ABSTRACT

Title of dissertation: COMPUTATIONAL FOUNDATIONS FOR SAFE AND EFFICIENT HUMAN-ROBOT COLLABORATION IN ASSEMBLY CELLS

Carlos Morato, Doctor of Philosophy, 2016

Dissertation directed by: Professor Satyandra K. Gupta
Department of Mechanical Engineering

Human and robots have complementary strengths in performing assembly operations. Humans are very good at perception tasks in unstructured environments. They are able to recognize and locate a part from a box of miscellaneous parts. They are also very good at complex manipulation in tight spaces. The sensory characteristics of the humans, motor abilities, knowledge and skills give the humans the ability to react to unexpected situations and resolve problems quickly. In contrast, robots are very good at pick and place operations and highly repeatable in placement tasks. Robots can perform tasks at high speeds and still maintain precision in their operations. Robots can also operate for long periods of times. Robots are also very good at applying high forces and torques. Typically, robots are used in mass production. Small batch and custom production operations predominantly use manual labor.

The high labor cost is making it difficult for small and medium manufacturers to remain cost competitive in high wage markets. These manufactures are mainly

involved in small batch and custom production. They need to find a way to reduce the labor cost in assembly operations. Purely robotic cells will not be able to provide them the necessary flexibility. Creating hybrid cells where humans and robots can collaborate in close physical proximities is a potential solution. The underlying idea behind such cells is to decompose assembly operations into tasks such that humans and robots can collaborate by performing sub-tasks that are suitable for them.

Realizing hybrid cells that enable effective human and robot collaboration is challenging. This dissertation addresses the following three computational issues involved in developing and utilizing hybrid assembly cells:

- We should be able to automatically generate plans to operate hybrid assembly cells to ensure efficient cell operation. This requires generating feasible assembly sequences and instructions for robots and human operators, respectively. Automated planning poses the following two challenges. First, generating operation plans for complex assemblies is challenging. The complexity can come due to the combinatorial explosion caused by the size of the assembly or the complex paths needed to perform the assembly. Second, generating feasible plans requires accounting for robot and human motion constraints. The first objective of the dissertation is to develop the underlying computational foundations for automatically generating plans for the operation of hybrid cells. It addresses both assembly complexity and motion constraints issues.

- The collaboration between humans and robots in the assembly cell will only be practical if human safety can be ensured during the assembly tasks that

require collaboration between humans and robots. The second objective of the dissertation is to evaluate different options for real-time monitoring of the state of human operator with respect to the robot and develop strategies for taking appropriate measures to ensure human safety when the planned move by the robot may compromise the safety of the human operator. In order to be competitive in the market, the developed solution will have to include considerations about cost without significantly compromising quality.

- In the envisioned hybrid cell, we will be relying on human operators to bring the part into the cell. If the human operator makes an error in selecting the part or fails to place it correctly, the robot will be unable to correctly perform the task assigned to it. If the error goes undetected, it can lead to a defective product and inefficiencies in the cell operation. The reason for human error can be either confusion due to poor quality instructions or human operator not paying adequate attention to the instructions. In order to ensure smooth and error-free operation of the cell, we will need to monitor the state of the assembly operations in the cell. The third objective of the dissertation is to identify and track parts in the cell and automatically generate instructions for taking corrective actions if a human operator deviates from the selected plan. Potential corrective actions may involve re-planning if it is possible to continue assembly from the current state. Corrective actions may also involve issuing warning and generating instructions to undo the current task.

Computational Foundations for Safe and Efficient Human-Robot Collaboration in Assembly Operations

by

Carlos W. Morato

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Satyandra K. Gupta, Chair/Advisor
Professor Jaydev Desai
Professor Jeffrey Herrmann
Assistant Professor Monifa Vaughn-Cooke
Professor Amitabh Varshney (Dean's representative)

© Copyright by
Carlos W. Morato
2016

Dedication

To my wife, my daughters, my mother, my grandmother

and all my teachers.

Acknowledgments

First and foremost, my deepest gratitude to my Ph.D. advisor Professor Satyandra K. Gupta for giving me an opportunity to work with him and mentoring me as a researcher. No words are enough to express my gratitude to Prof. Gupta for the innumerable technical and non-technical conversations through both explicitly and implicitly he helped me to become a better researcher. He has always posed new set of challenges in front of me and guided me through the process through his constant support, wisdom, and moral boost. I am very grateful to him for providing me with a flexible research environment encouraging me to try out different ideas and at the same time challenging my analytical thinking.

I would also like to thank Prof. Jaydev P. Desai, Prof. Jeffrey Herrmann, Prof. Monifa Vaughn-Cooke, and Prof. Amitabh Varshney for accepting to serve in my dissertation committee. Their valuable suggestions related to my dissertation helped me a lot during my Ph.D. I thank them for being very prompt and responsive about my dissertation needs despite of having unbelievably busy academic and research commitments of their own. I would then like to thank Defense Advanced Research Projects Agency (DARPA) for funding (through my advisor) my research.

Then, I would like to thank Dr. Krishna Kaipa, for being a very good friend and a great colleague. Krishna's bast experience, persistence in research pursuit and attention to details, help me to propose and answer scientific questions. I would also like to thank him for meticulously proofreading some parts of my dissertation. I especially would like to thank Prof. David Bourne from Carnegie Mellon University

for provide valuable insights that made possible some parts of this dissertation. I am also thankful to META team at Vanderbilt University for providing assemblies used in this paper.

My deepest gratitude goes to my family without whose support and encouragement none of this could have been possible. I would like to thank my wife for her love, patience and support. I would like to thank my two little girls that make all the impossible, possible. Their voices, smiles and laughs provide energy into my life. I would like to thank my mother for her boundless love and my grandmother for her wisdom and support.

I would like to thank my former teachers Prof. D. Akin and Prof. M. Bowden from the Department of Aerospace Engineering, for opening the doors of this prestigious school of engineering. I am especially grateful to Prof. Hugh Bruck and Prof. Nikhil Chopra for encouraging me to be part of the Mechanical Engineering program and embark on the journey of Ph.D.

Then, I would like to thank my colleagues and friends Dr. Atul Thakur, Dr. Sagar Chowdhury, Dr. Petr Svec, Brual Sha, Boxuan Zhao, Jiashun Liu, Ariyan Kabir, Shaurya Shriyam and Cunbo Zhuang in Simulation-Based System Design Laboratory (SBSD) for their support, encouragement, and for creating a enjoyable work atmosphere in the laboratory.

Last but not the least I thank the Institute for Systems Research staff (Jason Strahan and Beverly Dennis) and Mechanical Engineering administrative staff (Lita Brown, Penny Komsat, Lee Ellen Harper, Amarildo DaMata, and Janet M. Woolery) who were very helpful and prompt in helping me with administrative issues and

thereby making my stay at the University of Maryland enjoyable and smooth.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Historically, industrial robots have been primarily used in mass production lines for repetitive tasks like painting and welding. However, recent advances in perception, planning, and learning have enabled their use in small batch manufacturing, involving highly non-repetitive tasks. Representative examples include kitting [8, 9], bin-picking [10, 11, 12, 13, 14], assembly [15], and cleaning [16, 17].

Assembly operations are an integral part of the overall manufacturing operation. After parts are produced, they need to be assembled together to impart the desired functionality to products. Assembly operations vary significantly based on the size and complexity of the products. Representative examples of assembly operations include [18, 19]:

- Pick and place
- Joining using threaded fasteners
- Joining using rivets
- Welding, soldering, and brazing
- Adhesive bonding

- Snap fits

Figure 1.1: *Human Assembly Cell [1]*

Human workers offer the following benefits:

- *Versatility*: Humans are able to do a wide variety of assembly operations and able to work with many different types of tools.

- *Dexterity*: Humans are able to perform assembly tasks that require complex coordinated motions in very tight spaces.

- *Ability to perform in − process inspection*: Humans are able to perform in-process inspection to reduce process error.

- *Ability to handle contingencies and recover from errors* : Humans are good at recovering from errors introduced in previous steps and handling unexpected

situations.

Human workers have the following limitations:

- *Consistency*: Humans are unable to maintain consistency over long periods of time because of physical and mental fatigue.
- *Labor cost*: Developed countries have high wages and manufactures often find it difficult to compete with countries with low wages. So labor costs are a major contributing factor to cost competitiveness.
- *Size and weight limitations*: There are natural limitations on the size and weight of parts that can be manipulated by human workers.
- *Speed*: There are natural limitations on the speed of assembly operations that can be achieved by human workers.

In mass production lines, robots are often utilized to overcome limitations of human workers. They have become popular for several different assembly operations. Representative examples include:

- Pick and place
- Welding (arc, spot, etc.)
- Bolt and fastening
- polishing
- Materials handling

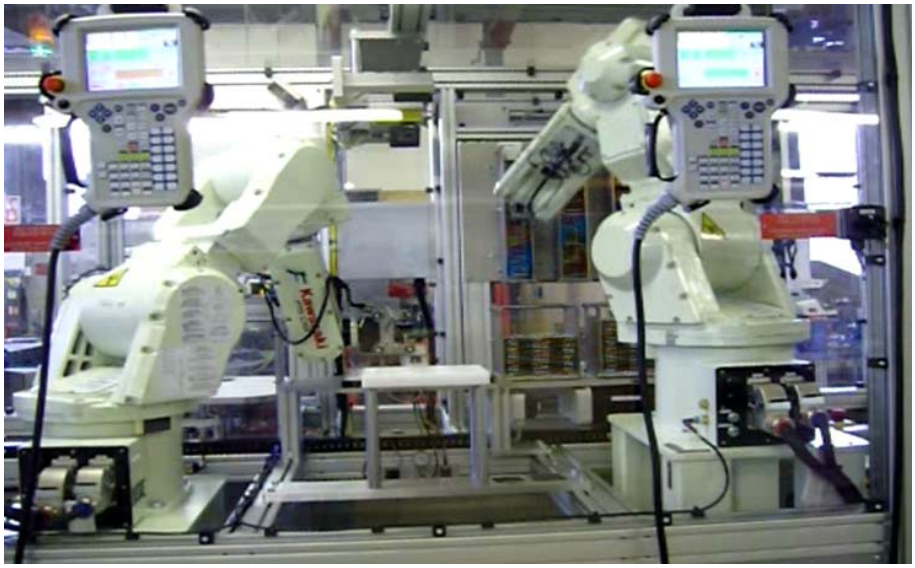


Figure 1.2: *Robotic Assembly Cell [2]*

However, currently robots have the following limitations:

- *High capital cost*: Current generation robots cost significant amount of money. So it is not possible to use robots unless utilization is extremely high.

- *Long programming time*: Currently it takes a long time to program robots for performing complex tasks. So robots cannot be used in many small and medium manufacturing operations.

- *Limited dexterity*: Current generation robots have limited dexterity. To use robotics based assembly, products need to be designed to ensure that assembly operations do not require high level of dexterity.

Current generation industrial robots impose safety risks to humans, so physical separation has to be maintained between humans and robots. This is typically accomplished by putting the robot in a cage. In order for the robot to be operational,

the cage door has to be locked and elaborate safety protocol has be followed to ensure that no human operator is present in the cage. This makes it very difficult to design assembly cells where humans and robots can collaborate effectively.

Figure 1.3: *Robotic Assembly Cell showing safety barriers around the welding robot [3]*

1.2 Motivation

The National Association of Manufacturers (NAM) defines small manufacturers as companies with 500 or fewer employees and medium-sized manufacturers as companies with 2,500 or fewer employees. The NAM estimates that the US has close to 300,000 small and medium manufacturers (SMM), representing a very important segment of the manufacturing sector. As we move towards shorter product life cycles and customized products, the future of manufacturing in the US will depend

upon the ability of SMM to remain cost competitive. However, SMM in the US have largely stayed away from using industrial robots.

The high labor cost is making it difficult for SMM to remain cost competitive in high wage markets. They need to find a way to reduce the labor cost. Clearly, setting up purely robotic cells is not an option for them.

Recently several advances have been made in industrial robots that make them safer for humans [20, 21, 22] and hence presenting an opportunity for creating hybrid cells where humans and robots can collaborate in close physical proximities. The underlying idea behind such cells is to decompose assembly operations into tasks such that humans and robots can collaborate by performing tasks that are suitable for them.

Humans are very good at perception tasks in unstructured environments. They are able to recognize and locate a part from a box of miscellaneous parts. They are also very good at complex manipulation in tight spaces. Humans also excel at visual inspection tasks. The sensory characteristics of the humans, motor abilities, knowledge and skills give the humans the ability to react to unexpected situations and solve problems with, in some cases, minimal information.

In contrast, robots are very good at pick and place operations and being highly repeatable in placement tasks. Robots can perform tasks at high speeds and still maintain precision in their operations. Robots can also operate for long periods of time and preserve the properties mentioned above. Robots are also very good at applying high forces and torques.

Realizing hybrid cells that enable effective human-robot collaboration (HRC)

is challenging due to the following reasons:

- Humans and robots will be working in close proximity of each other in hybrid cells. Humans might forget about the move planned by the robot. So they might accidentally come in the way of the robot. Robots will need to make sure that they avoid collision with humans by taking an appropriate collision avoidance strategy.

- Usually it takes a long time to program robots. In order for hybrid cells to be utilized in small and medium manufacturing operations, we will need to develop planning approaches that can automatically program robots.

- Humans are flexible but also prone to making errors and doing operations in a different manner. If robots do not react to human errors appropriately, then the system can be highly inefficient and it may take a long time to recover from errors. Robots should be able to re-plan in response to an unpredictable human behavior and modify their motion according to the new plan. They should be able to communicate with the human as well to point out the error.

This dissertation will attempt to develop computational foundations for addressing the above described challenges.

1.3 Representative Model of One-Human One-Robot Hybrid Assembly Cell

We will use a representative model of one-human one-robot to explain the research issues being investigated in this dissertation. We assume that the human and robot will collaborate to assemble a product. The cell will operate in the following manner.

- The cell planner will generate a plan that will provide instructions for the human and the robot in the cell.
- Instructions for the human operator will be displayed on a screen in the assembly cell.
- The human will be responsible for retrieving parts from bins and bringing them within the robot workspace.
- The robot will pick up parts from its workspace and assemble them into the product.
- If needed, the human will perform the dexterous fine manipulation to secure the part in place in the product.
- The human and robot operations will be asynchronous.
- The cell will be able to track the locations of parts, the human, and the robot at all time.
- If the human operator makes a mistake in following an assembly instruction, re-planning will be performed to recover from that mistake. As a part of the

re-planning process, appropriate warnings and error messages will be displayed in the cell.

- If the human come too close to the robot to cause a collision, the robot will perform a collision avoidance strategy.

