, allowing for a more precise volume reconstruction around the edges of the box as well as potentially reducing the number of processed points, since less points would be required as long as no significant variation was detected.

Another important aspect is the stability of the mechanical support that this system relies on (servomechanism) which might pose additional difficulties in moving environments (for example a truck's cargo area). However, this LRF system has the advantage of being easy to implement and is capable of operating in the dark (unlike other systems that require external light sources).

6.2.2. Computer Stereo Vision Prototype

To validate this system, two measurements were taken on a container with the dimensions 0.50 x 0.58 x 0.38 (width x length x height) in meters. The first box had the dimensions of 0.35 x 0.22 x 0.19 and the second box had 0.27 x 0.22 x 0.14, both in meters. The cameras were positioned on the top middle of the *Width x Height* plane and had a baseline of 0.2 meters.

Table 6.14: Stereo vision prototype results

3D Plot	Measured Volume (m^3)	Actual Volume (m^3)	Error (%)
	0.0166	0.0146	13.6
	9.02×10^{-3}	8.91×10^{-3}	12.3

As shown by Table 6.14, both cases have a low error. This is expected from a stereo system since it is capable of recognizing the whole relevant surfaces of the box allowing for a full reconstruction of the volume. This system also requires less input data (frames taken per target) than the previous LRF prototype, which makes it output the results faster.

However, this system still has limitations such as requiring sufficient ambient light (from an external source) to illuminate the scene while being sensitive to excess light which can input noise during the color differentiation of each box (the subtraction of background phase). During measurements, ambient light had to be controlled through experimental adjustment until a sufficient and not intrusive value (i.e. an intensity value that would introduce noise in the measurement) was reached.

An improvement to deal with the ambient light issue is to execute edge detection for the identification of boxes (along with a formal corner detection algorithm) instead of the

color/intensity subtraction method. This is a more complex and computationally intensive approach than the method used, however it would provide a more robust method of box detection.

The system physical stability is not as critical since there are no moving parts and the whole structure is firmly attached to the container. Additionally, stereo rectification can be implemented to fully compensate for alignment errors for more problematic environments.

7. Conclusions and Future Work

In this dissertation, a state of the art review of the available literature in sensing technology for volume measurement and reconstruction, cargo planning and volume assignment approaches as well as market available solutions was conducted, while focusing on their working principles and characteristics.

A heuristic was designed and implemented, producing solutions for two logistic scenarios:

1. Pallet assembling

Packing patterns are produced using a volume placement strategy (walls and towers building) that promotes compactness and vertically stable pallets with a weakly heterogeneous set of boxes.

2. Cargo loading

A cost function was devised using some of the important factors for logistic operators. The calculated output of this function was then used as the selecting criteria for the 0/1 KSP algorithm, allowing for the generation of optimal loading patterns according to the considered constraints.

Both scenarios were integrated using a data structure developed to model this logistic process, allowing for an easy representation in 2D or 3D of the output produced by the heuristic.

Tests conducted on the heuristic revealed its correct functionality according to the assumptions made for the developed model and implemented approaches.

When it comes to sensing technology and volume reconstruction, two prototypes were designed and developed using two different approaches as proof of concept:

1. Laser rangefinder prototype

A robust approach (independent of ambient light since it has its own light source) in stable environments, that scans the indicated volume with a reconstruction error dependent on the number of measurement points defined (each point requiring a frame to be taken and processed).

2. Computer Stereo Vision prototype

An approach less robust (ambient light dependency) but with a high reconstruction precision that is more physically stable and needs less frames to be processed.

Both prototypes obtained satisfactory results on the executed tests, with the most promising approach being the Computer Stereo Vision that with further development can become a robust solution to be tested in real world environments such as a truck's cargo area.

7.1. Future Work

Future improvements on this work would include:

Heuristic

- Implementing a horizontal layer construction along with the current tower building recursive method to better fill the base of the pallet.

This could be implemented by calculating which boxes can better fill the container's floor and position as many full layers as the number of boxes allow. The result could then be viewed as a small container (since the bottom is filled with layers of boxes) were the tower building strategy could be applied to place the remaining boxes.

- Implement horizontal weight distribution to improve on the overall pallet stability.

By having a horizontal layer placement strategy, the selection of boxes to be placed could be done so that it alternates heavy and light weights, creating a chessboard matrix of weights.

- Further develop the placement heuristic by improving the selection criteria of placed boxes and overhaul results with backtracking search and repositioning.

An immediate improvement would be to pre-calculate which box types would better occupy the pallet's volume. Using this calculation and the number of boxes of each type, the heuristic could prioritize the selection of said boxes, in order to potentially improve the volume usage.

Sensing technology

- Devising and implementing a variable density grid of measurement points for the LRF prototype, in order to reduce the number of frames taken while maintaining or improving the reconstruction precision (for academic purposes or for implementation in stable environments).

This could be achieved by taking an initial snapshot of the volume being measured and perform edge detection, then the grid could be set up so that the system would take more measurements along the computed edge's positions and less in other places.

- Implementing an edge detection method for boxes recognition along with a corner detection algorithm, in order to improve the robustness of the SV prototype against the ambient light.

Possibly with Canny edge detection and Harris corner detector algorithm fine-tuned or even adapted for the target environments.

- Implement stereo rectification for the SV system.

Either applying available algorithms or delve into further epipolar geometry to implement a more specialized solution.

References

- [1] M. Christopher, *Logistics & Supply Chain Management*, Dorchester: FT Press, 2011.
- [2] C. Kille, M. Schwemmer and C. Reichenauer, *TOP 100 in European Transport and Logistics Services*, Fraunhofer SCS, 2015.
- [3] A. Bortfeldt and G. Wäscher, "Constraints in container loading – A state-of-the-art review," *European Journal of Operational Research*, no. 229, pp. 1-20, 2013.
- [4] G. Sansoni, M. Trebeschi and F. Docchio, "State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation," *Sensors*, vol. 9, no. 1, pp. 568-601, 2009.
- [5] M.-C. Amann, T. Bosch, M. Lescure, R. Myllyla and M. Rioux, "Laser ranging: a critical review of usual techniques for distance measurement," *Society of Photo-Optical Instrumentation Engineers*, vol. I, no. 40, pp. 10-19, 2001.
- [6] "Distance laser sensor model MRL3," Metrology Resource Co., 2010. [Online]. Available: <http://www.metrologyresource.com/laser-sensor-MRL3.php>. [Accessed 5 September 2016].
- [7] T. Danko, "Webcam Based DIY Laser Rangefinder," 27 August 2009. [Online]. Available: https://sites.google.com/site/todddanko/home/webcam_laser_ranger. [Accessed 29 April 2016].
- [8] J. Mrovlje and D. Vrančić, "Distance measuring based on stereoscopic pictures," in *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, Izola, 2008.
- [9] M. A. Mahammed, A. I. Melhum and F. A. Kochery, "Object Distance Measurement by Stereo VISION," *International Journal of Science and Applied Information Technology (IJSAIT)*, vol. 2, no. Special Issue of ICCTE 2013, pp. 05-08, 2013.
- [10] T. MASUDA, E. ISHIYAMA and H. TAMAYAMA, "Development of "3D Measurement System" Using Images Taken with a "FinePix REAL 3D W3" 3D Digital Camera," *FujiFilm Res & Dev*, no. 57, pp. 38-41, March 2012.