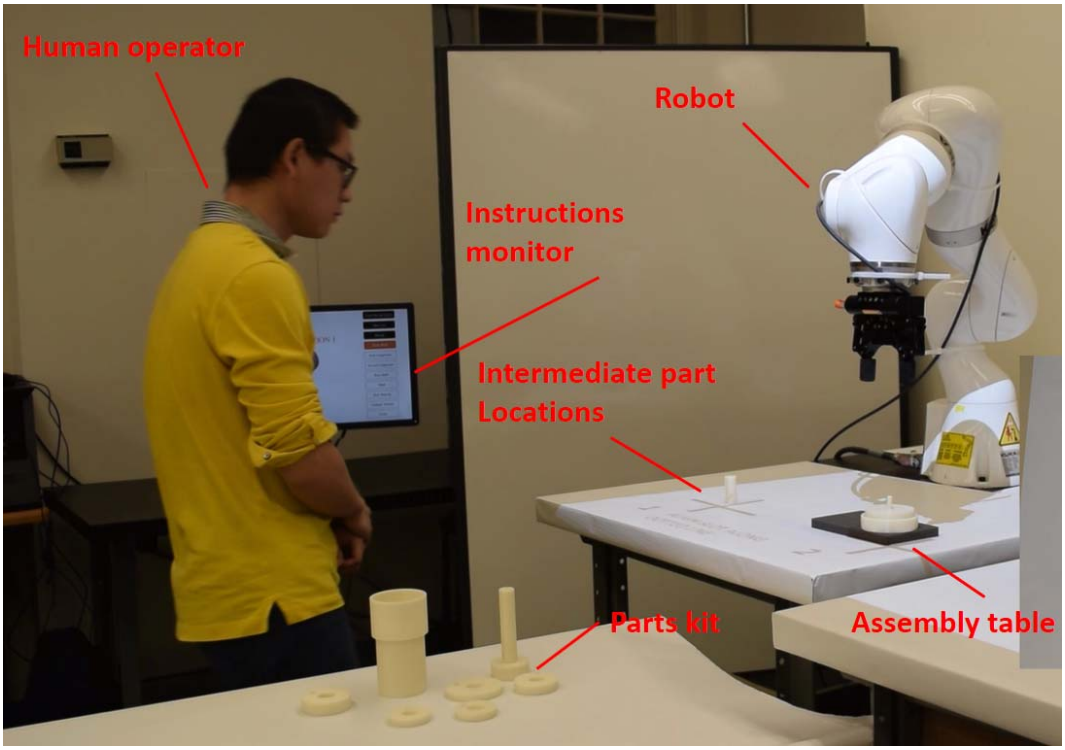


Figure 1.4: *Hybrid Assembly Cell*

1.4 Research Objectives

This dissertation address the following three computational issues involved in developing and utilizing hybrid assembly cells:

- *Automated planning for hybrid assembly cell operation*: We should be able to automatically generate plans to operate hybrid assembly cells to ensure

efficient cell operation. This requires generating feasible assembly sequences and instructions for robots and human operators, respectively. Automated planning poses the following two challenges. First, generating precedence constraints for complex assemblies is challenging. The complexity can come due to the combinatorial explosion caused by the size of the assembly or the complex paths needed to perform the assembly. Second, generating feasible plans requires accounting for robot and human motion constraints. The first objective of this dissertation is to develop the underlying computational foundations for automatically generating plans for the operation of hybrid cells. It will address both assembly complexity and motion constraints issues.

- *Monitoring the state of the human operator in the assembly cell to ensure safe operation*: A critical issue that is hampering the entry of humans into traditional robotic environments is safety. The cooperation between humans and robots in the assembly cell will only be practical if human safety can be ensured during the assembly tasks that require collaboration between humans and robots. The second objective of this dissertation is to evaluate different options for real-time monitoring of the state of human operator with respect to the robot and develop strategies for taking appropriate measures to ensure human safety when the planned move by the robot may compromise the safety of the human operator. In order to be competitive in the market, the developed solution will have to include considerations about cost without significantly compromising quality. Hence cost considerations will play a major role in the

selection of technology option for human monitoring.

- *Monitoring the state of the assembly cell and contingency planning*: In the envisioned hybrid cell, we will be relying on human operators to bring the part into the cell. If the human operator makes an error in selecting the part or placing it correctly, the robot will be unable to correctly perform the task assigned to it. If the error goes undetected, it can lead to a defective product and inefficiencies in the cell operation. The reason for human error can be either confusion due to poor quality instructions or human operator not paying adequate attention to the instructions. In order to ensure smooth and error-free operation of the cell, we will need to monitor the state of the assembly operations in the cell. The third objective of this dissertation is to develop algorithms to identify and track parts in the cell and automatically generate instructions for taking corrective actions if a human operator deviates from the selected plan. Potential corrective actions may involve re-planning if it is possible to continue assembly from the current state. Corrective actions may also involve issuing warning and generating instructions to undo the current task.

Chapter 2

Literature Review

This chapter[1] presents literature related to the work described in this dissertation. The topic of this dissertation is at the intersection of Manufacturing, Robotics, and Perception.

In Section 2.1, we present different approaches for automatically generating assembly sequences for assemblies with non-deformable parts. Assembly sequence planning is a vast area of research where extensive work has been done during the past three decades. Our literature survey presents several key approaches in the assembly sequence planning field and describes methods that are closely related to the techniques presented in this dissertation.

Section 2.2 deals with different approaches for safety when humans and robots interact and work together sharing the same work space. We survey existing literature that characterizes safe HRC in two broad areas: Pre-collision and Post-collision. The Pre-collision problem deals with devising control strategies that allow the robot to prevent an imminent collision with a human operating in its proximity, and the Post-collision problem aims to reduce the impact/injury after an unexpected collision has occurred between the robot and the human. As our approach falls into the former category, we discuss methods related to pre-collision strategy.

In Section 2.3 we survey relevant human tracking technologies. One of the main

[1] Work in this chapter is derived from the published work in [23, 24, 25, 26, 27, 28, 29].

requirements for an efficient and safe HRC in assembly operations is the ability to localize the human operator. This section describes the main features of the relevant human tracking systems.

In Section 2.4 we draw inspiration from intelligent tutoring systems to develop a robust contingency handling approach for hybrid assembly cells. In this section, we survey literature at the intersection of 3D-part recognition and knowledge based systems to generate reactive behavior to deal with situations when humans consciously or unconsciously modify the the assembly plan.

2.1 Assembly Sequence Planning

Research in automated assembly sequencing has rapidly increased over the past few decades [30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. The problem of finding a valid assembly sequence for general cases that allow complex combination of motions was shown to be impractical, primarily owing to the issue of combinatorial explosion ([40, 41, 42]). This led to an increase in the number of assumptions made in order to address restricted cases of the assembly sequence planning. Examples include monotonic assembly sequences (each assembly operation leads to the part being in the location in the assembly) and two-handed assembly sequences (each assembly operation merges exactly two assembly parts or components).

Assembly sequence generation is considered one of the most important problems in Assembly Planning. Algorithms to solve the assembly sequence follow a combinatorial approach if the the operation precedence constraints are known in

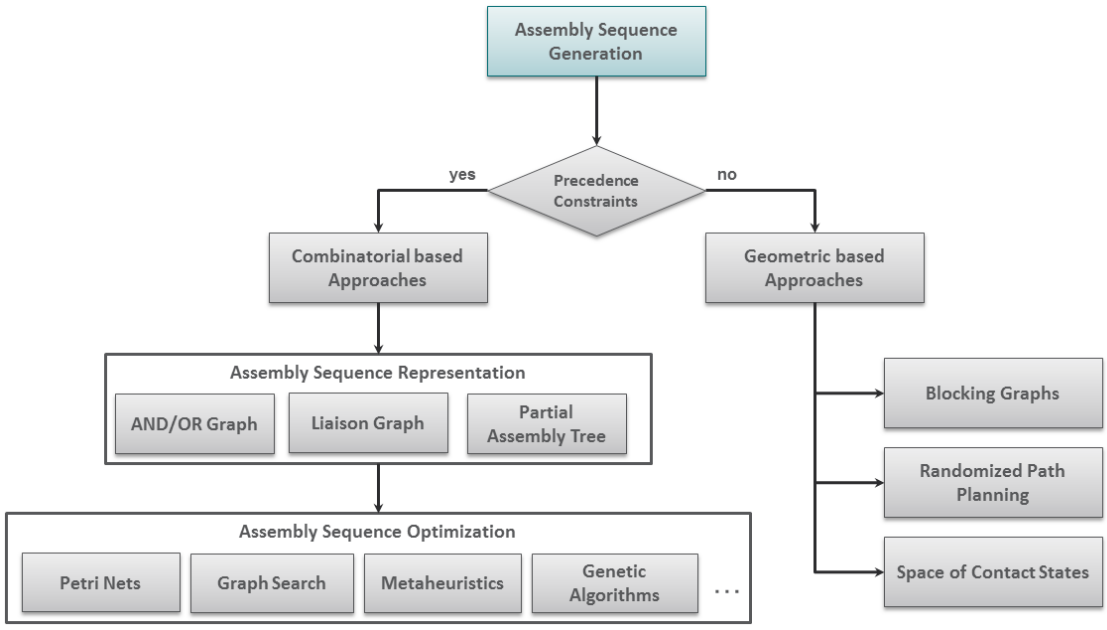


Figure 2.1: *Taxonomy of Assembly Sequence Planning approaches.*

advance. If the precedence constraints are unknown, then the assembly sequence is generated by algorithms that follow a geometric approach to figure out the feasibility of the assembly operation. Figure 2.1 shows the taxonomy of different assembly sequence planning approaches in the form of a tree containing three hierarchical levels. The solid lines denote direct inheritances of different techniques from parent classes.

AND/OR Graph, Liaison Graph or Partial Assembly Trees are some of the algorithms that can be used as an assembly sequence representation if the knowledge of the precedence constraints/relationship is known. In this case, the goal is to find the optimal assembly sequence using as established techniques such as Petri Nets, Graph search, Genetic Algorithms, Ant Colony Optimization or Simulated Annealing among others.

The major issue with the combinatorial approaches is that the precedence

constraints are determined by the user who may not completely capture all constraints accurately. If the knowledge of the precedence constraints is not present then simply following a combinatorial approach is not feasible. Algorithms such as blocking graphs, randomize path planning or Space Contact states among others must therefore be used in such circumstances. These algorithms follow a geometric approach in order to assess the feasibility of the assembly operation.

The assembly process starts from an initial configuration in which all the parts are in a completely disassembled state and finishes with a goal configuration that corresponds to the final assembly. An assembly sequence plan specifies the order in which each part and/or subassembly must be inserted into an incrementally expanding subassembly that eventually leads to the final assembly. The components to be assembled may be quite different from each other in terms of component geometry, precedence, accessibility, and other types of constraints. The assembly sequence planning for a 3D assembly, with large number of parts and complex assembly relationships between its individual parts is a large-scale combinatorial problem ([43, 44]). While the associated constraints play an important role in limiting the number of sequences ([40]) there are numerous combinations of feasible assembly sequences that are valid ([31]). Among valid sequences are ones that optimize the assembly process with respect to one or more criteria ([45, 46, 47, 48]).

One of the early approaches of representing the assembly components based on the liaisons diagram was proposed by [49] and consists of a series of yes-no questions for each disassembling operation were answered the user. The geometric feasible assembly sequences were reasoned out based on the set of answers given

by the user. Bourjault proposed the concept of assembly precedence relations and applied these assembly precedence relations to express the precedence constraints among parts or liaisons in an assembly. The approach was focused on geometric feasible assembly sequence. This sequence represents an operation order by which components are assembled without geometric intersections.

In [30] De Fazio & Withney proposed an augmented version of the liaison graph which captures dimensional constraints between one or more degrees of freedom between the parts. The liaison-sequence analysis method allowed showing one or more favorable assembly sequences. The main difference relies in the number of questions needed to be answered in order to allow algorithmic generation of assembly sequences. On the other hand, the simplified generation of assembly sequence had similar limitation to [49].

To simplify the querying process, Homem de Mello and Sanderson [31] proposed the cut-set analysis method with three simplification rules. They developed a complete algorithm that generates all the possible assembly sequence for a specific product. In order to achieve this goal the algorithm used a relational model where all the connections between parts are explicitly described. The problem of generating an assembly sequence was transformed into a problem of generating a disassembly sequence. Inversing the problem is feasible under the geometrical assumption that a part can be placed in the same way as it is taken away from the assembly.

It was also assumed that two parts are connected or joined each time, and that while the parts are joined with more than one part forming a subassembly the contacts between the parts in that subassembly are established in the relational

model. The relational representation allows the model to represent different types of contacts between parts such as planar surface, cylindrical shaft, cylindrical hole, polyhedral shaft, polyhedral hole, etc. At the same time, it allows it to also have a detailed description of the types of attachments such as glue attachment, pressure fit, clip, screw, etc.

The algorithm returns the AND/OR graph representation of assembly sequences. The algorithm is complete and correct based on the assumption that it is always possible to decide correctly whether two subassemblies can be joined based on geometrical and physical criteria. Most of these geometrical and physical criteria were introduced by the human expert.

Wilson and Latombe [50] determined that assembly sequence planning approaches either had either too large a quantity of easy questions (as yesno queries) or a few number of very complex questions (as what queries) that caused problems in the assembly planning process. Wilson proposed an improved cut-set analysis approach. When the disassembly feasibility for a cut-set does not exist, the system allows the user to indicate the interference parts by referring to the CAD data. Wilson found that it was not very complicated to answer the questions and the quantity of queries can be reduced further.

Assembly precedence relations inferring methods can be classified into two types, direct and indirect. The yesno query-based method and the simple cut-set method are indirect approaches. These approaches start the reasoning process with the liaisons diagram instead of the assembly structure. The user needs to answer quite a number of questions to tell what the assembly looks like given that the

liaisons diagram is too simple to represent the assembly structure.

Wilsons method was supported by the CAD data and is classified as a direct/indirect mixed approach. Direct approaches start with the assembly design itself. Approaches based on the contact and interference vectors are also considered direct methods. Direct approaches need more computational efforts compared with the fully explored indirect approaches.

Wilson introduced the notion of a Non-Directional Blocking Graph as a result of the analysis of the blocking relations (interferences) among the parts in the assembly. This representation described the combinatorial set of parts and its characteristics where geometrical constraints among parts change. Moreover, complexity measurements were presented in order to evaluate the complexity of the product and therefore the quality of the solution. Those characteristics and the idea that the blocking relations are computed and fixed based on subdivision of translations into finite cells makes this approach very difficult to implement because of its sensitive to floating point approximations. The complexity grows exponentially with the space dimensions of allowed motions.

To be more efficient, Dong et al. [51] designed a knowledge-based Assembly Sequence Planning approach in which the assembly is represented as a Connection-Semantics-Based Assembly Tree and a knowledge-based assembly sequence planning system was developed. Dong method was based on the analysis of contact and interference vectors defined in the orthogonal directions. The vector representation allows a very straight forward computation of the geometric feasible assembly sequence. However, it is only valid for orthogonal assemblies in which all the components must

be assembled along with the orthogonal directions.

Niu et al. [52] proposed an advanced direct approach that directly starts with the mating relation graph derived from the component CAD model. However, the approach is also limited to the orthogonal directions. The need of a more powerful direct assembly sequence planning (ASP) approach was also discussed, where the discovery process of the correct and complete assembly precedence constraints is proposed to be more efficient and without the limitations mentioned above. Inspired by this consideration, a hierarchical direct ASP approach was proposed. The ASP reasoning process was based on a comprehensive assembly model, in which the assembly draft, the liaisons diagram, the assembly tree structure, the part parameters, and the liaison parameters are all included [37, 38].

The differentiating factor in these approaches became the CAD information and how this information was used. Approaches that exploit geometric information to perform randomized path planning became more popular. In contrast, approaches that used the spatial representations of all the parts in order to determine blocking relations between them, as we detailed previously, became less practical for large and complex assemblies.

In this context, Wilson [50] used the C-Space representation of any part as the set of all possible values of the degrees of freedom. Using the similar rationale the obstacles (C-obstacles) were represented as a subset of the C-space. Wilson developed an interference diagram to determine collision free path (C-free= C-Space C-obstacles) for an assembly by using this representation. A similar representation was used and later extended in [7] by La Valle. La Valle built a roadmap based on

local planners and then applied graph search on this roadmap to find the solution path.

Local methods are based on the definition of a potential function where the maximum of the function is in the surface of the obstacles and the minimum is located at the goal position. This generates a difficulty of defining a potential function with a single minimum. It is therefore easy for these methods to get stuck in local minima. To avoid these minima, the potential function can guide the steering angle instead of the position of the mobile part, and creates a potential that depends on the position and orientation of the mobile component.

In [53] a rapid-growing random tree-based based motion planning technique RRT is used in order to sole the disassembly sequence planning. The RRT approach tries to find a collision-free trajectory for an object that goes from an initial configuration to a goal configuration. This work assumed that the environment is static and that the only part moving is the selected part to be disassembled. The problem is reduced to compute a collision-free path among the obstacle objects. The performance of the algorithm relies in the required resolution for a good enough voxel or octree-based spatial partition subdivision. Complex shapes in part assemblies that belong to crowded assemblies would require a high resolution. On the other hand, the use of high resolution affects the algorithms performance because of memory overload in the case of voxels or computation time in the case of octrees.

This section presents several key approaches in the field and describes methods that are closely related to the techniques presented in this dissertation. Survey papers such as [54, 55, 56, 57, 58] present a more comprehensive coverage of the

field.

2.1.1 Assembly-by-disassembly and Graph Theory based Approaches

Woo et al. [59] used the "onion peeling" approach and introduced connection graphs for disassembly analysis. Later, this approach was used by [60] to derive optimal disassembly sequences. Chen et al. [61] used the onion peeling approach for parallel disassembly of components. Most existing methods are based on graph theory and involve additional information such as contact analysis, fastener matrix, disassembly precedence matrix, etc. [54, 55, 57]; standard graph search algorithms can be applied to graph representations of assembly structures in order to determine feasible or optimal assembly sequences [62]. Wilson [50] introduced non-directional blocking graph, a compact data structure to represent a combinatorial set of part intersections for an assembly, which implicitly contains the geometric constraints [63]. Romney [64] extended this approach into their Stanford Assembly Analysis Tool (STAAT). Romney's work mainly focused on developing geometric assembly planning models rather than on optimization aspects. Khosla and Roy developed an assembly sequence method from a 3D CAD model and exact geometry representation based face adjacency graph. This method used data related to the instances of parts and subassemblies to generate assembly sequences [60] *IS THIS THE RIGHT REFERENCE*. Large number of parts in this representation may lead to very expensive geometric tests with the costs of nearest neighbor function and the collision checking procedure representing the major bottle necks in performance ([65]). A

method to reduce spatial representation and computational time to perform assembly simulation was introduced by [66].

2.1.2 Artificial Intelligence based Approaches

De Mello and Sanderson formulated the assembly sequence problem as a discrete search and optimization problem. They proposed AND/OR graphs to represent precedence relations between parts [30, 31]. Following this approach, a variety of methods were proposed using knowledge based systems [67], genetic algorithms where each possible assembly sequence is represented in the initial populations [68, 69, 70, 71], memetic algorithms [72], neural networks with self-organized maps and optimal nets [73, 74], fuzzy sets [75, 76], data mining [77], Bayesian networks [78], and simulated annealing [79, 80].

2.1.3 Motion Planning based Approaches

Motion planning based approaches were developed for simulating assembly operations [81, 82, 83]; the goal was to perform assembly operation planning that includes tool uasge planning, and task planning. Motion planning based on randomized methods have became popular in robot path planning applications. Algorithms such as probabilistic roadmaps (PRM) ([84]) and RRT ([7]) have been used to solve high degrees of freedom (DOF) motion planning problems. However, current motion planners may not work well on part disassembly due to highly constrained environments that require the generation of a finer motion of the parts. [85] described

the disassembly planning problem as a repeated occurrence of "narrow" passages in the Configuration space that makes it impractical to use PRM for these problems. However, [86] performed assembly sequence analysis based on RRT by using collision models. They also showed the complexity of the domain ([87]) and the possibility of improvements to the representation. In the similar direction, [88, 89] proposed some strategies to improve the performance of motion planning methods to solve the assembly sequencing problem.

2.1.4 Subassembly Detection Based Approaches

[90] formulated one of the first approaches to detect subassemblies based on a mathematical model of the product. The model was a function of the product's interference, contact, and connection matrices. [91] described a procedure to automatically derive the feasible assembly sequence and detect the subassemblies for automobile body assembly. The procedure is based on the definition of a connection matrix and a contracted matrix. These two matrices represent the precedence constraint knowledge among components and sub-assemblies. [92] constructed a hierarchical assembly system and used a matrix operation to generate assembly sequences. Later, [93] extended this assembly modeling to generate feasible assembly sequences automatically. However, in all these cases, a complete description of the assembly was needed.

2.2 Human-Robot Collaboration

When the collaboration between robots and humans is performed in close proximity, a pre-collision analysis is required for contingency handling. Contingency handling strategies have been mainly researched w.r.t. mitigating safety risks for human operators during collaboration with robots. They can be broadly divided into two categories: pre-collision [94, 95, 96, 97, 98, 99, 100, 101] and post-collision [102, 103, 104]. The former problem deals with devising controllers that allow the robot to prevent imminent collisions with a human. However, the latter aims to reduce the impact/injury after an unexpected human-robot collision has occurred. Our focus is on the first category. The underlying principle of most pre-collision methods consists of tracking the physical separation between the robot and the human and enabling the robot to take preventive actions whenever the separation is below a specified threshold.

We have identified two families of pre-collision approaches that significantly differ from each other in their underlying philosophies and, consequently, in their implementation techniques. The first line of research direction treats the problem in a two dimensional Euclidean space by working with the projections of the human and the robot onto a 2D range-image plane [100, 95, 96, 105]. The second approach analyzes the problem directly in a three dimensional Euclidean space by using explicit 3D models for the human and the robot [94, 106]. We briefly describe these approaches in this section.

2.2.1 Interaction analysis in 2D Euclidean space

Schiavi et al. [100] presented an approach to generate safe robot motion goals based on human presence/position detection in the work cell. The intersection between the robot and the human was determined based on analysis in a 2D plane. The human was not explicitly modeled. Instead, it was treated as a general moving obstacle and a corresponding depth image was generated by using a stereo camera based range sensing system. The robot's 3D-occupancy[2] w.r.t. the global reference frame was computed from its 3D CAD model and kinematics. Next, the occupancy data was projected onto the camera image plane, giving rise to the depth image of the robot. Now, an intersection between the two projections was used as a necessary condition for a collision between the robot and the obstacle. That is, the robot and the obstacle are physically separated in 3D, if their respective projections on the image plane don't intersect. However, if the projections intersect with each other, then there is a possibility that the robot and the obstacle are in collision or may collide with each other in the near future. A sufficient condition was used to evaluate this possibility: There exists at least one pixel in the overlapping region of the image plane at which the depth to the robot is greater than or equal to that of the obstacle (within a small margin of safety). The authors presented a test result, in which a physical robot, commanded to move from one point to another, used the proposed method to safely navigate around a human hand and reach the target configuration.

In this work, a single depth sensor is used to monitor the environment, which leads to lack of information in the blind zones of the sensor. Moreover, when parts

[2] collection of all points in the work cell that are occupied by the robot

of the obstacle are occluded by the robot, the obstacle depth information at the corresponding pixel locations is not available, which could lead to a system failure. In order to address the problem of occlusions, Flacco and De Luca [95] extended the approach in [100] to multiple depth sensors. The collision detection performance was maximized by solving an optimal sensor placement problem that was formulated by using a probabilistic framework. In particular, they decomposed the work cell into discrete cells and derived expressions for probabilities of each cell falling in occlusions and unobserved regions as a function of pose parameters of the sensors. Now, a cost function, to be minimized for optimal sensor placement, was defined as a weighted sum of the derived probabilities. The authors used numerical simulations to compute optimal sensor placements for the cases of one, two, and three sensors. However, their work was limited to a theoretical treatment and computer simulations. No physical experiments were used to evaluate the efficacy of their approach. Later, Flacco et al. [96] presented a slightly different approach, in which the distances between the robot and the obstacles were computed directly from depth data obtained from a Kinect based range sensor, instead of projecting the depth data into a robot-oriented space. These computed distances were then used in a potential field based technique that allowed the robot to avoid collisions with humans and other moving obstacles. The authors reported results from physical experiments in which a 7 DOF KUKA Light-Weight-Robot IV safely avoided collisions with a human in the work cell.

2.2.2 Interaction analysis in 3D Euclidean space

Balan and Bone [94] addressed the human-robot collision problem by using sphere-based geometric models for the human and robot. Their algorithm selected search directions that balanced between the two objectives of robot approaching the target configuration and maximizing its distance to the human throughout its motion. The robot's motion was predicted by using a transfer function model of its time response at the joint level. The human's motion was predicted at the sphere level by using a weighted mean of past velocities. As a test scenario, the authors developed a simulation of a human walking towards a moving Puma robot arm. The authors used captured human motion data to create a realistic animation. They used Monte Carlo simulations, consisting of 1000 random human walking paths passing through the robot workspace, to validate their approach. However, no real robot experiments were conducted.

Najmaei and Kermani [99, 106] also addressed the human-robot collision problem by incorporating explicit 3D modeling of the human into their approach to safe HRC. For this purpose, they developed floor mat, a sensing system comprising a grid of nodes that got activated under human body weight. The human localization was derived based on which clusters of nodes were activated as a function of time. This information, along with the average human body dimensions, was used to obtain a human model, which was then represented as a moving obstacle in the human-robot interaction framework.

2.2.3 Observations

In all the above approaches to safe HRC that used range or camera based systems to detect humans, the human-robot separation was analyzed in a 2D Euclidean space by using the depth information extracted from the camera images. However, our approach performs the analysis in a 3D Euclidean space, similar to [94, 106], by working with an explicit 3D human model generated from Kinect measurements and a forward 3D simulation of the robot's motion in a physics-based virtual environment. Whereas the 2D based approaches discussed above were proposed to overcome the speed limitations of 3D space analysis based techniques [100], we show that our approach, which belongs to the latter category, still achieves satisfactory real time performance. Also, we develop a multiple Kinect based framework in order to take care of occlusions as opposed to using a single sensor as done in previous work [100, 96].

2.3 Human Tracking Technologies

Motion tracking systems can track a moving object. A human motion tracking system needs to be self-contained, complete, accurate, fast, robust to occlusions, and wireless. Human motion capture systems are based on sensor technologies [107]. These technologies can be classified as mechanical, magnetic, optical and inertial. The advantages and disadvantages of the motion capture systems are related to the sensor's physical properties. In this context the human tracking technologies can be classified as magnetic, mechanical, inertial and optical motion capture systems.

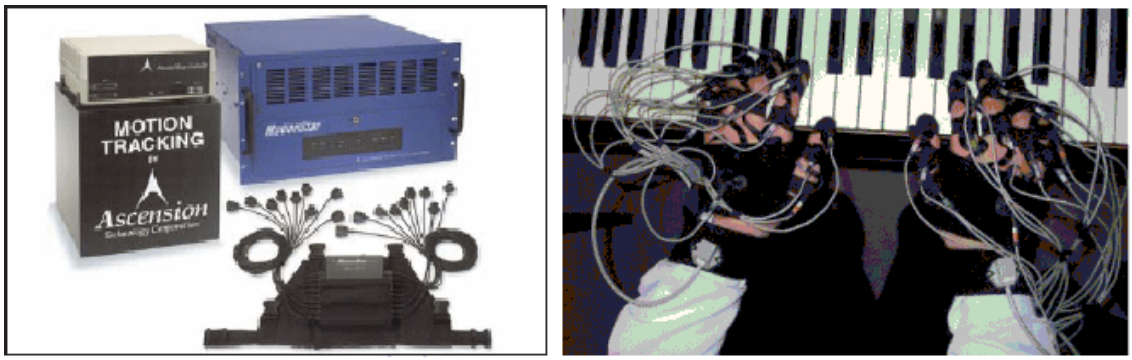


Figure 2.2: *(a) Magnetic Motion capture systems. (b)Hand Motion capture system using Liberty to learn piano playing skills from human experts [4]*

Magnetic motion capture systems uses sensors that measure the direction of the magnetic field vector belonging to the local sensor. *Liberty* [108] and *MotionStar* [109] are the most representative systems using magnetic sensors. *Liberty* uses 16 sensors with a resolution of 0.046cm for translation and 0.035 degree for rotation with an update rate of 240Hz. Meanwhile *MotionStar* contains 18 sensors with a resolution of 0.0762 for translation and 0.1 degree for rotation with an update rate of 120Hz. Magnetic sensor-based motion capture systems compute absolute position and orientation in real-time related to the magnetic source. These systems are very sensitive to the presence of noise, for example noises that are generated by ferromagnetic materials or electrical devices that are present in the environment.

Mechanical motion capture systems are composed of wearable articulated rigid links known as exoskeletons. In these systems, human motion is tracked and translated through the interconnected electromechanical transducers residing in the links located over different limbs and parts of the humans body. Human motion generates variations in the electrical signals that are then translated to relative motion between links. $Gypsy-7$ [110] 2.3 and *SARCOS Sensuit* [111] are the state-of-

the-art systems in this category. GYPSY-7 contains 15 joint sensors with an update rate of 120Hz while SARCOS Sensuit contains 35 joint sensors with an update rate of 100Hz. The advantages of this type of motion capture are: low latency, non-magnetic interference and high accuracy in the measurements. However, the use of exoskeletons for human operators can represent a drawback in limiting human mobility and exhausting human limbs for long term operations.

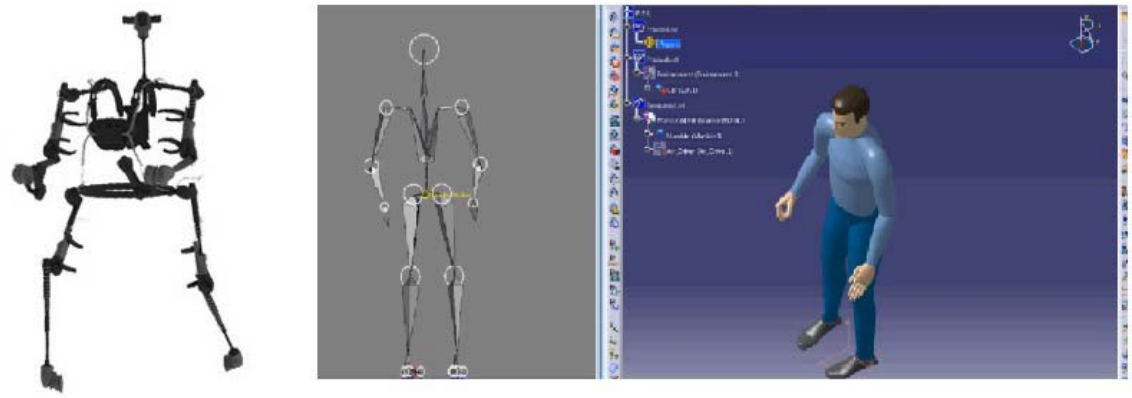


Figure 2.3: *AnimazooGypsy−7 Mechanical motion capture system based on wearable exoskeleton*

Inertial motion capture systems use inertial measurements units IMU which are composed of gyroscopes and accelerometers [112]. The IMUs are attached to the body of the human operator. Linear and angular acceleration are used to compute variations in the operators position and orientation [113, 114, 115]. The advances in micro-electro-mechanical systems technology have dramatically increased the use of these types of sensors in the field of motion tracking. The main advantages of these systems are the low latency, high update rate and their portability. However, inertial motion capture systems propagate the errors in time that result in large accumulation of errors over long tracking periods. This plus the limitation of the

IMU to return relative rotational measurements are the two main disadvantages of this type of technology.

Optical motion capture systems are based on a set of cameras. The cameras detect some descriptors/ markers and the 3D registration is made by triangulation when two or more cameras can see the same feature 2.4. The nature of a descriptor can be passive or active. A passive descriptor is made of a material that reflects the light emitted by LED rings attached around the camera. $ViconMX$ [116] motion capture system uses passive descriptors and can track a maximum number of 150 descriptor with a resolution of 0.1mm for translation and 0.15 for rotation. An active descriptor uses LEDs to emit light that is detected and tracked by the cameras. $Impulse$ [117] is a system that uses up to 120 active descriptors with an update rate of 140Hz.