- [11] M. Miura, S. Sakai, J. Ishii, K. Ito and T. Aoki, "An Easy-to-Use and Accurate 3D Shape Measurement System Using Two Snapshots," in *International Workshop on Advanced Image Technology*, Nagoya, 2013.
- [12] T. Jia, Z. Zhou and H. Gao, "Depth Measurement Based on Infrared Coded Structured Light," *Journal of Sensors*, vol. 2014, 2014.
- [13] F. Alhwarin, A. Ferrein and I. Scholl, "IR Stereo Kinect: Improving Depth Images by Combining Structured Light with IR Stereo," in *PRICAI 2014: Trends in Artificial Intelligence*, Springer International Publishing, 2014, pp. 409-421.
- [14] M. Hansard, S. Lee, O. Choi and R. Horaud, *Time-of-Flight Cameras: Principles, Methods and Applications*, Springer London, 2013.
- [15] L. Li, "Time-of-Flight Camera – An Introduction," Texas Instruments, 2014.
- [16] X. Zhang, J. Morris and R. Klette, "Volume Measurement Using a Laser Scanner," CITR, The University of Auckland, New Zealand, Auckland, 2005.
- [17] Z. Xiang and W. Zhou, "3D Map Building for Mobile Robots Using a 3D Laser Range Finder," in *Intelligent Computing in Signal Processing and Pattern Recognition*, vol. 345, Springer Berlin Heidelberg, 2006, pp. 785-790.
- [18] W. Morris, I. Dryanovski and J. Xiao, "3D indoor mapping for micro-UAVs using hybrid range finders and multi-volume occupancy grids," in *RSS 2010 workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Zaragoza, 2010.
- [19] D. A. Matos, *Deteção do Espaço Navegável para o ATLASCAR usando informação 3D*, Aveiro: Universidade de Aveiro, 2013.
- [20] P. Dias, M. Matos and V. Santos, "A Fast Low-Cost 3D Scanner for Navigation and Modelling," in *Encontro Científico do Festival Nacional de Robótica 2004*, 2004.
- [21] P. Dias, M. Matos and V. Santos, "3D Reconstruction of Real World Scenes Using a Low-Cost 3D Range Scanner," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 7, pp. 486-497, 2006.
- [22] Z.-J. Wang, K. W. L. Dr. and J. K. Levy, "A heuristic for the container loading problem: A tertiary-tree-based dynamic space decomposition approach," *European Journal of Operational Research*, vol. 191, no. 1, pp. 84-97, 2008.

-
- [23] H. Gehring and A. Bortfeldt, "A genetic algorithm for solving the container loading problem," *International Transaction in Operational Research*, vol. 4, no. 5/6, pp. 401-418, 1997.
- [24] A. Moura and J. F. Oliveira, "A GRASP Approach to the Container-Loading Problem," *IEEE Intelligent Systems*, vol. 20, no. 4, pp. 50-57, 2005.
- [25] F. Parreño, R. Alvarez-Valdes, J. Oliveira and J. Tamarit, "A Maximal-Space Algorithm for the Container Loading Problem," *Inform Journal on Computing*, vol. 20, no. 3, pp. 412-422, 2008.
- [26] L. J. P. d. Araújo and P. R. Pinheiro, "A Hybrid Methodology Approach for Container Loading Problem Using Genetic Algorithm to Maximize the Weight Distribution of Cargo," in *Real-World Applications of Genetic Algorithms*, Rijeka, InTech, 2012, pp. 183-198.
- [27] Velodyne LiDAR, "Velodyne HDL-64E LiDAR," [Online]. Available: http://velodynelidar.com/docs/datasheet/63-9194_Rev-D_HDL-64E_Data%20Sheet_Web.pdf. [Accessed 16 September 2016].
- [28] T. Deyle, "Velodyne HDL-64E Laser Rangefinder (LIDAR) Pseudo-Disassembled," Hizook, 4 January 2009. [Online]. Available: <http://www.hizook.com/blog/2009/01/04/velodyne-hdl-64e-laser-rangefinder-lidar-pseudo-disassembled>. [Accessed 16 September 2016].
- [29] SICK AG Waldkirch, "LMS200/211/221/291 Laser Measurement Systems," 2006. [Online]. Available: <http://sicktoolbox.sourceforge.net/docs/sick-lms-technical-description.pdf>. [Accessed 16 September 2016].
- [30] T. Deyle, "SICK Laser Rangefinder (LIDAR) Disassembled," Hizook, 15 December 2008. [Online]. Available: <http://www.hizook.com/blog/2008/12/15/sick-laser-rangefinder-lidar-disassembled>. [Accessed 16 September 2016].
- [31] J. Schöning and G. Heidemann, "Taxonomy of 3D Sensors - A Survey of State-of-the-Art Consumer 3D-Reconstruction Sensors and their Field of Applications," in *International Conference on Computer Vision Theory and Applications*, 2016.
- [32] Microsoft, "Kinect for Windows Sensor Components and Specifications," [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131033.aspx?f=255&MSPPError=-2147217396>. [Accessed 16 September 2016].
-

- [33] Microsoft, "Kinect Sensor for Xbox One," [Online]. Available: https://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-Sensor-for-Xbox-One/productID.2267482500. [Accessed 16 September 2016].
- [34] Microsoft, "Kinect for Windows v2," 2015. [Online]. Available: <http://blogs.microsoft.co.il/msdn/2015/01/07/kinect-for-windows-v2/>. [Accessed 16 September 2016].
- [35] Point Grey, "Stereo Vision Introduction and Applications," 3 November 2015. [Online]. Available: <http://www.ptgrey.com/support/downloads/10353?>. [Accessed 16 September 2016].
- [36] Point Grey, "Bumblebee 2 Stereo Vision System Price/Product List," 18 August 2006. [Online]. Available: <http://yuriythebest.g0dsoft.com/Bumblebee2%20Price%20List%2008-18-06.pdf>. [Accessed 16 September 2016].
- [37] J. L. Blanco, "Supported hardware and sensors," MRTP, 8 October 2013. [Online]. Available: http://www.mrpt.org/tutorials/supported_hardware_and_sensors/#42_BumblebeeBumblebee2_Stereo_cameras. [Accessed 16 September 2016].
- [38] Stereolabs, "ZED - Depth Sensing and Tracking," Stereolabs, [Online]. Available: <https://www.stereolabs.com/zed/specs/>. [Accessed 16 September 2016].
- [39] ReconstructMe, "Intel RealSense R200 Review," [Online]. Available: http://reconstructme.net/qa_faqs/intel-realsense-r200-review/. [Accessed 14 October 2016].
- [40] Intel, "Intel® RealSense™ Data Ranges," 1 March 2016. [Online]. Available: <https://software.intel.com/en-us/articles/intel-realsense-data-ranges>. [Accessed 14 October 2016].
- [41] Intel, "Intel® RealSense™ Developer Kit (R200)," [Online]. Available: <http://click.intel.com/intel-realsense-developer-kit-r200.html>. [Accessed 14 October 2016].
- [42] S. Sarwar, "Software Architecture - To analyze cargo loading optimization algorithm," 24 March 2010. [Online]. Available: <https://softarchitect.wordpress.com/2010/03/24/software-architecture-to-analyze-cargo-loading-optimization-algorithm/>. [Accessed 17 September 2016].

-
- [43] 3DBinPacking, "Single Container Packing," 3DBinPacking, [Online]. Available: <https://3dbinpacking.com/products/singleBinPacking>. [Accessed 17 September 2016].
- [44] Softruck, "Truck, Container and Pallet Loading Software," Softruck, 2016. [Online]. Available: <http://www.softtruck.com/index.htm>. [Accessed 17 September 2016].
- [45] Dreamsofts Optimization, "Cargo Optimizer Enterprise," Dreamsofts Optimization, 15 June 2016. [Online]. Available: http://www.cargooptimizer.com/Cargo_Optimizer_Enterprise_EN.html. [Accessed 17 September 2016].
- [46] Logen Solutions, "Truck, Container and Pallet Loading Software," Logen Solutions, 2013. [Online]. Available: http://www.logensolutions.com/VMS/CubeMaster/Cargo_Load_Plan_Optimization_Software_Overview.html. [Accessed 17 September 2016].
- [47] Packer 3d, "Packer3d," 2015. [Online]. Available: <http://www.packer3d.com/>. [Accessed 17 September 2016].
- [48] E. W. Weisstein, "Cuboid," MathWorld -- A Wolfram Web Resource, [Online]. Available: <http://mathworld.wolfram.com/Cuboid.html>. [Accessed 4 March 2016].
- [49] "Types of trucks," Albacor Shipping Inc., [Online]. Available: <http://albacorshipping.ru/en/useful-info/types-of-railway-containers/>. [Accessed 24 May 2016].
- [50] "Wikipedia: Pallet," Wikimedia Foundation, Inc., [Online]. Available: https://en.wikipedia.org/wiki/Pallet#European_pallets. [Accessed 24 May 2016].
- [51] "Wikipedia: Knapsack problem," Wikipedia Foundation, Inc., [Online]. Available: https://en.wikipedia.org/wiki/Knapsack_problem. [Accessed 20 March 2016].
- [52] M. Hristakeva and D. Shrestha, *Different Approaches to Solve the 0/1 Knapsack Problem*.
- [53] T. Danko, "Webcam Based DIY Laser Rangefinder," 27 August 2009. [Online]. Available: https://sites.google.com/site/todddanko/home/webcam_laser_ranger. [Accessed 29 April 2016].
- [54] P. Fonseca, *Sistemas de Instrumentação Electrónica*, Universidade de Aveiro, 2011/12.