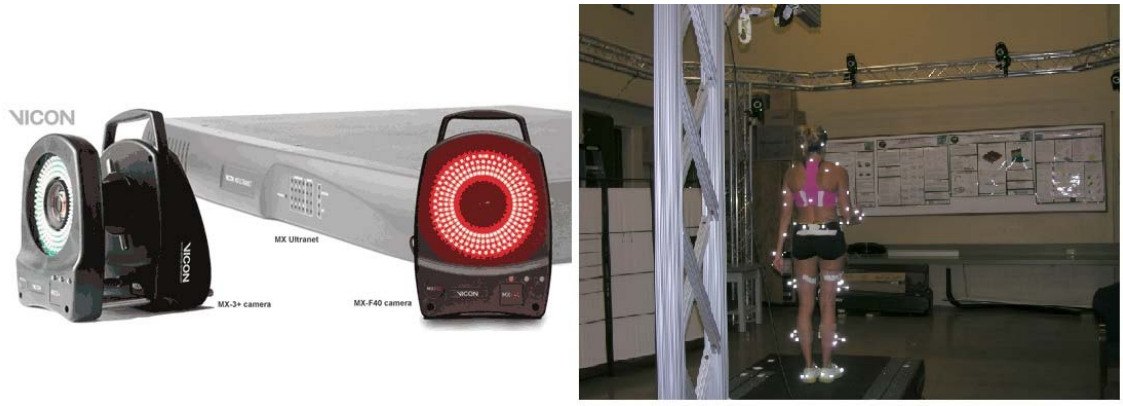


Figure 2.4: *$ViconMX$ Optical motion capture system with passive descriptors*

Descriptorless optical motion capture systems are becoming a popular human tracking technology because they do not require the human operator to wear any kind of device (active or passive). $Microsoft\ Kinect$ [118] and $iPI\ Soft$ [119] are two examples of descriptorless motion capture systems. Recently, several researchers

Technology based	Advantages	Disadvantages
Magnetic	Occlusion free Absolute position and orientation	Sensitive to magnetic distortion area boundaries proportional to the magnetic field
Mechanical	Real-time operation Very low Latency	Relative attitude Human factors
Inertial	Real-time operation Occlusion free	Relative attitude Accumulation of error
Optical (Passive Descriptors)	High precision Small and weightless descriptors	Sensitive to occlusion Calibration
Optical (Active Descriptors)	High precision descriptors correspondence	Sensitive to occlusion Calibration
Optical (Descriptorless)	descriptors free human body standard devices	Sensitive to light Pseudo-real-time Sensitive to occlusion

Table 2.1: *Motion Capture Technologies*

have used Microsoft Kinect sensors [118] for human tracking [120, 121, 122, 123, 124]. Table 2.1 summarizes some of the advantages and disadvantages of the various motion capture technologies referenced in this chapter.

2.4 Reactive re-planning for assembly operation performed by humans and robots

Primary means by which information can be delivered to human operators include speech, text, graphics [125, 126, 127, 128], virtual 3D environments [129, 130], and augmented reality [131, 132, 133]. Examples of augmented reality systems include a tracked head worn display that augments a human operator's view with text, labels, arrows, and animations [132] and laser pointer mounted on a robot highlighting where a cable must be inserted [131]. Human operators usually deliver task-specific

information to the robot either by directly teleoperating the robot or by using a graphical user interface [134].

2.4.1 Support Human Operation in the Assembly Cell

Recent advances in information visualization and human-computer interaction have given rise to different approaches to automated generation of instructions that aid humans in assembly, maintenance, and repair. Heiser et al. [125] derived principles for generating assembly instructions based on insights into how humans perceive the assembly process. They compare the instructions generated by their system with factory-provided and hand-designed instructions to show that instruction generation informed by cognitive design principles reduces assembly time significantly. The instructions generated by their automated system were limited to 2D images. Also, the authors restricted their approach to furniture assembly.

Dalal et al. [126] developed a knowledge-based system that generates temporal multimedia presentations. The content included speech, text, and graphics. The authors used a multi-stage negotiation mechanism to coordinate temporal media. They tested their multimedia generation tool by using it to update patient information to caregivers in hospitals. Zimmerman et al. [127] developed web-based delivery of instructions for inherently-3D construction tasks. The authors used quantitative and qualitative studies to examine factors like user interface, delivery technology and their influence on user interaction level and success in performing inherently

3D operations. They tested the instructions generated by their approach by using them to build paper-based origami models. Kim et al. [128] used recent advances in information visualization to evaluate the effectiveness of visualization techniques for schematic diagrams in maintenance tasks. They focused on diagram highlighting, distortion, and navigation while preserving context between related diagrams.

Several research efforts have indicated that instruction presentation systems can benefit from augmented reality techniques. Kalkofen et al. [133] integrated exploded view diagrams into augmented reality. The authors developed algorithms to compose visualization images from exploded/non-exploded real world data and virtual objects. They presented methods to restore missing hidden information in cases where there is a deficiency of information after relocating real world imagery. The authors showed how to use their approach to automatically compute task dependent layout and animation of the explosion diagrams.

Henderson and Feiner [132] developed an augmented reality system for a mechanic performing maintenance and repair tasks in a field setting. Their prototype supported military mechanics conducting maintenance tasks inside an armored vehicle turret. The system consisted of a tracked head worn display to augment a mechanic's view with text, labels, arrows, and animations. The tasks performed by a mechanic included installation and disassembly of fasteners, lights, and cables within the cramped turret. The authors carried out a qualitative survey to show that the system enabled easier task handling. Dionne et al. [129] developed a model

of automatic instruction delivery to guide humans in virtual 3D environments. The authors proposed a multi-level scheme to address issues like what kind of instructions must be presented to the user in each state and how to generate the final order of instructions.

Brough et al. [130] developed Virtual Training Studio (VTS), a virtual environment based system that allows (i) training supervisors to create instructions and (ii) trainees to learn assembly operations in a virtual environment. Their system mainly focused on cognitive aspects on the training. A survey of virtual environments-based assembly training applications can be found in [135].

2.4.2 Assembly Part Recognition

The increasing availability of low-cost, 3D sensors such as laser scanners, time-of-flight cameras, stereo cameras, and depth cameras has stimulated research in the intelligent processing of 3D data. Detecting the presence of a part and estimating its pose is related to 3D object recognition, which is a vast area of research in the computer vision community [136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150].

In the past decade, researchers were focused on designing robust and discriminative 3D features to find reliable correspondences between 3D points sets [139, 140, 142, 143, 144, 151]. Very few approaches that are available for the detection of an object based on feature correspondences [152, 153, 138] when scenes are

characterized by clutters and occlusions. In addition, the so called retrieval methods cannot deal with the presence of multiple instances of a given model. This is the particular case of bag-of-3D features methods [147, 148, 149, 150]. Tangelnder et al. present a detailed survey [154].

Feature free approach have also been developed based on the information available from depth cameras. The use of depth cameras is not new but became very popular after the introduction of the low cost Kinect technology. Kinects provide good quality depth sensors with real-time measurements. The Kinect camera uses a structured light technique [155] to generate real-time depth maps containing discrete range measurements of the physical scene. This data can be re-projected as a set of discrete 3D points (or point cloud).

Approaches based on local shape descriptors are expected to perform better [152, 153] in environments with many objects that have different shapes. However, these approaches do not work in the presence of symmetries and objects with similar shapes.

2.5 Summary

Various approaches of path planning and motion planning for robotic applications are now well-established. In particular, some of the techniques are quite optimized for less-crowded scenes. However, the slow speed and low performance in crowded scenes limit the applications in assembly sequence generation. In this dissertation we leverage the existing motion planning methods to generate assembly precedence

constraints. These precedence constraints are utilized to generate plans for operating hybrid assembly cells. Significant extensions are needed to handle assemblies that consist of a large number of parts and require complex motions to perform assemblies. We also need to be able to account for constraints of robots and humans in generating instructions.

Kinect sensors present an interesting option for developing a low cost human tracking technology. However, a single Kinect sensor is not adequate to accomplish this. We need to develop new algorithms for fusing data coming from multiple Kinect sensors. We also need to determine how to place multiple Kinect sensors to cover a given workspace. Later, tracking information can be used for ensure the safety of the human operator by integrating this information with the Assembly cell.

Methods to track the state of the assembly cell by tracking the state of the robot, parts, and the human operator have to be developed. This information then can be used to determine if the operation in the assembly cell is following the plan or not. If a deviation from the plan is detected, then we need to automatically generate contingency plans to deal with the deviation. We will need to develop new algorithms for developing contingency plans and generating instructions for executing these plans.

Chapter 3

Improving Assembly Precedence Constraint Generation by Utilizing Motion Planning and Part Interaction Clusters

In this chapter[3] , we present a technique that combines motion planning and part interaction clusters to improve generation of assembly precedence constraints. In particular, this technique automatically finds, and clusters, parts that can mutually affect each other's accessibility, and hence may impose assembly constraints. This enables generation of accurate precedence constraints without needing to examine all possible assembly sequences. Given an assembly model, our technique generates potential disassembly layers: Spatial clustering is used to generate part sets. Next, motion planning based on RRT with multiple trees is used to evaluate the interaction between these part sets. Specifically, motion planning is used to determine which part sets can be removed from the assembly. These sets are added to the first disassembly layer and removed from the assembly. Part sets that can be removed from the simplified assembly are then added to the second layer. If the process gets stuck, parts in the parent set are regrouped, and the process continues until all disassembly layers are found. The resulting structure reveals precedence relationships among part sets, which can be used to generate feasible assembly sequences for each part set and the whole assembly. We present theoretical results related to

[3] The work in this chapter is derived from the published work in [23] and [24].

the algorithms developed in this chapter. Computational results from tests on a variety of assemblies are presented to illustrate our approach.

3.1 Introduction

Assembling a complex product requires careful planning ([30, 34, 35, 50, 54, 81, 82, 55, 58]). Shapes and sizes of parts in the assembly impose restrictions on the order in which assembly operations can be performed. In order to generate a detailed assembly plan, we need to first understand precedence constraints among assembly operations and be able to generate feasible sequences that are consistent with precedence constraints.

Intuitively, assembly precedence can be determined by analyzing accessibility ([34, 156, 35, 50, 157]). Notions of semi-infinite accessibility and infinitesimal-motion are well understood. Unfortunately, requirement of semi-infinite accessibility leads to elimination of feasible operations. On the other hand, feasibility of infinitesimal-motion alone does not guarantee that the assembly operation will be feasible. So these notions are not very useful in assembly sequence determination. However, representing accessibility in finite space is computationally challenging. So we instead would like to rely on motion planning to ensure that a proposed assembly operation is feasible.

The goal of assembly sequence planning is to generate a sequence of operations to construct a product from its individual parts. The complexity of this problem is proportional to the number of parts in the assembly ([50, 43]). However, a product

may be composed of a hierarchical structure, in which parts that mutually affect each other's accessibility belong to a common cluster that form a part set; in turn multiple part sets may mutually affect each other's accessibility constraints in the next level of assembly. Parts in such sets can be separately assembled before the final assembly of the product. Hence, the main goal of sequencing the individual parts can be divided into several subgoals that will reduce the number of assembly operations involved in each subgoal. However, the information about the interaction between part sets is not known beforehand. Therefore, we introduce a methodology to automatically detect the part interaction clusters in a product so that the assembly sequencing problem can be applied to part sets at multiple levels of hierarchy.

We start by grouping individual parts into spatial clusters based on the proximity between parts. Each cluster identified in this way is treated as an individual part set. Now, we consider the whole assembly and use motion planning to determine which part sets can be removed from the assembly. These sets are added to the first part set removal layer and removed from the assembly. We again determine which part sets can be removed from the simplified assembly. These are then added to the second layer and removed from the assembly. If the process gets stuck, parts in the parent set are regrouped, and the process is continued until all parts have been removed from the assembly. This information is used to impose precedence constraints among the part sets. Finally, we generate feasible assembly sequences using the precedence constraints for each part set and the whole assembly.

Recent advances in rapidly-exploring random trees (RRT) based motion planning ([7]) enable efficient generation of motion plans in highly crowded scenes. How-

ever, parts in their final positions in the assembly have very limited amount of feasible motions. Hence, we need to ensure that moves being tried by the RRT-based motion planner are able to correctly assess motion feasibility. This requires us to create multiple trees to ensure that if a feasible path exists, then we can find it with a very high probability.

Implementing a system based on the above ideas requires tuning a large number of parameters to ensure reasonable system performance on moderately complex assemblies. This chapter describes our approach for combining RRT-based motion planning and part interaction cluster detection for generating improved precedence constraints, which in turn can be used to generate feasible assembly sequences.

The input to the system is a 3D assembly comprising a set of parts that are described by their geometric models and relative positions. Our approach takes into account product characteristics such as component geometry, inter relationships between components, component material, and tolerances. We assume that the product is made of rigid parts and restrict ourselves to finding a feasible sequence of collision-free motions for each part (part set). We define each part (part set) as a free-flying object. Therefore, we do not consider grasping of the objects, the forces involved, or the stability of the part (part set) during motion. However, the approach can be easily extended to handle constraints imposed by tools and human hand during the assembly operation. Additionally, we consider non-linearity in the assembly operations: an assembly operation may involve more than two part sets at the same step. We assume only monotone assembly sequences: when an operation has assembled a part into a part set, that part may no longer be moved relative

to that part set. Although restrictive in application, these assumptions are very common in assembly sequencing and can be applied to a majority of products.

The approach of assembly-by-disassembly relies on constructing a disassembly sequence and then reversing the entire sequence to obtain a feasible assembly sequence. In general, the steps involved in the sequences are not necessarily symmetric, for instance, when considering flexible parts which may undergo deformation during assembly. However, under our assumptions, these two operations are indeed symmetrical.

As mentioned earlier, the main challenges encountered in assembly sequence planning include sequence generation, combinatorial explosion, non-linearity, and interaction between part sets. The main contributions of this chapter that address these problems include the following:

1. A spatial clustering based method to automatically detect the part interaction clusters in a product.

2. A technique that uses a variation of RRT-based motion planning in order to assess motion feasibility.

3. An algorithm to generate improved assembly precedence constraints by combining the part interaction cluster detection and the motion planning methods into an assembly-by-disassembly approach.

3.2 Assembly Precedence Constraints Framework

On the one hand, assembly sequence planning is a large-scale combinatorial problem; however, on the other hand, the assembly precedence relationships between parts make it a highly constrained problem. The number of potential assembly sequences is given by $n!$, where n is the number of parts in the assembly. This leads to a combinatorial explosion in the number of sequences. In addition, the linearity assumption of placing one part at a time may not be valid in complex assemblies where some parts cannot be singly added/removed (i.e., in isolation with respect to others) and the numbers of parts to be added/removed simultaneously is not always the same. These kind of non-linearities in assembly sequences increases the size of the solution space to $\frac{(2n-2)!}{(n-1)!}$. However, absolute constraints such as geometrical, precedence, and accessibility severely reduce the number of potential assembly sequences. Nevertheless, determining these constraints for an assembly problem dramatically increases the problem complexity. Assembly sequence planning was shown to be NP-complete ([40, 42]). As a result, most of the past and present work in this area focus on restricted variants of the problem.

In our framework, we combine motion planning and part interaction clusters in order to derive precedence relationships that can be used to generate assembly sequences for complex assemblies. We consider a mechanical product as a hierarchical structure of part sets. Usually, parts in each set can be separately assembled before the final assembly of the product. Hence, the original goal of sequencing the individual parts can be divided into several smaller subgoals that will reduce the

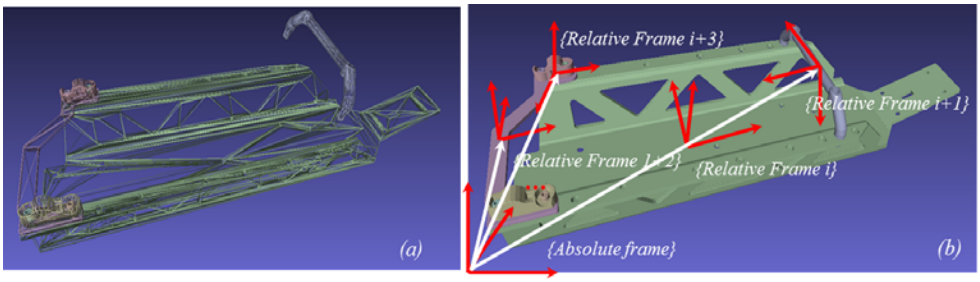


Figure 3.1: *(a) Polygonal triangulation applied to a simplified chassis assembly used in the experiments. (b) Origins of absolute and relative reference frames extracted from the 3D assembly model.*

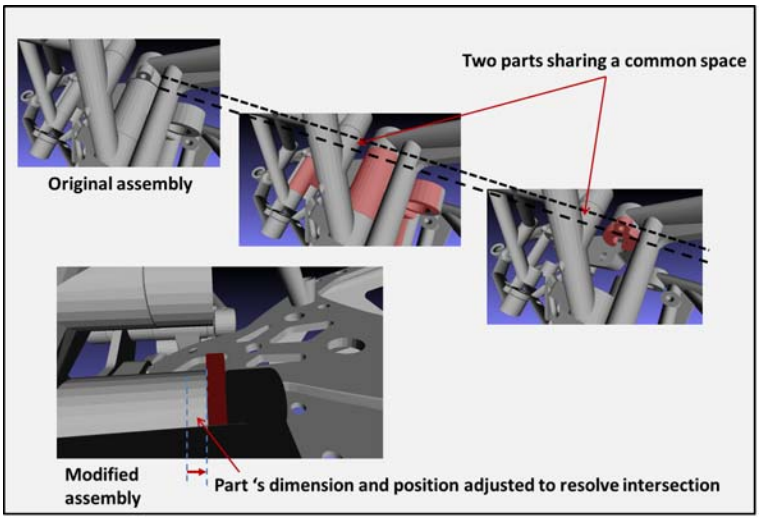


Figure 3.2: *Example of an error in the input CAD model of a complex chassis assembly caused due to intersection between two of its parts*

number of assembly operations involved in each subgoal. This, thereby, reduces the combinatorial explosion of the number of sequences that span the solution space. Since the structure of interaction between the part sets is not known beforehand, we introduce a methodology to automatically extract this information so that the assembly sequencing problem can be applied to part sets at multiple levels of hierarchy.

3.2.1 Problem Formulation

The input to the system is a 3D assembly model of a mechanical product, which is a geometrical representation of a set of individual parts (that constitute the product) in their assembled configuration. The output of the system is a set of precedence constraints that can be used to generate a feasible assembly sequence in which components are assembled to give rise to the desired product.

The CAD models used in this work were obtained from the META team at Vanderbilt University. These models, originally created in SolidWorks, were converted into a stereolithography (*.STL*) format and used as inputs to the system. Given the assembly model in .STL format, the system automatically extracts the total number of individual parts n and all the implicit geometrical information based on polygonal triangulation. An example of the polygonal triangulation generation of a simplified chassis assembly is shown in Fig. 3.1(a). The assembly model must satisfy some consistency requirements in order to have a feasible 3D workspace. For this purpose, all the assembly parts must respect their shape and volume as a rigid body in every pose; no intersections of part models in the assembled configuration are allowed. An example of error in the CAD model of a complex chassis assembly, which is caused due to intersection between two of its parts is shown in Fig. 3.2. These issues were resolved by manually adjusting the dimensions and positions of the part causing the intersection before using the assembled model as input to the system. For example, in Fig. 3.2, the CAD model of the spacer part (shown in red) was modified by changing its length and position in the assembly by trial-and-error

until the part intersection was resolved. In addition, the position and orientation of each part in relative and absolute reference frames are automatically queried (refer to Fig. 3.1(b)) and used to compute the transformation between the two frames. All the input data analysis, except resolving part intersection errors, was carried out automatically.

Next, we formalize the problem by laying out the assumptions underlying the framework, some definitions, and the problem statement as follows:

Assumption 1. *All parts are rigid, all connections between parts are rigid, and once a connection or liaison is made, it remains in this way.*

Assumption 2. *Screws and nuts are included as members of the assembly; therefore, they belong to the assembly as parts. Any fastening method other than screw/nut must be removed from the 3D assembly model before our approach can be applied.*

Assumption 3. *The geometric disassembly and assembly precedence constraints are only based on the information present in the assembly model.*

Assumption 4. *The disassembly sequence is completely reversible to turn it into an assembly sequence.*

We make the assumption 2 as we address only assemblies that use screws and nuts for fastening. Therefore, some modifications have to be made to our framework presented in the chapter before it can be applied to assemblies that use other fastening methods. For example, if glue is used for fastening two parts, then this must be specified the input data. Then, this additional information can be used in the precedence constraints analysis accordingly.

Definition 1. Posture*: The posture of a part ω is defined as $q^{\omega} = (p^{\omega}, \theta^{\omega})$, where $p^{\omega} \in \Re^3$ is the position $(x^{\omega}, y^{\omega}, z^{\omega})$ in the Euclidian space and $\theta^{\omega} \in \Re^3$ is the orientation $(\alpha^{\omega}, \beta^{\omega}, \gamma^{\omega})$ in the Euclidian space.*

Definition 2. Assembly*: We define the assembly as a set of parts $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n \mid \forall\ \omega_i : q^{\omega_i} \in \Re^6, c^{\omega_i} \in \Re^3\}$, where q^{ω_i} and c^{ω_i} represent the posture and the center of mass of the part ω_i, respectively. The set Ω includes all the parts in the input assembly.*

Definition 3. Obstacles*: We define a set of obstacles $O = \{o_1, o_2, \ldots, o_n \mid \forall\ o_i : q^{o_i} \in \Re^6\}$, where q^{o_i} represents the posture of an obstacle o_i. When we select a part to evaluate its motion feasibility, all the remaining parts of the assembly are considered as potential obstacles. Therefore, initially, the set O includes all the parts in the input assembly.*

Definition 4. Part set*: A part set is defined as a set of parts and/or part subsets. A part set is considered to be a parent C^p for a set of child sets $\{C_i^c : i = 1, \ldots, k\}$; in turn, each child set is a parent for subsequent child sets, recursively, until each child is a single part. Accordingly, it is clear that the root part set C^r refers to the whole assembly Ω.*

Definition 5. Disassembly structure*: The disassembly structure for a given n-part assembly Ω is defined as a set of hierarchical layers $\mathbf{H} = \{h_1, h_2, \ldots, h_{|\mathbf{H}|}\}$, where h_i represents the i^{th} disassembly layer and $|\mathbf{H}|$ is the number of layers. Every layer is composed of its part sets $h_i = \{C_1, C_2, \ldots, C_{|h_i|}\}$, where every C_i is defined according to Definition 4 and $2 \leq \sum_{i=1}^{|\mathbf{H}|} |h_i| = n$.*

Definition 6. Disassembly part set precedence $\prec$: *Let* $C_i^c, C_j^c \in C^p$. *If* C_j^c *can be removed only after removing* C_i^c, *then we say* $C_i^c \prec C_j^c$.

Definition 7. Disassembly part set equivalence $\sim$: *Let* $C_i^c, C_j^c \in C^p$. *If* C_i^c *can be removed before* C_j^c *and* C_j^c *can be removed before* C_i^c, *then we say* $C_i^c \sim C_j^c$.

Definition 8. Disassembly layer precedence $\prec\prec$: *Let* $h_i, h_j \in \mathbf{H}$ *such that* $i < j$ *(i.e., layer* h_i *is generated before layer* h_j*), then we say* $h_i \prec\prec h_j$. *Moreover, if* $C_i \in h_i$ *and* $C_j \in h_j$ *then* $C_i \prec\prec C_j$.

Definition 9. Disassembly layer equivalence $\approx$: *Let* $h \in \mathbf{H}$. *If* $C_i, C_j \in h$, *then we say* $C_i \approx C_j$.

Definition 10. Assembly precedence $\rightsquigarrow$: *Let* $C_i^c, C_j^c \in C^p$. *If* C_j^c *can be assembled only after assembling* C_i^c, *then we say* $C_i^c \rightsquigarrow C_j^c$.

Definition 11. Assembly equivalence $\leftrightsquigarrow$: *Let* $C_i^c, C_j^c \in C^p$. *If* C_j^c *can be assembled either before or after assembling* C_i^c, *then we say* $C_i^c \leftrightsquigarrow C_j^c$.

Problem Statement: *Given a 3D assembly model of a n-part mechanical assembly* Ω *(Definition 2), find a disassembly structure* $\mathbf{H} = \{h_1, h_2, \ldots, h_{|\mathbf{H}|}\}$ *(Definition 5) along with a set precedence relationships* $\mathbf{P}$ *that can be used to generate feasible assembly sequences.*

Next, we describe our overall approach to find a solution to the above problem.

3.2.2 Overview of Approach

Initially, with the assumption that the largest part of the assembly is the driving part that guides the assembly process, we extract this part from the given CAD model and keep it aside. If there are multiple largest parts, then heuristic information (e.g., which of these parts has maximum surface contact with the ground) is used to pick one of them. When there is lack of such information, one of them is picked randomly.

Next, we group the remaining $n-1$ parts into k $(=2)$ part sets using spatial $k-means$ clustering. Under this new arrangement, the assembly is composed of $k+1$ part sets−the largest part, part set 1, part set 2, ..., part set k −in the first step. Now, we verify the assembleability of this new assembly. We are defining assembleability as the feasibility to assemble a part into a partially completed assembly. For this purpose, we use motion planning to determine which part sets can be removed from the assembly (explained in Section 3.3). These part sets are added to the first disassembly layer and removed from the assembly. We again determine which part sets can be removed from the simplified assembly. These part sets are then added to the second layer and removed from the assembly. If the process gets stuck before all part sets have been removed, we go to the first step, in which parts are rearranged into a different grouping by increasing the number of clusters by one. This results in $k+1$ new clusters. The cycle is continued until all part sets have been removed from the assembly and all part set removal layers are found. The techniques and algorithms used to extract part interaction clusters are described in

detail, using an illustrative example, in Section 3.4.

Note that until now, the disassembly layers have been generated for the whole assembly while considering the identified clusters as individual part sets. Therefore, the above procedure is applied recursively to generate disassembly layers for each cluster identified in the previous step. The information extracted during the above procedure is used to impose precedence constraints among the part sets. Finally, we generate feasible assembly sequences using the precedence constraints for each part set and the whole assembly. Some theoretical results related to the properties of the proposed algorithms are presented in Section 3.5. Computational results from tests on a variety of assemblies are reported in Section 3.6.

Based on the complexity of the assembly, we cannot reject some part sets that do not pass the assembleability test. This occurs when at least two parts must be moved into their final assembly locations simultaneously. Therefore, whenever there is a failure, we recheck the assembleability for the largest assembly part and a single set obtained by merging the part sets. If the assembly structure passes the test then a temporary assembly location is needed to generate a feasible assembly sequence. This process give us the information about the existence of nonlinearity in the assembly. An example of this assembly scenario is explained in Section 3.6.5.

Next, we describe the principal techniques used to implement the above approach − motion planning, generation of disassembly layers, and spatial partitioning based part interaction cluster extraction.

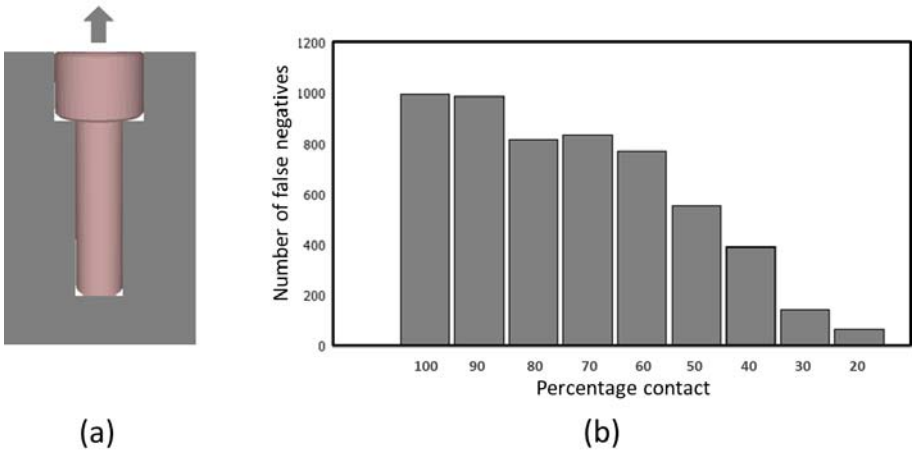


Figure 3.3: *(a) A screw which is in 100 % contact with an external surface. (b) Performance of rapidly exploring random trees as a function of percent contact.*

3.3 Motion planning to assess feasibility of part set Disassembly

The capability of sampling-based motion planners to perform assembly analysis of complex product models can be mainly attributed to their computational efficiency. For example, the Manhattan-like RRT based planner presented in [89] can handle models with hundreds of degrees of freedom. Therefore, as mentioned earlier, we rely on motion planning to evaluate the feasibility of an assembly operation. In particular, we developed a multiple RRT based motion planning algorithm to compute a collision-free escape path to move a part set from its assembled configuration to a given location that lies outside of the assembly.

Considering that the parts are moved and disassembled one by one, an assembly admits a disassembly sequence if an escape path for disassembling each part $\omega_i \in \Omega$ can be found. Given the initial assembled state configuration $\{q_{initial}^{\omega_i} : i = 1, 2, \ldots, n\}$, the problem consists of computing a collision free escape path, from $q_{initial}^{\omega_i}$ to a disassembled configuration $q_{goal}^{\omega_i}$ for all $\omega_i \in \Omega$.

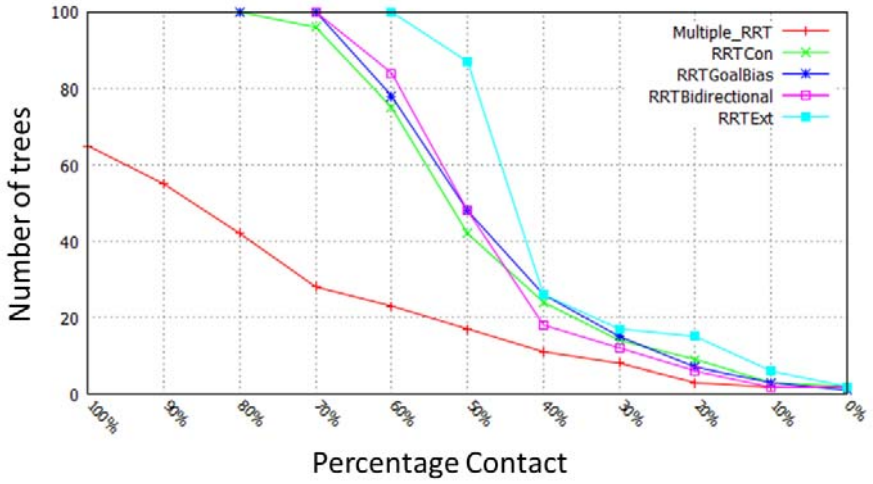


Figure 3.4: *Number of trees used to resolve problem of false negatives with percentage contact (Maximum number of trees = 100). Performance of DMRT compared with other RRT variants ([5, 6, 7])*

Rapidly-exploring random trees (RRT) based motion planning provides feasible solutions for part navigation in crowded scenes, a problem representative of searching non-convex, high-dimensional spaces. A RRT is constructed incrementally by randomly sampling valid configuration states and quickly computing the nearest neighbor to states that already belongs to the tree. This process quickly reduces the expected distance of a randomly-chosen point to the tree. However, RRT cannot be directly applied to complex assemblies that are composed of large number of parts, which often lead to scenes with high obstacle densities. The large number of obstacles generates a highly constrained environment with very narrow passages, resulting in the RRT's failure to generate a valid state. This, thereby, increases the false negatives dramatically.

In order to assess assembly feasibility, we used the RRT with multiple random

trees based motion planning approach described in our earlier work ([23]). The approach performs a robust analysis of part motion feasibility. It uses multiple RRTs that dynamically modify the environment description in order to generate a valid escape path. It dynamically modifies the number of trees for each assembly part based on the environment constraints associated with the part. A highly constrained environment may require a large number of trees to find the escape path for an assembly part. Multiple RRTs provide greater robustness across narrow passages and crowded environments.

We use a screw which is in 100 % contact with an external surface, as shown in Fig. 3.3(a), in order to illustrate the impact of percent contact on the performance of RRT. Figure 3.3(b) shows the number of failures out of 1000 attempts made by the algorithm in order to fully remove the screw out of its initial location as a function of percent contact of the part. The graph shows that the traditional RRT finds it increasingly difficult to find a successful path to remove the screw with an increase in the percent contact. The same example shown in Fig. 3.3(a) is used to compare the performance of the multiple RRT algorithm used for assessing motion feasibility in this chapter with that of other RRT variants as shown in Fig. 3.4. Note that our approach uses the least number of trees to find a solution for all cases of percent contact.