Annexes

A. Triangulation Error Graphs

This annex presents complimentary graphs and discussion on the measurement errors tolerated by the sensing systems. All graphs will only show error values below 10% of the considered distance of measurement.

A.i. Laser rangefinder

In 3.2.2, discussion is held on how increasing the baseline reduces the measurement errors. A series of graphs for different baselines and distances is shown to support this claim.

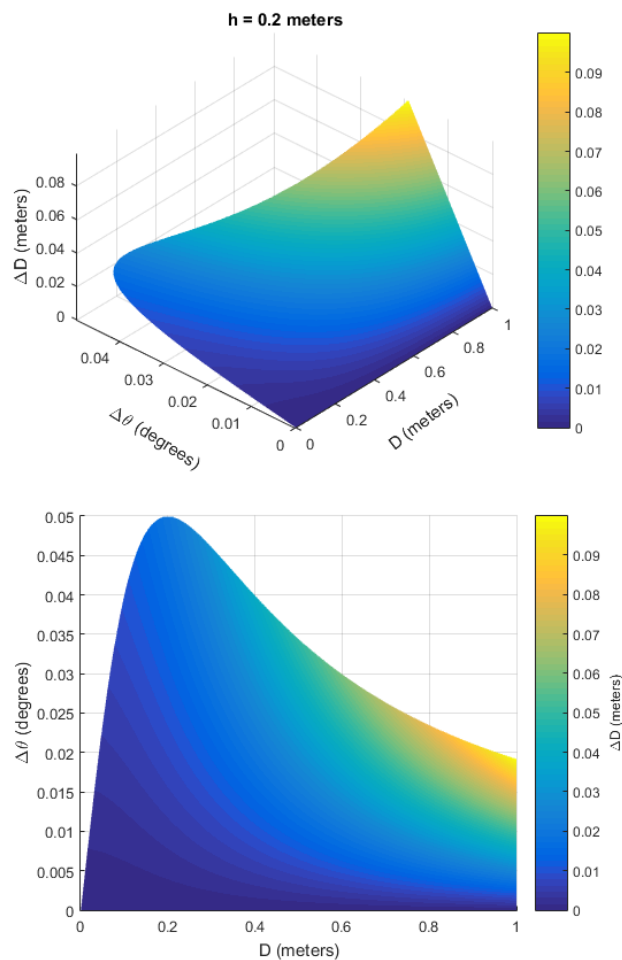


Figure 0.1: LFR error variation - 1

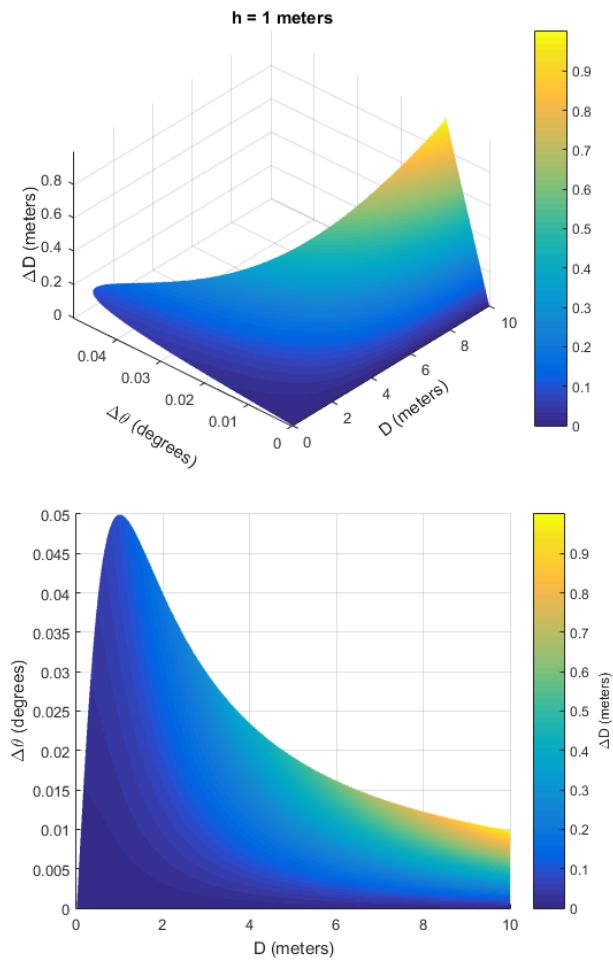


Figure 0.2: LFR error variation - 2

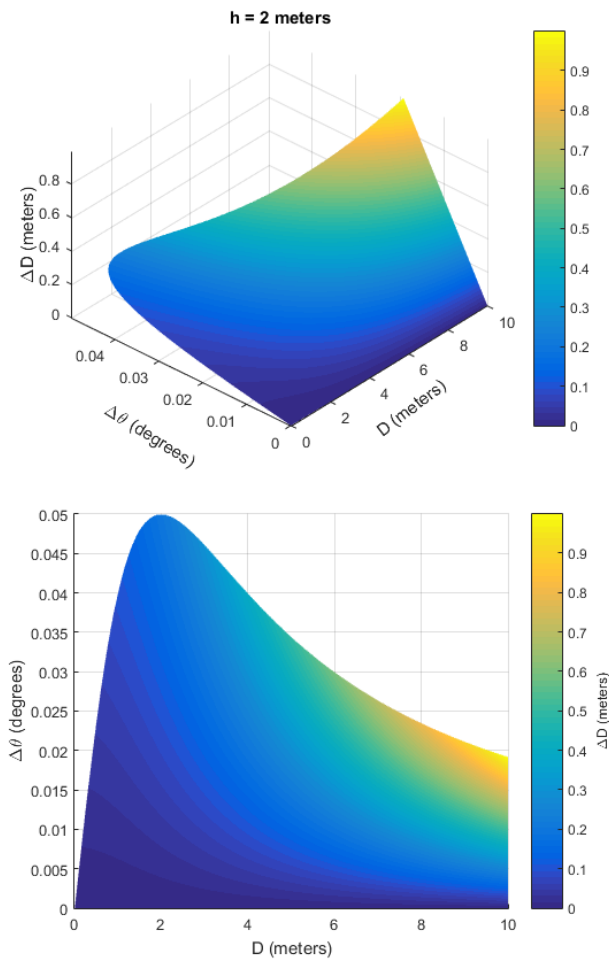


Figure 0.3: LFR error variation - 3

Comparing Figure 3.18 (page 45) with Figure 0.1, the increase of 10cm in the baseline resulted in a wider range of tolerated error variation. The same conclusion can be taken from wider baselines as shown in Figure 0.2 and Figure 0.3.

All the previous figures consider a static baseline, i.e., no error is introduced by variations of Δh . However, as seen by Figure 3.19 (page 46), a small error in the baseline can impact the measurement reliability of the system. Figure 0.4 shows this impact at a greater distance.