[158] proposed a multiple RRT algorithm, in which the number of trees is kept fixed. However, in our approach, the number of trees is changed on-the-fly; that is, a new tree is added only if it is required by merging the information about the "old" tree with the "new" tree instantaneously, thereby, increasing the probability

of finding a valid state in the next step by using the knowledge of the previous trees.

3.4 Generation of Part Interaction Clusters

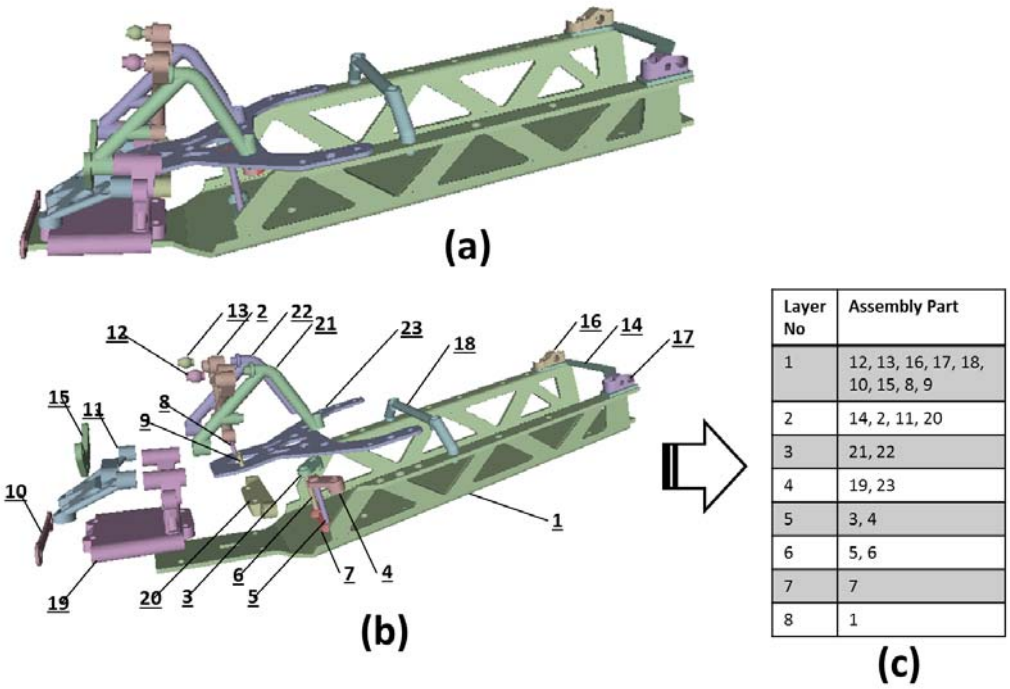
3.4.1 Basic Idea Behind Identification of Disassembly Layers

The motion planning technique uses the geometric information extracted from the assembly model to find an escape path that allows a part to be completely separated from the assembly. This enables the system to determine which part movements are forbidden, which movements are feasible, and thereby, determine which part(s) can be removed at each stage of disassembly. Using Definition 5, we consider a hierarchical disassembly structure $\mathbf{H} = \{h_1, h_2, \ldots, h_{|\mathbf{H}|}\}$, such that $h_i \prec\prec h_{i+1}, \forall\ i \neq |\mathbf{H}|$. Therefore, each layer h_i represents a precedence constraint for the layer h_{i+1} for disassembly.

For simplicity, we first describe the generation of disassembly layers without grouping part into clusters and by applying motion planning only to individual parts. Specifically, the system considers each part and uses the motion planning algorithm to check if it is physically blocked by another part before it can be fully removed out of the assembly. Parts that can be removed in this manner during the first attempt fill the first layer h_1. This process is repeated to fill the second layer h_2 and so on, until all the parts are disassembled. As a result, the process is bounded by a maximum of n^2 iterations. The resulting hierarchically layered structure comprising part groups that can be removed at each layer gives rise to disassembly precedence

Algorithm 1 Assembly sequencing using DMRT based motion planning.

Input: Ω, O
Output:
$Assembleability \in \{\text{TRUE, FALSE}\}$,
$AssemblySequence$
Procedure $DMRT(\mho) \rightarrow (DisassemblyLayer, \mho)$
$O \leftarrow \mho$;
while $(|\ \mho\ | \neq 0)$ **do**
$\tau.init(q^{\omega}_{init})$;
$O \leftarrow RemovePart(O, \omega)$;
while $(EscapePathFound == FALSE)$ **do**
$q^{\omega}_{rand} \leftarrow RandomStage()$;
for *each* $TREE$ **do**
$q^{\omega}_{near} \leftarrow NearestNeighbor(q^{\omega}_{rand}, \tau)$;
$u \leftarrow SelectInput(q^{\omega}_{rand}, q^{\omega}_{near})$;
$CheckCollision()$;
if $(Collision == FALSE)$ **then**
$q^{\omega}_{new} \leftarrow Newstate(q_{near}, u, \Delta t)$;
$\tau.AddVertex(q^{\omega}_{new})$;
$\tau.AddEdge(q^{\omega}_{near}, q^{\omega}_{new}, u)$;
else
ReinitializeNewTree;
end if
end for
end while
if $(EscapePathFound == TRUE)$ **then**
$DisassemblyLayer \leftarrow AddPart(DisassemblyLayer, \omega)$;
$O \leftarrow AddPart(O, \omega)$;
$\mho \leftarrow RemovePart(\mho, \omega)$;
$|Paths| + +$;
end if
end while
$\mho \leftarrow \Omega$; $i = 1$; $|Paths| = 0$;
while $(|\ \mho\ | \neq 0)$ **do**
$(h_i, \mho) \leftarrow DMRT(\mho)$;
$\mathbf{H} \leftarrow AddLayer(\mathbf{H}, h_i)$;
end while
if $(|\ \Omega\ | \ == \ |\ Paths\ |)$ **then**
$AssemblySequence \leftarrow ReverseSequence(\mathbf{H})$;
Return $(TRUE, AssemblySequence)$
else
Return $Assembleability = FALSE$
end if

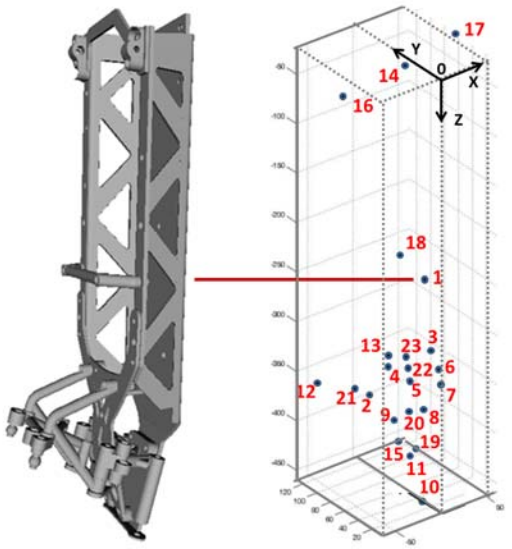


Layer No	Assembly Part
1	12, 13, 16, 17, 18, 10, 15, 8, 9
2	14, 2, 11, 20
3	21, 22
4	19, 23
5	3, 4
6	5, 6
7	7
8	1

Figure 3.5: *(a) 23-part assembly. (b) Exploded view. (c) Generation of disassembly layers for 23-part assembly.*

relations. Generation of such disassembly layers for a chassis assembly of 23 parts is shown in Fig. 3.5.

The disassembly layers generated by the above process can be reversed and turned into a linear assembly sequence as shown in Algorithm 1. However, the above process does not fully represent the precedence relations between parts across different layers. For instance, H provides precedence relation between layers $h_i \prec\prec h_j$ for disassembly. Therefore, if $\omega_i \in h_i$ and $\omega_j \in h_j$, then $\omega_i \prec\prec \omega_j$. That is ω_i can be removed before ω_j. However, this doesn't necessarily imply whether or not ω_j can be removed before ω_i. Consequently, we do not know for sure if $\omega_i \rightsquigarrow \omega_j$ for assembly. Also, the precedence relations between the parts in the same layer are



	X	Y	Z
1	0	26.85	-232.28
2	0.22	107.425	-360.295
3	23.255	50.24	-322.675
4	-23.25	50.235	-322.68
5	-16	29	-329.345
6	16	29	-329.345
7	0	5.68	-329.345
8	16	51.15	-379.54
9	-16	51	-379.54
10	0	32.155	-458.805
11	-0.01	48.615	-419.765
12	-38.78	130.59	-370.62
13	39.22	130.34	-370.69
14	-0.025	55.5	-29.75
15	0.005	65.73	-413.345
16	-62.455	64.225	-43.45
17	61.955	64.225	-22.55
18	0.01	62.89	-223.5
19	0	41.57	-410.51
20	0	49.955	-375.705
21	-30	90.34	-361.085
22	30	90.14	-361.115
23	0	55.35	-323.785

Figure 3.6: *Definition of the centers of mass* $L = \{c_1^{\omega}, c_2^{\omega}, \ldots, c_n^{\omega}\}$ *based on the 3D assembly model.* $L = \{$*(0, 26.85, -232.28), (0.22, 107.425, -360.295), . . . , (0, 55.35, -323.785)*$\}$.

not evaluated by this method.

In the next subsection, we show how the assembly structure identified by our new approach reveals a more detailed precedence relationships by reorganizing the parts in different layers into different part sets.

3.4.2 Spatial Partitioning based Part Interaction Cluster Formation

We developed a spatial partitioning algorithm based on $k-means$ clustering technique in order to generate part interaction clusters. K-means clustering (MacQueen, 1967) is a method commonly used to automatically partition a data set into k groups. It proceeds by selecting k initial cluster centers and then iteratively refining them as follows: (a) Each instance c^{ω_i} is assigned to its closest cluster. (b) Each cluster center C_j is updated to be the mean of its constituent instances. The algorithm

converges when there is no further change in assignment of instances to clusters or alternatively, when there is no change in the location of the cluster centers. The pseudocode for spatial clustering is given in Algorithm 2.

Algorithm 2 K-Means based spatial partitioning.

Input:
$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$
$L^{\Omega} = \{c^{\omega_1}, c^{\omega_2}, \ldots, c^{\omega_n}\}$ (Centers of mass for parts to be clustered)
K (number of clusters)

Output:
$C = \{p_1, p_2, \ldots, p_K\}$ (Set of partitions) $L^C = \{c^{p_1}, c^{p_2}, \ldots, c^{p_K}\}$; (cluster centroids)
$m : \Omega \rightarrow C$; (cluster membership)

Procedure SpacePartitioning$(\Omega, L^{\Omega}, K) \rightarrow (C, L^C, m)$
  $L^C \leftarrow InitialValue($ Random selection of K values from $L^{\Omega})$;
  for $\omega_j \in \Omega$ **do**
    $m(\omega_j) = \arg\min_{k \in \{1,2,\ldots,K\}} Distance(c^{\omega_j}, c^{p_k})$;
  end for
  while (there is change in m) **do**
    for $j \in \{1, \ldots, k\}$ **do**
      Recompute c^{p_j} as the centroid of $\{\omega \mid m(\omega) = p_j\}$;
    end for
    for $\omega_j \in \Omega$ **do**
      $m(\omega_j) = \arg\min_{k \in \{1,2,\ldots,K\}} DISTANCE(c^{\omega_j}, c^{p_k})$;
    end for
  end while
  Return C, m

We illustrate the part interaction cluster extraction technique using the example shown in Fig. 3.6. We initialize the number of clusters $K = 2$ and initialize the cluster centers using instances chosen at random from the assembly set Ω. The data set is composed of 3D positions that represent the center of mass c_i^{ω} of each part ω_i. We use the Euclidean distance metric to compute closeness of a data point to cluster centers. The cluster centers of the product affect the assembly feasibility in the sense that the resulting parts in one cluster may or may not be physically separated from parts in other clusters. Therefore, whenever a part partitioning corresponding to a set of cluster centers is not assembleable, then K is incremented by 1 and new

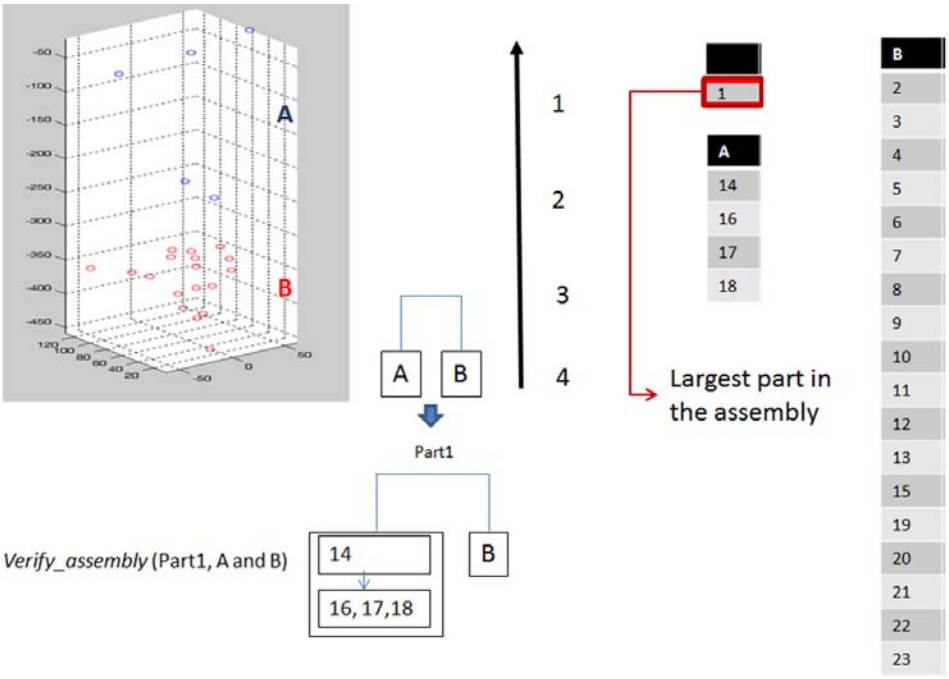


Figure 3.7: *First cycle of partitioning for the chassis assembly. Two valid part sets A and B are obtained with k = 2.*

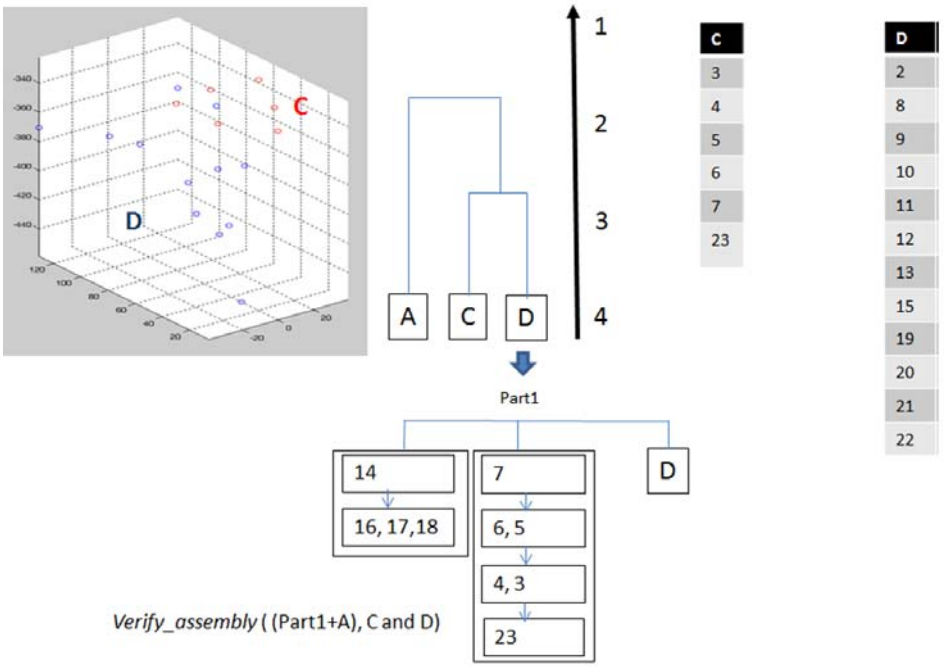


Figure 3.8: *Second cycle of partitioning for the chassis assembly. B is partitioned into two valid part sets C and D.*

cluster centers are identified. This process is repeated until cluster centers that lead to a successful partitioning are found.