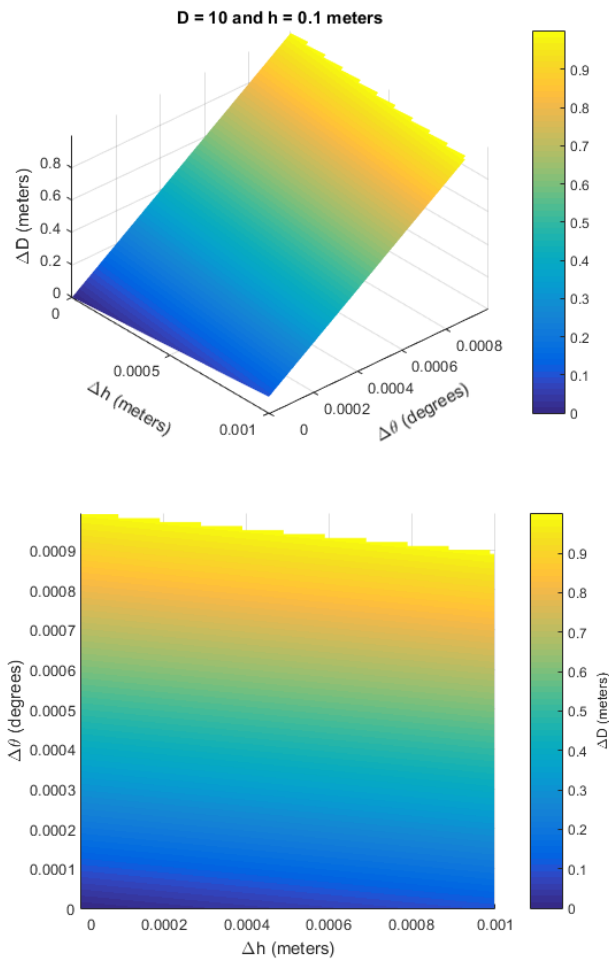


Figure 0.4: LFR error variation - 4

As it can be seen, at greater distances the impact is even more noticeable as the error variation permitted is even smaller, with $\Delta\theta$ taking values only achievable by having a high resolution camera.

A.ii. Stereo Vision

In 3.2.3, like the previous system, graphs where shown with the evolution of the measurement error. The following graphs are complimentary to the discussion held in the section.

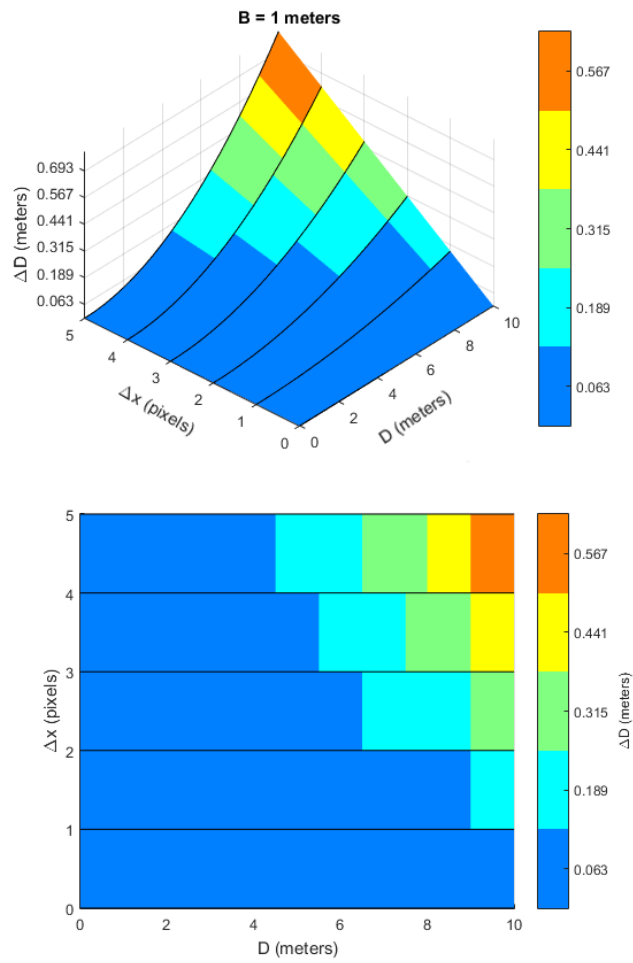


Figure 0.5: SV error variation - 1

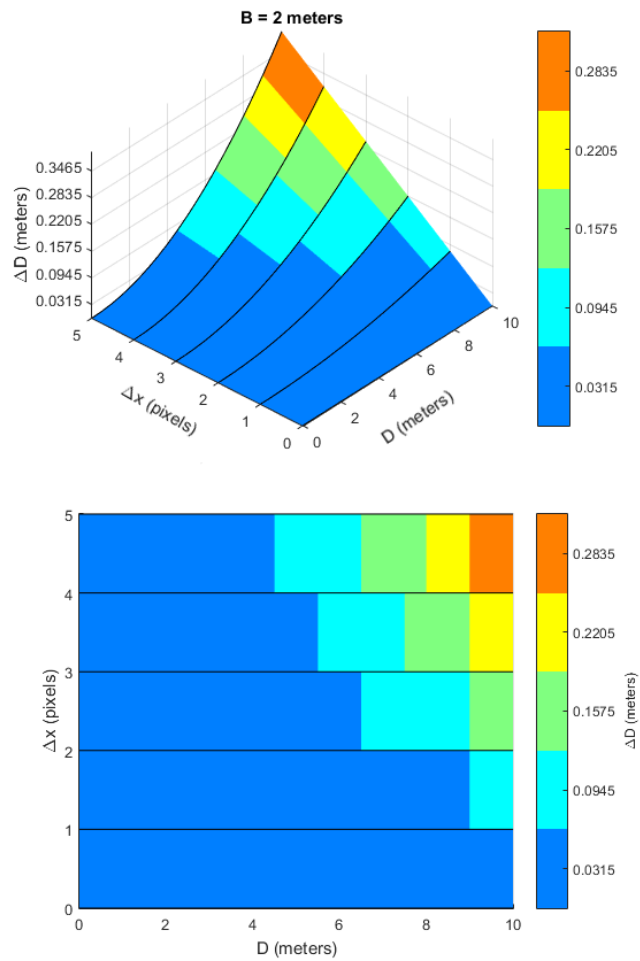


Figure 0.6: SV error variation - 2

When comparing Figure 0.5 and Figure 0.6, it can be seen that a bigger baseline reduces the measurement error for greater distances.

Both graphs consider no error introduced by baseline variations ΔB , however as previously seen in Figure 3.23 (page 50) this system is better prepared for this sort of error especially with bigger baselines. The next set of graphs further illustrate this claim.

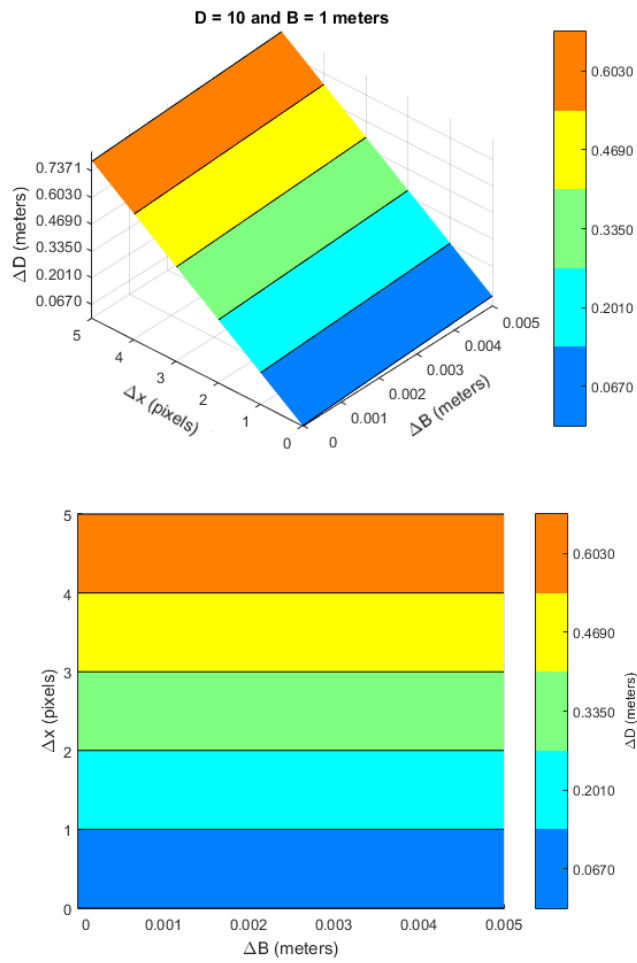


Figure 0.7: SV error variation - 3

As it can be seen by Figure 0.7, a bigger baseline increases even further the tolerance margin for eventual errors.

B. 0/1 Knapsack Problem Algorithm

A dynamic programming approach of the 0/1 KSP can be implemented as shown by the following pseudo code (adapted from [51]):

```
// Input:
// weights (array w)
// values (array v)
// knapsack capacity (W)

// define/calculate the number of elements (n)

for j from 0 to W do:
    m[0, j] := 0

for i from 1 to n do:
    for j from 0 to W do:
        if w[i-1] > j then:
            m[i, j] := m[i-1, j]
        else:
            m[i, j] := max(m[i-1, j], m[i-1, j-w(i-1)] + v[i-1])
return m[n,W] //best
```

To find the elements which the sum yields the maximum value of $m[n, W]$ (as defined in 3.1.6) the process is backtracked:

```
// Input:
// weights (array w)
// calculated solution table (m[i,w])
// maximum value found (best)
// number of elements (n)
// knapsack capacity (W)

while best > 0 do:
    while m[n,W] == best do:
        n = n - 1
    end
    W = W - w(n)
    n = n - 1
    best = m[n,W]
    selected_items(n+1) = 1 // the index of the item is put to 1
end
return selected_items
```

B.i. Application Example

A knapsack with a total weight capacity of $W = 7$ has to be filled with a combination of items with the best sum of value possible. The items available are the following:

Table B.1: Input values

item (n)	weight (w)	value (v)
1	1	1
2	3	4
3	4	5
4	5	7

The search procedure begins with a construction of a table for the values of n and the range values of w . The first column is filled with zeros. The table is then filled from left to right, top to bottom according to the conditions shown in 3.1.6.

Table B.2: 0/1 KSP results table

			weight range							
n	v	w	0	1	2	3	4	5	6	7
1	1	1	0	1	1	1	1	1	1	1
2	4	3	0	1	1	4*	5	5	5	5
3	5	4	0	1	1	4	5	6	6	9*
4	7	5	0	1	1	4	5	7	8	9

The best sum of value reached is 9. To find which items are selected, the table is backtracked by finding which items contributed for the best value combination. This backtracking is done from bottom up, right to left. In this example the backtracked values are marked by ‘*’ and correspond to the items 2 and 3.

C. Generated Data for the Heuristic and Examples

C.i. Pallet Assembling

In 6.1.2, a scenario using 3-box types was presented as the standard application of the heuristic. The following tables show the randomly generated boxes (as selected by the 0/1 KSP) and the boxes left out after the placement.

Table C.3: Generated input data

box type	weight (weight units)
1	0,475
1	0,484
1	0,417
1	0,49
1	0,411
1	0,475
1	0,473
1	0,472
1	0,413
2	0,345
2	0,351
2	0,353
2	0,386
2	0,368
2	0,381
2	0,353
2	0,396
2	0,307
2	0,354
2	0,328
2	0,348

2	0,368
2	0,321
2	0,361
3	0,009
3	0,01
3	0,009
3	0,009
3	0,01
3	0,009
3	0,009
3	0,009
3	0,009
3	0,009
3	0,009
3	0,01

Table C.4: Boxes left out of the placement

box type	weight (weight units)
1	0,417
1	0,411
1	0,472
1	0,413
2	0,307
2	0,328
2	0,321

The following table shows the box placement list generated by the heuristic, for the example at hand, as stored in the database.

Table C.5: Placement list generated by the heuristic

x (meters)	y (meters)	z (meters)	box type	weight (weight units)	azimuthal angle (binary value)
0	0	0	2	0,396	0
0	0	0,3	2	0,386	0
0	0	0,6	2	0,381	0
0	0	0,9	3	0,01	0
0	0,2	0,9	3	0,01	0
0,4	0	0	2	0,368	0
0,4	0	0,3	2	0,361	0
0,4	0	0,6	2	0,354	0
0,4	0	0,9	3	0,01	0
0,4	0,2	0,9	3	0,009	0
0,8	0	0	2	0,368	0
0,8	0	0,3	2	0,353	0
0,8	0	0,6	2	0,353	0
0,8	0	0,9	3	0,009	0
0,8	0,2	0,9	3	0,009	0
0	0,4	0	1	0,49	0
0	0,4	0,2	1	0,484	0
0	0,4	0,4	1	0,475	0
0	0,4	0,6	1	0,475	0
0	0,4	0,8	1	0,473	0
0,5	0,4	0	2	0,351	0
0,5	0,4	0,3	2	0,348	0
0,5	0,4	0,6	2	0,345	0
0,5	0,4	0,9	3	0,009	0
0,5	0,6	0,9	3	0,009	0
0,9	0,4	0	3	0,009	1
0,9	0,4	0,1	3	0,009	1
0,9	0,4	0,2	3	0,009	1

C.i.1. Other examples

Using the same 3-box type set as the standard case in 6.1.2, some extra examples of outputs are shown along with their statistics, to further illustrate the heuristic procedure.

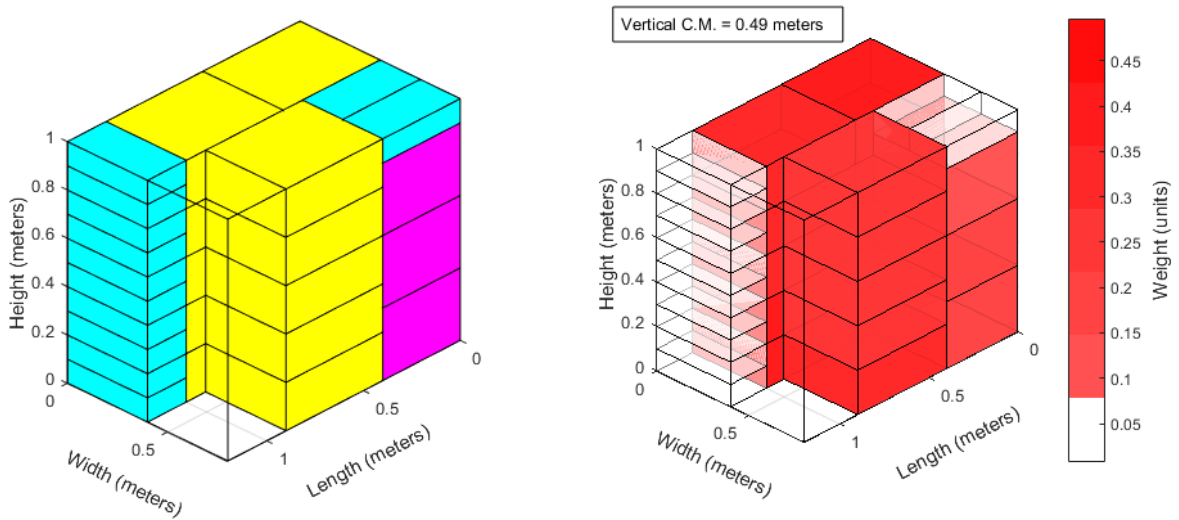


Figure 0.8: Example - 1

Table C.6: Statistics - 1

Container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)	Number of boxes used	Number of boxes left out	Box usage (%)
0.96	0.84	87.5	7.98	30	4	88.24

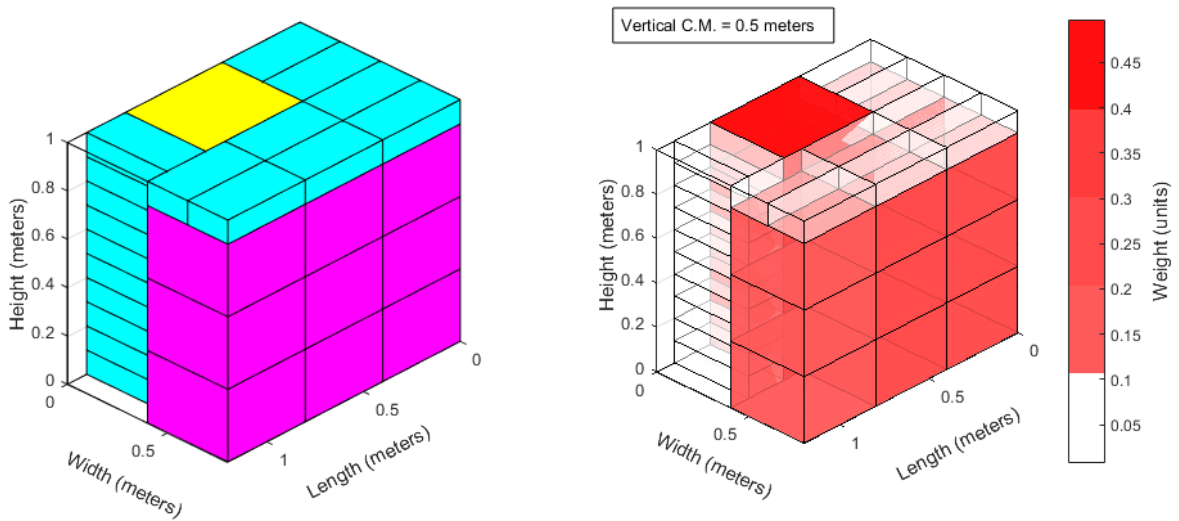


Figure 0.9: Example - 2

Table C.7: Statistics - 2

Container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)	Number of boxes used	Number of boxes left out	Box usage (%)
0.96	0.92	95.93	7.06	35	38	47.95