Every time an assembly set is repartitioned into a set of K clusters, we must test whether the new organization of the parts is feasible to assemble or not. For this purpose, we treat these clusters as individual part sets by merging all elements that belong to one cluster into one part set. Therefore, the reorganized assembly structure is composed of the set $\Omega_{reorg} = \{\omega_{rest}, C_1, C_2, \ldots, C_K\}$, where ω_{rest} is either the largest part in the first iteration or the merged part set composed of all the assembly elements excluded from the current partitioning analysis. The pseudocode used to implement assembleability testing is shown in Algorithm 3. We introduce a parameter c_{min}, which represents the minimum number of elements contained in a valid cluster. That is, whenever $|C_i| \leq c_{min}$, C_i is not partitioned further. In all our experiments $c_{min} = 6$[1].

The first cycle of partitioning ($K = 2$) is shown in Fig. 3.7. In this case, the reorganized assembly Ω_{reorg} is composed of three part sets: Part set A, part set B, and ω_{rest}, which is equal to the largest part of the assembly $\omega_{largest}$ detected at the beginning of the algorithm. Symbolically, $\Omega_{reorg} = \{\omega_{largest}, A, B\}$. The motion planning module finds that the resulting assembly structure is assembleable. Therefore, the algorithm proceeds to the second cycle of partitioning. Note that $|A| = 4$ ($< c_{min}$). Therefore, A doesn't undergo further partitioning. However, note that $|B| = 18$ ($> c_{min}$). Therefore, B is further partitioned into C and D

[1] A low value of cmin (= 6) for the cluster results in obtaining manageable part clusters at the leaf node of the assembly tree structure whenever possible. Similar results will be obtained if this value is changed to 5 or 7.

as shown in Fig. 3.8. The new assembly is composed of three part sets C, D, and ω_{rest}, where ω_{rest} is a merged part set composed of $\omega_{largest}$ and A. Therefore, $\Omega_{reorg} = \{\omega_{largest}, A, (B \rightarrow C + D)\}$ The updated assembly structure is verified to be assembleable. In the third cycle, C doesn't undergo partitioning ($|C| = 6$) and D is partitioned into E and F ($|D| = 12$) (Fig. 3.9). Similarly, F is partitioned into G and H (Fig. 3.10). However, the motion planning module finds that the resulting partitioning is not feasible for assembly. We continue repartitioning of a invalid cluster only when at least one of its subclusters contains c_{min} elements. Since $|G| < 6$ and $|H| < 6$, the algorithm halts resulting in F as a single part set. Therefore, We obtain the final assembly structure $\Omega_{reorg} = \{\omega_{largest}, A, (B \rightarrow (C+(D \rightarrow E+F)))\}$. The pseudocode to implement the spatial partitioning and motion planning based automated generation of assembly precedence constraints is given in Algorithm 4.

Algorithm 3 Assembleability testing.

Input:
$\omega_{rest}^{assembly}$ (Part excluded from partitioning analysis. Initially it is the largest part).
K (number of clusters)
Output:
$Assembleability \in$ {TRUE, FALSE},
$AssemblySequence$
Procedure VerifyAssembly: $(\Omega, \omega_{rest}^{assembly}, K) \rightarrow (Assembleability, AssemblySequence)$
  $Assembleability = FALSE$;
  while $(k \leq \Omega \vee AssemblySequence == FALSE)$ **do**
    $(C, L^C, m) \leftarrow SpacePartitioning(\Omega, L^{\Omega}, K)$;
    for $k \in \{1, 2, \ldots K\}$ **do**
      $(\hat{\mathbf{\Omega}}) \leftarrow MergeElements(C, m, \Omega)$;
    end for
    $(Assembleability, AssemblySequence) \leftarrow DMRT(\hat{\mathbf{\Omega}})$;
  end while
  Return $(Assembleability, AssemblySequence)$;

As mentioned earlier, a linear assembly sequence is one in which each operation places a single part into the assembly. Although not all products can be assembled

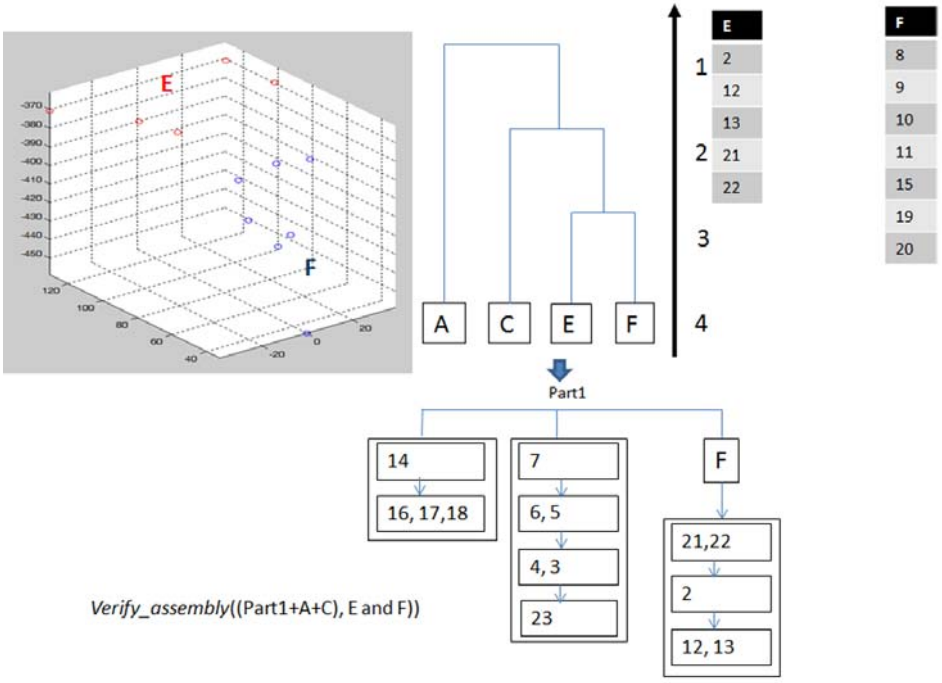


Figure 3.9: *Third cycle of partitioning for the chassis assembly. D is partitioned into two valid part sets E and F.*

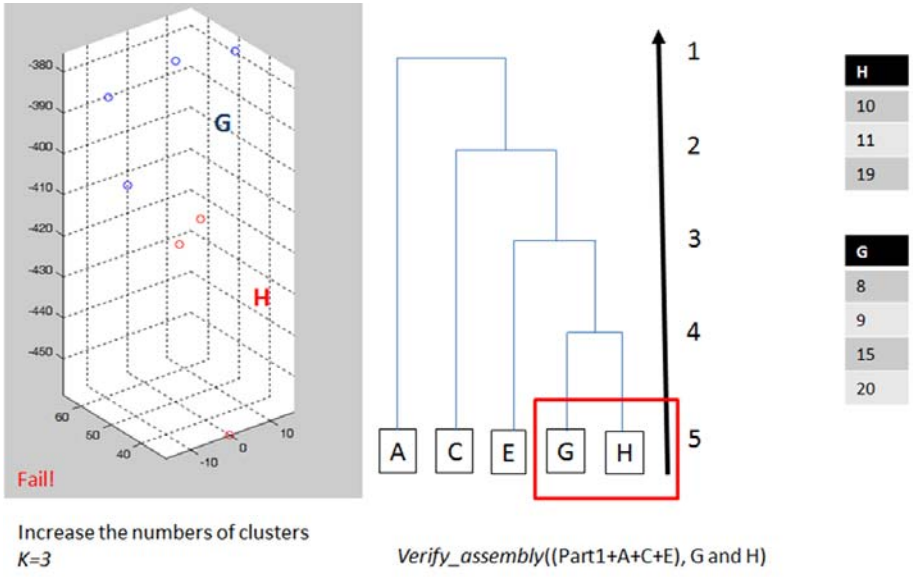


Figure 3.10: *Fourth cycle of partitioning for the chassis assembly. F is partitioned into two part sets G and H. However, this partitioning fails the assembleability test making the part sets invalid.*

Algorithm 4 Spatial partitioning and motion planning based automated generation of assembly precedence constraints.

Input: Ω
Output: $Assembleability \in \{\text{TRUE, FALSE}\}$; $PartSet$; $TotalPartSetSequence$

$\omega_{largest} \leftarrow IdentifyLargestPart$;
$\Omega \leftarrow RemovePart(\omega_{largest})$;
$\omega_{rest}^{assembly} = \omega_{largest}$;
$L^{\Omega} \leftarrow PartCenterOfMass(\Omega)$;
$c_{min} = 6$;
$Cluster = \{C_1, C_2, ...C_u \forall u \in \aleph^+\}$(List of all clusters in the assembly);
$ClusterSet = \{C_1 : C_1 = \Omega\}$
$NumberOfNewClusters = 1$;
$i = 0$;
while $(NumberOfNewClusters > 0)$ **do**
  $NumberOfPreviousClusters = NumberOfNewClusters$;
  $i++; k = 1$;
  $\omega_{rest}^{assembly} = \omega_{largest} + ClusterSetElements(C_1,]dots, C_{i-1})$
  repeat
    $k++$;
    $Assembleability \leftarrow VerifyAssembly(C_i, \omega_{rest}^{assembly}, k)$;
  until $(Assembleability == FALSE)$
  if $(Assembleability == TRUE)$ **then**
    for $C_j \in \{1, \ldots, K\}$ **do**
      if $|C_j| \leq c_{min}$ **then**
        $\hat{\mathbf{\Omega}} \leftarrow C_j$;
        $PartSetSequence_j \leftarrow DMRT(\hat{\mathbf{\Omega}})$;
      else
        $ClusterSet \leftarrow AddCluster(C_j)$;
      end if
    end for
  else
    $(\hat{\mathbf{\Omega}}) \leftarrow MergeElementsInCluster(C, m, \Omega)$;
    $(Assembleability, AssemblySequence) \leftarrow VerifyAssembly(C, \omega_{rest}^{assembly}, k)$;
    if $(Assembleability == TRUE)$ **then**
      $AssembleSametime(C_i \in C)$;
      $ClusterSet \leftarrow AddCluster(C)$
    end if
  end if
  $NumberOfNewClusters = |ClusterSet| - NumberOfPreviousClusters$;
end while
if $(\sum_{i=1}^{|ClusterSet|}(C_i) = |\Omega|)$ **then**
  Return $TotalClusterSequence \leftarrow TotalAssembly(ClusterSequence_j)$;
else
  Return $Assembleability = FALSE$;
end if

linearly, such sequences are used in manufacturing owing to their simplicity. Here, we showed that our approach can be used to achieve more complex organizational levels in which assembly of part sets can be assigned to different stations and later, these part sets can be put together to form the final assembly.

3.5 Properties Of Proposed Algorithms

In this section, we present some theoretical results related to the techniques and the associated algorithms developed in this chapter. The notations and definitions used in the lemmas and theorems were described in Section 3.2.1.

Lemma 1. Transitive precedence*: If* $C_i \prec C_j$ *and* $C_j \prec C_k$*, then* $C_i \prec C_k$.

Proof.

Let $C_i, C_j, C_k \in \mathbf{H}$.

From Definition 6, we have $C_j \prec C_k$

$$\Rightarrow C_k \text{ can be removed only after removing } C_j. \tag{3.1}$$

However, $C_i \prec C_j$

$$\Rightarrow C_j \text{ can be removed only after removing } C_i. \tag{3.2}$$

Therefore, from (3.1) and (3.2), we have $C_i \prec C_k$.

□

For example, in the 23-part assembly (Fig. 3.5), Part-23 $\prec$ Part-4 and Part-4 $\prec$ Part-5. Therefore, according to Lemma 1, Part-23 $\prec$ Part-5. Also, from visual

inspection of the exploded view of the part assembly (Fig. 3.5(b)), it can be verified that Part-23 must be removed before removing Part-5, which is consistent with the above lemma.

Lemma 2. *Let $C_i, C_j \in H$. Now, if $C_i \approx C_j$ then $C_i \sim C_j$.*

Proof. $C_i \approx C_j \Rightarrow C_i, C_j \in h$ for some $h \in H$. That is, the part sets belong to the same layer. Therefore, from Definition 5 and Definition 6, we have $C_i \sim C_j$. □

Remark 1. *Note that the converse may or may not be true.*

For example, in the 23-part assembly (Fig. 3.5), Part-16 ≈ Part-17 (the two parts belong to the same layer). Therefore, Part-16 ∼ Part-17 (Part-16 can be removed before Part-17 and vice-versa). Note that for Part-6 ∼ Part-4 the converse, Part-6 ≈ Part-4, is not true.

Lemma 3. *Let $C_i \in h_i$ and $C_j \in h_j$ for some $h_i, h_j \in H$, where $i \neq j$. Now, if $C_i \prec C_j$ then $C_i \prec\prec C_j$.*

Proof. Suppose $\neg(C_i \prec\prec C_j)$. This implies $C_j \prec\prec C_i$ or $C_i \approx C_j$.

If $C_j \prec\prec C_i$, from Definition 8 $h_j \prec\prec h_i$. This implies $\neg(C_i \prec C_j)$, which is a contradiction.

If $C_i \approx C_j$, then from lemma 2, we have $C_i \sim C_j$, which is also a contradiction. Therefore, we have the result. □

Remark 2. *Note that the converse may or may not be true. In particular, the converse is true if, additionally, there is no C_k, $k \neq i$ such that $C_k \prec C_j$.*

For example (Fig. 3.5), Part-4 $\prec$ Part-5. From Lemma 3, Part-4 $\prec\prec$ Part-5. Note that the converse of Part-4 $\prec$ Part-5 is true as Part-3 $\prec$ Part-5 is not true. Now, Part-4 $\prec\prec$ Part-6. However, the converse of Part-4 $\prec\prec$ Part-6, Part-4 $\prec$ Part-6, is not true.

Theorem 1. *For any two adjacent layers h_i and h_{i+1}, a layer precedence relationship $h_i \prec\prec h_{i+1}$ is established if and only if, for every part set $C_{i+1} \in h_{i+1}, \exists$ at least one part set $C_i \in h_i$ such that $C_i \prec C_{i+1}$.*

Proof. First, we prove the "if" part: Let $\mho \subseteq h_i$ such that $|\mho| \geq 1$ and $C_i \prec C_{i+1}, \ \forall \ C_i \in \mho$. Using lemma 3, we have $C_i \prec\prec C_{i+1}, \ \forall \ C_i \in \mho$. This implies $h_i \prec\prec h_{i+1}$. Hence, the "if part" is proved.

Next, we prove the "only if" part:

$$\text{Given } h_i \prec\prec h_{i+1} \tag{3.3}$$

Suppose the result is untrue. That is, for some part set $C_{i+1} \in h_{i+1}$, $\neg(C_i \prec C_{i+1})$ for every $C_i \in h_i$.

$$\Rightarrow C_i \sim C_{i+1}, \ \forall C_i \in h_i \tag{3.4}$$

$$\text{or } C_{i+1} \prec C_i, \ \forall C_i \in h_i \tag{3.5}$$

(3.4) $\Rightarrow C_{i+1} \in h_i$. However, this is a contradiction, since $h_i \cap h_{i+1} = 0$.

(3.5) $\Rightarrow C_{i+1} \prec\prec C_i$ (From lemma 3). Equivalently, $h_{i+1} \prec\prec h_i$, which is also a contradiction. Hence, we have the result. □

From Fig. 3.5(c), $h_2 \prec\prec h_3$, where $h_2 = \{$Part-2, Part-11, Part-14, Part-20$\}$ and $h_3 = \{$Part-21, Part-22$\}$. Note that Part-2 $\prec$ Part-21 and Part-2 $\prec$ Part-22. That is, for every Part-i $\in h_3$, there exists at least one Part-j $\in h_2$ such that Part-j $\prec$ Part-i. This is consistent with Theorem 1. Note that Part-21 and Part-22 have no individual precedence relationships with the remaining parts in h_2 even though they fall in different layers.