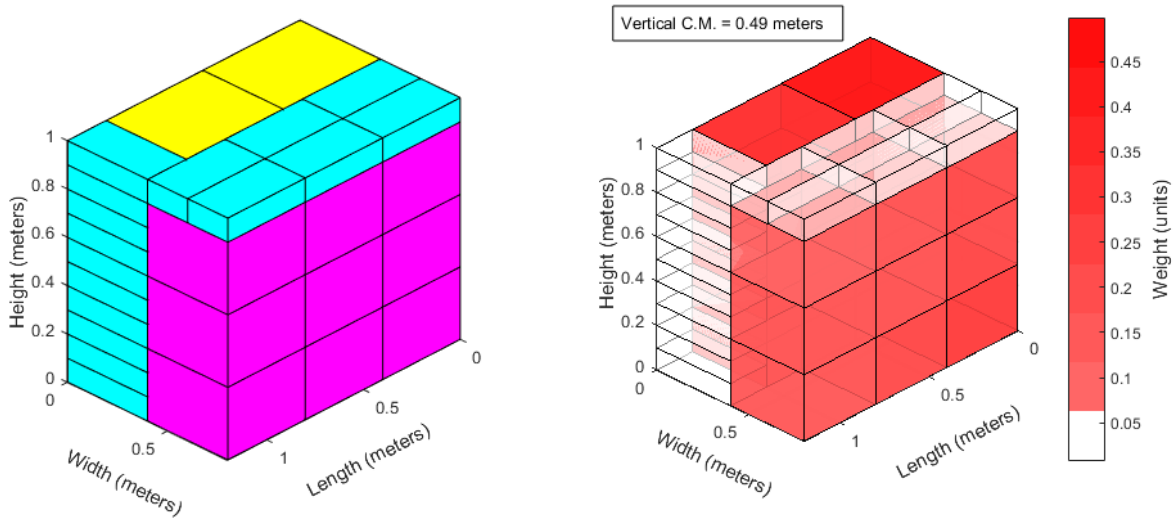


Figure 0.10: Example - 3

Table C.8: Statistics - 3

Container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)	Number of boxes used	Number of boxes left out	Box usage (%)
0.96	0.96	100	7.872	35	0	100

C.ii. Cargo Loading

In 6.1.3, an example of cargo loading (along with pallet assembling) was presented. The following tables the placement list of the pallets in both presented weigh distribution scenarios.

Table C.9: Bundle layout placement (loading priorities)

x (meters)	y (meters)	bundle number	azimuthal angle (binary value)	pallet weight (weight units)
0	0	1	0	8,06
1,2	0	1	0	7,88
0	0,8	2	0	9,03
1,2	0,8	2	0	6,43

Object Grouping in Limited Spaces

0	1,6	2	0	6,93
1,2	1,6	2	0	7,56
0	2,4	2	0	7,44
1,2	2,4	3	0	8,91
0	3,2	3	0	8,67
1,2	3,2	3	0	8,07
0	4	3	0	8,21
1,2	4	3	0	8,66
0	4,8	3	0	8,27
1,2	4,8	4	0	4,91
0	5,6	4	0	7,19
1,2	5,6	4	0	9,24
0	6,4	4	0	8,13
1,2	6,4	4	0	7,47
0	7,2	4	0	7,85
1,2	7,2	5	0	8,91
0	8	5	0	8,54
1,2	8	5	0	6,85
0	8,8	5	0	7,66
1,2	8,8	5	0	8,17
0	9,6	5	0	8,04
1,2	9,6	5	0	7,95
0	10,4	6	0	8,65
1,2	10,4	6	0	6,48
0	11,2	6	0	7,67
1,2	11,2	6	0	8,28
0	12	6	1	8,20
0,8	12	6	1	7,95
1,6	12	6	1	8,11

Table C.10: Bundle layout placement (uniform weight distribution)

x (meters)	y (meters)	bundle number	azimuthal angle (binary value)	pallet weight (weight units)
0	0	4	0	9,24
1,2	0	2	0	6,43
0	0,8	5	0	6,85
1,2	0,8	5	0	8,91
0	1,6	3	0	8,67
1,2	1,6	4	0	7,19
0	2,4	4	0	7,47
1,2	2,4	6	0	8,65
0	3,2	6	0	8,28
1,2	3,2	5	0	7,66
0	4	4	0	7,85
1,2	4	3	0	8,21
0	4,8	5	0	8,17
1,2	4,8	5	0	7,95
0	5,6	5	0	8,04
1,2	5,6	6	0	8,11
0	6,4	1	0	8,06
1,2	6,4	3	0	8,07
0	7,2	4	0	8,13
1,2	7,2	6	0	7,95
0	8	1	0	7,88
1,2	8	6	0	8,20
0	8,8	3	0	8,27
1,2	8,8	6	0	7,67
0	9,6	2	0	7,56
1,2	9,6	5	0	8,54
0	10,4	3	0	8,66
1,2	10,4	2	0	7,44
0	11,2	2	0	6,93

Object Grouping in Limited Spaces

1,2	11,2	3	0	8,91
0	12	6	1	6,48
0,8	12	2	1	9,03
1,6	12	4	1	4,91

D. Practical Case Study

A logistic operator has a warehouse in Aveiro where cargo freights (sets of boxes) are received from different clients to palletize and send to their destination. On a given day, the operator has an empty truck that goes from Aveiro to another warehouse in Lisbon. At this time, the operator has boxes from different clients that have to be transported to Lisbon, however, the total cargo exceeds the capacity of the truck. The operator is then confronted with the decision of selecting which cargo of which clients should be prioritized for transportation in this truck so that it maximizes the logistic importance function, λ_r .

To perform this task, the operator employs the heuristic developed in this document to assemble the pallets for the clients and select which ones to load into the truck.

The operator knows that all their clients use specific box formats which are represented by Figure 6.11 (page 86) with their dimensions in Table 6.9 (page 86).

The amount of boxes per client is then showed in Table D.11.

Table D.11: Boxes per client

client	number of boxes
1	45
2	74
3	175
4	179
5	218
6	184
7	253
8	270
9	256
10	305

To palletize these boxes, the operator uses pallets of 1.2 x 0.8 x 2.0 meters (length x width x height), that then are loaded into the truck whose cargo space has 13.6 x 2.45 x 2.5 meters. The truck has a capacity of 33 pallets and supports up to 320 weight units, and is typically loaded using the layout (α) shown in Figure 3.14 (page 40).

Palletization is done according to the heuristic for pallet assembling (described in 5.2.1). Figure 0.11 shows a pallet assembled for client 8 (packing pattern) as an example of the output it provides.

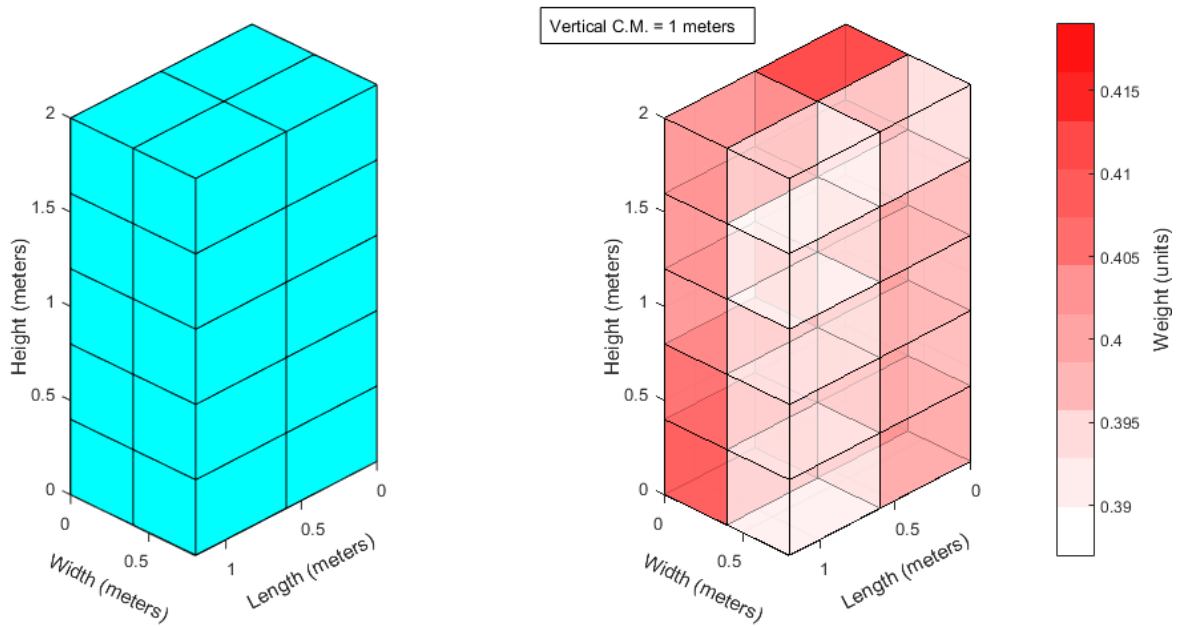


Figure 0.11: Assembled pallet from client 8

The heuristic produces compact and vertically stable packing patterns for all pallets that the operator can easily use as a guiding tool to do the physical assembly.

After all pallets are assembled, they are grouped into bundles according to the client they belong to. These bundles are shown in Table D.12.

Table D.12: Bundles

client	pallets	weight (weight units)	volume (m^3)	logistic importance
1	2	16,36	3,76	0,0741
2	2	15,48	3,55	0,1235
3	3	25,73	5,28	0,0843
4	4	31,85	7,52	0,1921
5	5	38,09	9,20	0,1703
6	6	51,40	11,28	0,1819
7	7	58,48	12,75	0,1784
8	8	64,69	14,85	0,2650
9	9	73,45	17,12	0,2382
10	10	82,06	18,74	0,2512

The logistic importance value (described in 3.1.3) is calculated using the same coefficients as the case presented in 6.1.3.

Having this information, the heuristic selects which bundles should be loaded that maximize the value of logistic importance (i.e., higher profit). Table D.13 shows the selected bundles.

Table D.13: Selected bundles (clients)

client	bundle	pallets	weight (weight units)	volume (m^3)	logistic importance
1	1	2	16,36	3,76	0,0741
3	2	3	25,73	5,28	0,0843
2	3	2	15,48	3,55	0,1235
5	4	5	38,09	9,20	0,1703
4	5	4	31,85	7,52	0,1921
9	6	9	73,45	17,12	0,2382
8	7	8	64,69	14,85	0,2650

The heuristic proceeds to produce a loading pattern for these selected bundles. Figure 0.12 shows how pallets should be loaded into the truck, and Table D.14 show the statistics obtained by this pattern.

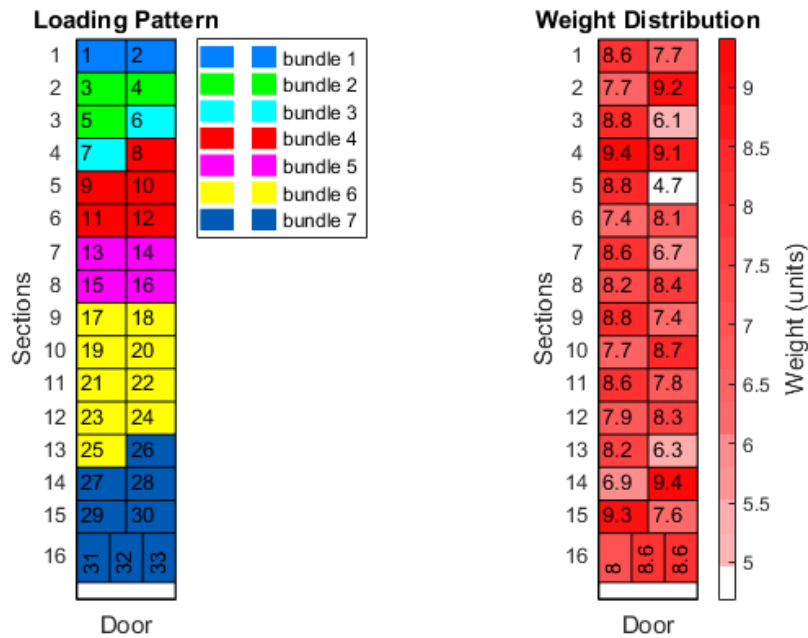


Figure 0.12: Loading pattern (efficient)

Table D.14: Selection statistics

Usable container volume (m^3)	Volume used (m^3)	Volume usage (%)	Weight (weight units)
63.36	61.28	96.72	265.65

The operator can also view the loading pattern in 3D to better perceive the final result of the loaded truck. Figure 0.13 shows the 3D loading pattern.

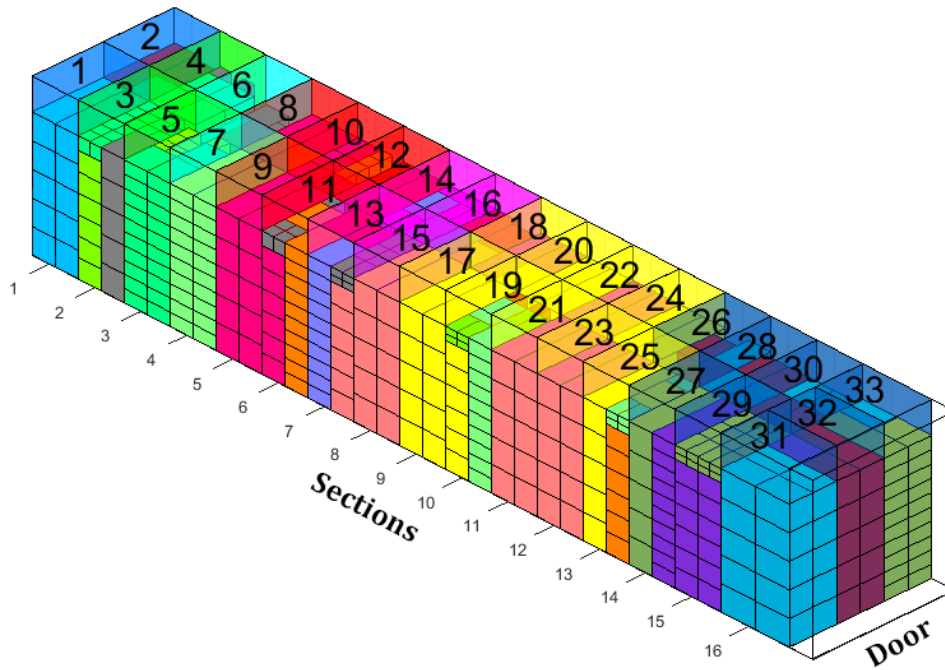


Figure 0.13: 3D loading pattern (efficient)

Additionally, the operator can view the weight distribution of this loading pattern and evaluate its stability. Figure 0.14 shows the longitudinal and transversal weight distribution of this load.

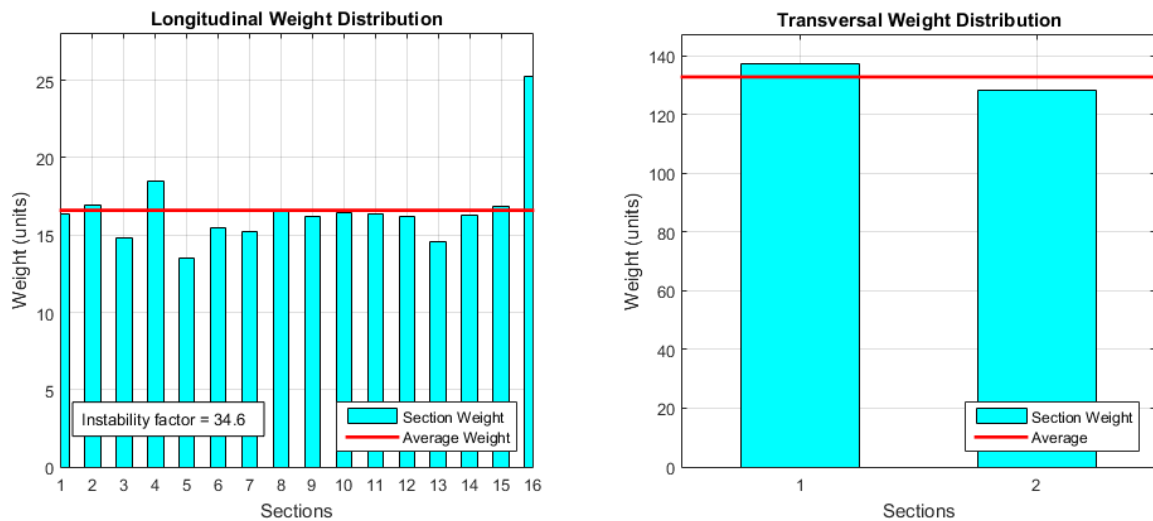


Figure 0.14: Weight distribution (efficient)

The *Instability* value provides the operator a tool to evaluate the stability of the cargo (detailed in 5.2.2.1).

If this cargo was instead going abroad, for example to Hamburg, German national laws impose the distribution of weight of the cargo uniformly across the cargo space for trucks entering the country. The heuristic can output a loading pattern that tries to distribute the weight in order to obtain a uniform distribution. However, this operation is done at the expense of the integrity of the bundles, which would require the truck to be unloaded at an intermediary warehouse (inside the country) to revert back to the efficient loading pattern.

Figure 0.15 shows the uniform loading pattern.

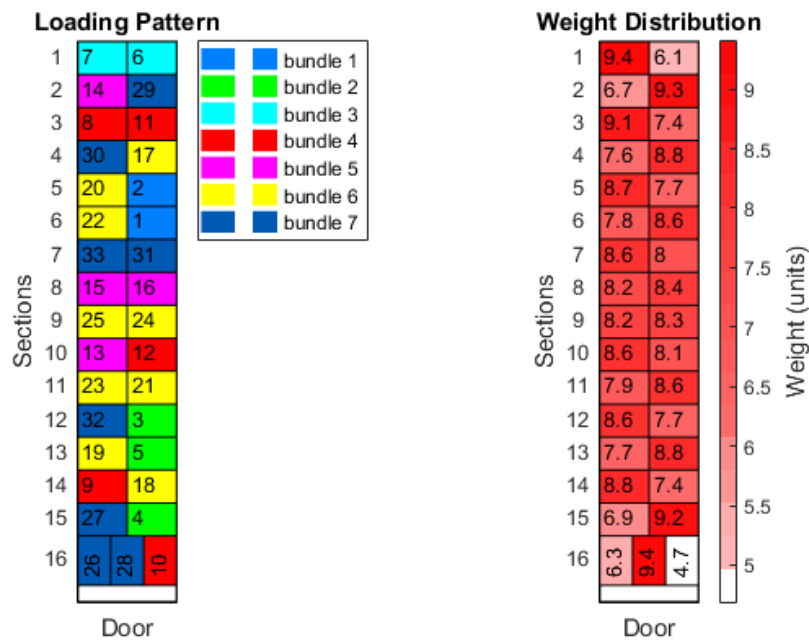


Figure 0.15: Loading pattern (uniform)

Like previously, this pattern can be viewed in 3D (as shown in Figure 0.16).

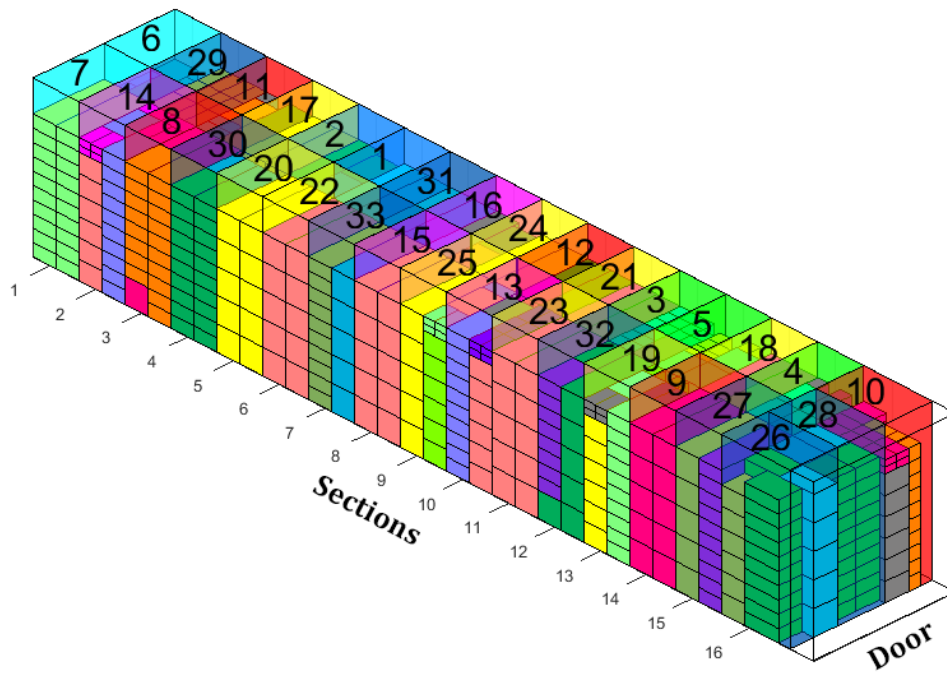


Figure 0.16: 3D Loading pattern (uniform)

The weight distribution is also presented to the operator (as shown in Figure 0.17).

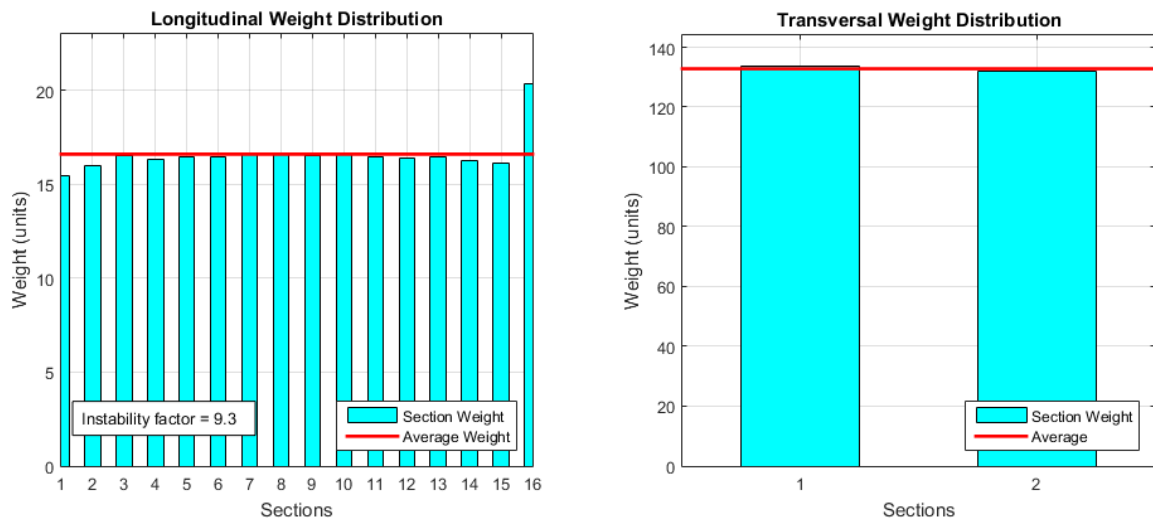


Figure 0.17: Weight distribution (uniform)