From theorem 1, we can state the following corollary.

Corollary: For any two layers h_i and h_j, a layer precedence relationship $h_i \prec\prec h_j$ is established if and only if, for every part set $C_j \in h_j, \exists$ at least one part set $C_i \in h_i$ such that $C_i \prec C_j$.

From Fig. 3.5(c), $h_1 \prec\prec h_3$, where $h_1 = \{$ Part-8, Part-9, Part-10, Part-12, Part-13, Part-15, Part-16, Part-17, Part-18$\}$. Note that Part-12 $\prec$ Part-21 and Part-13 $\prec$ Part-22, which is consistent with the above corollary.

Theorem 2. *Given the n-part assembly* Ω*, the part interaction clusters extracted by Algorithm 4 admit a tree structure* T *with* Ω *as its root part set and a monotonically decreasing bound on its branching factor. Further, the branching converges in a finite number of partition levels that is bounded by* $n - 2$*.*

Proof. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$.

Applying Algorithm 4, we get $\Omega = C^r = \{C_1, C_2, \ldots, C_k, \omega_{largest}\}$, where each

C_i is a child part set of the parent part set C^r according to the Definition 4 and $\omega_{largest}$ is the largest part. Algorithm 4 is applied recursively on each child part set until each part set is a single part. By considering each part set as a node and connecting each child part set to its parent by an edge, the resulting part sets can be represented by a tree structure T, where Ω is the root node.

Let $C^p = \{C_1^c, C_2^c, \ldots, C_r^c\}$ for some node $C^p \in T$. Note that the branching factor at $C^p = r$. The corresponding bound on the branching factor is $|C^p|$. This implies $2 \leq r \leq |C^p|$. Similarly, the bound on branching factor at each $C_i^c \in C^p = |C_i^c|$. However, $|C_i^c| < |C^p|$, which implies a monotonically decreasing bound on the branching factor of the tree. Note that the maximum bound on the branching factor is $n-1$, which occurs at the first partition level.

Consider a path from the root node to the leaf node. We note that the maximum number of partition levels is obtained by choosing a $k = 2$ partition at each level i, with $|n-1-i|$ number of part sets in one partition and a single part in the other partition. This results in the following branching structure:

$$
\begin{aligned}
C^1 &= \{C^2, \omega : |C^2| = n-2, \omega \in C^1 - C^2\} \\
C^2 &= \{C^3, \omega : |C^3| = n-3, \omega \in C^2 - C^3\} \\
&\vdots \\
C^i &= \{C^{i+1}, \omega : |C^{i+1}| = n-i-1, \omega \in C^i - C^{i+1}\} \\
&\vdots \\
C^{n-2} &= \{C^{n-1}, \omega : |C^{n-1}| = 1, \omega \in C^{n-2} - C^{n-1}\}
\end{aligned} \tag{3.6}
$$

From (3.6), $|C^{n-1}| = 1$. This implies that the branching terminates since the minimum number of parts in a part set for it to be partitioned is two. Therefore, Algorithm 4 partitions Ω into part sets in a finite number of partition levels, which is bounded by $n - 2$. □

The tree structures obtained for the 23-part chassis assembly, the 73-part chassis assembly, the crank shaft assembly, and the radial engine assembly are shown in Figs. 3.13, 3.16, 3.19, and 3.22, respectively. In all these examples, the branching converges in a finite number of partition levels.

Theorem 3. *The number of possible assembly sequences AS_n of an n-part assembly based on part interaction cluster structure extracted by Algorithm 4 is $\sum_{i=1}^{n-1} SA_i \cdot SA_{n-i}$, where SA_i is the number of possible assembly sequences for the parent part set with i child part sets.*

Proof. Let Algorithm 4 partition the parts into two part sets C_1 and C_2. Let $|C_1| = u$ and $|C_2| = v$ such that $u+v = n$, for some u and $v \in \aleph^+$. Accordingly, let us assume assembling u parts and v parts separately, and then combining the u-part and v-part sets to form the n-part assembly. In this particular case, the total number of possible assembly sequences of an n-part assembly is $SA_u \cdot SA_v$. In the general case, SA_n is the sum of all such cases:

$$\begin{aligned} AS_n &= SA_1 \cdot SA_{n-1} + SA_2 \cdot SA_{n-2} + ... + SA_{n-1} \cdot SA_1 \\ &= \sum_{i=1}^{n-1} SA_i \cdot SA_{n-i} \end{aligned}$$

□

3.6 Computational Results

We report the results from application of our technique to four different assemblies: (a) Chassis assembly (b) Crankshaft assembly (c) Radial crankshaft assembly, and (d) The five part puzzle.

3.6.1 Simple Chassis Assembly

In the previous section, we used the 23-part chassis assembly (Fig. 3.6(a)) in order to illustrate our approach of combining part interaction cluster extraction and motion planning to generate assembly sequences. Figure 3.11(a) shows the part sets identified by the system and Fig. 3.11(b) shows the corresponding assembly structure representing the precedence relations between different part sets and individual parts within each part set. Note that the algorithm identifies four part sets, in which F precedes E ($F \prec E$) and the rest of the part sets do not have any additional precedence constraints. This means that A, C, and F can be fit into the assembly (part 1) in any order, but E must be assembled after F. Also, note that the individual parts in each set can be assembled in parallel at different locations without any precedence between parts across different part sets. However, note that individual parts admit precedence relations with each other within each part set. The disassembly layers generated by the approach used in our earlier work ([23]) can be reversed and turned into a linear assembly sequence. However,

they do not fully represent the precedence relations between parts across different layers. In contrast, the algorithm developed in this chapter combines part cluster detection and disassembly layer extraction to generate precedence constraints for assembly. This distinction is illustrated in Fig. 3.12. For instance, part 16 belongs to the first disassembly layer and part 7 belongs to the seventh disassembly layer (Fig. 3.12(a)). This implies that part 7 can be assembled before part 16. However, it is not clear if the converse is true. The assembly structure identified by the new approach (Fig. 3.12(b)) clarifies this precedence relation: Part 7 and part 16 have no precedence constraint with respect to each other. That is, the set of parts ($\{2, 3, 4, 5, 6, 12, 13, 21, 22, 23\}$) that has to be removed before part 7 can be removed does not include part 16. The new assembly structure reveals these precedence relationships by reorganizing the parts in individual layers into interaction clusters, while maintaining the same hierarchy of layers found in Fig. 3.12(a). Figures 3.13(a) and (b) show the directed acyclic graph representation of the assembly structures obtained by the two techniques, respectively.

3.6.2 Complex Chassis Assembly

Now, we report results on a more complex assembly comprising 74 parts, which is obtained by augmenting the previous assembly with 51 new parts. Two views of this 74-part chassis assembly are shown in Fig. 3.14. The disassembly layer generation and the directed acyclic graph representation of the assembly structure are shown in Figures 3.15 and 3.16, respectively.

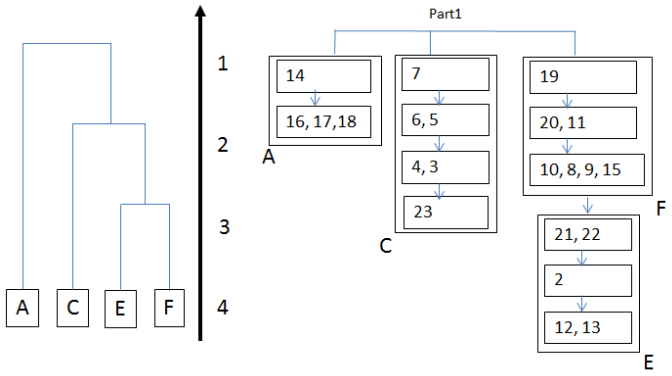


Figure 3.11: *(a) Part interaction clusters detected by the algorithm for the 23-part chassis assembly. (b) The corresponding assembly structure representing the precedence relations between different part sets and individual parts within each part set.*

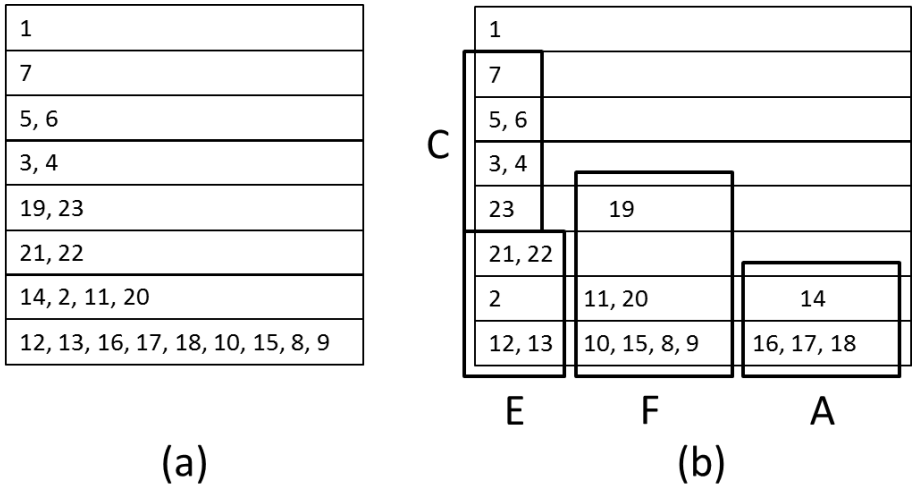


Figure 3.12: *Precedence relations between layers in the hierarchical exploration structure for the chassis assembly: (a) Simple disassembly generation. (b) Disassembly generation coupled with cluster detection.*

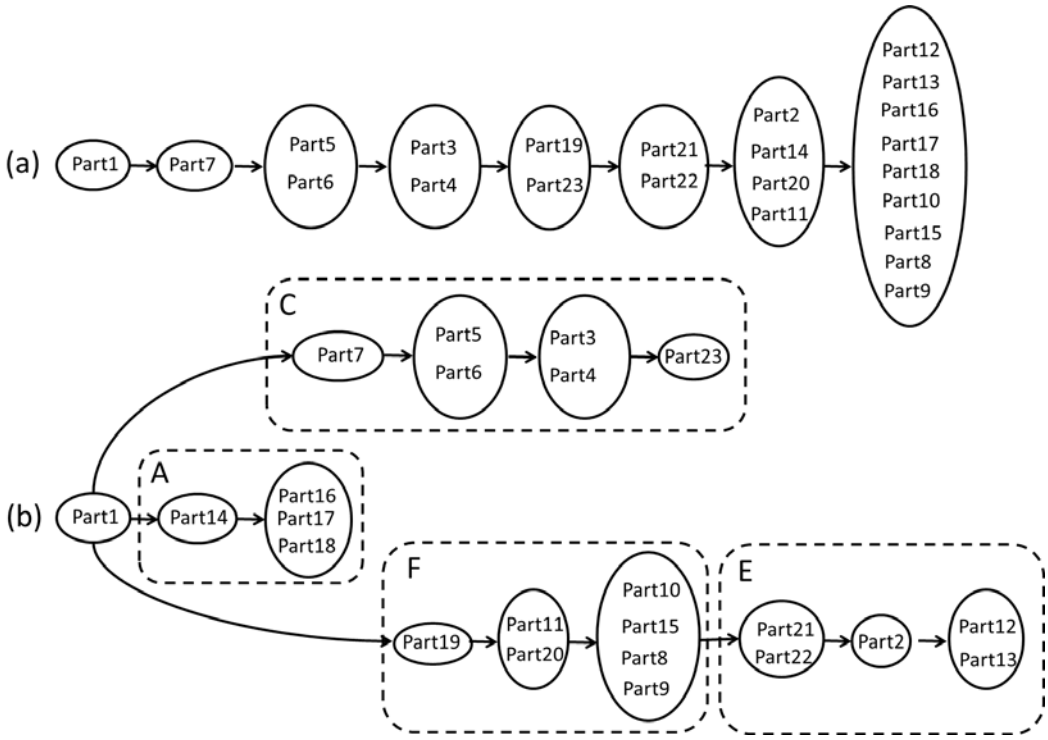


Figure 3.13: *Directed acyclic graph representation of the assembly structure for the chassis assembly: (a) Simple disassembly generation. (b) Disassembly generation coupled with cluster detection.*

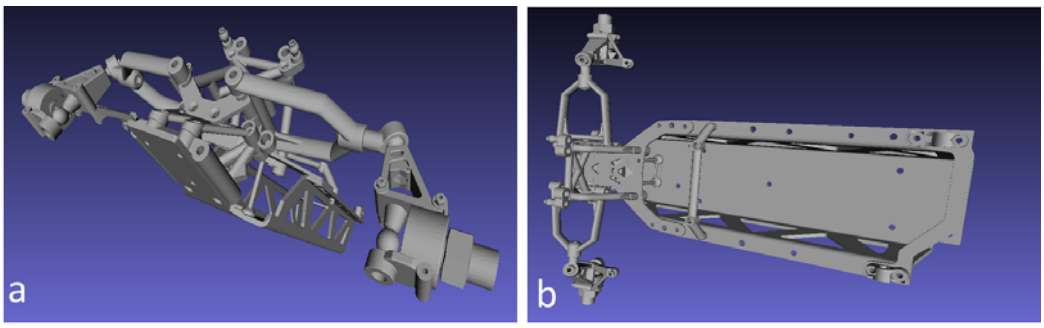


Figure 3.14: *The 73 part chassis assembly: (a) View 1. (b) View 2.*

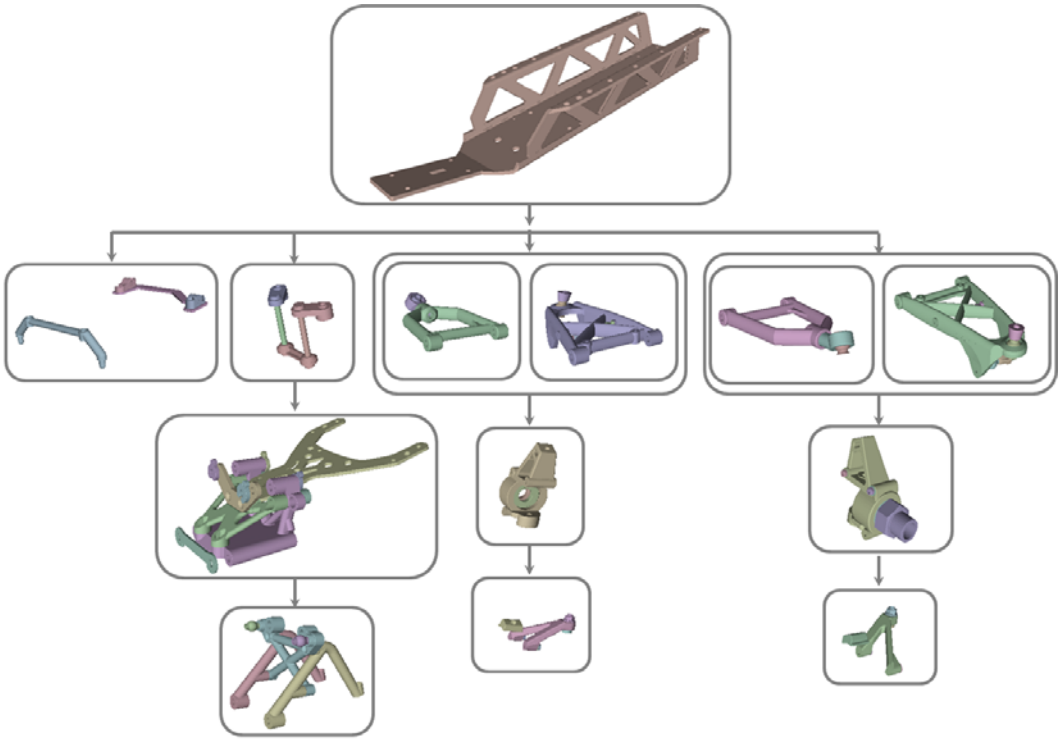


Figure 3.15: *Disassembly layer generation for the 73-part chassis assembly by combining motion planning and part interaction cluster detection methods.*

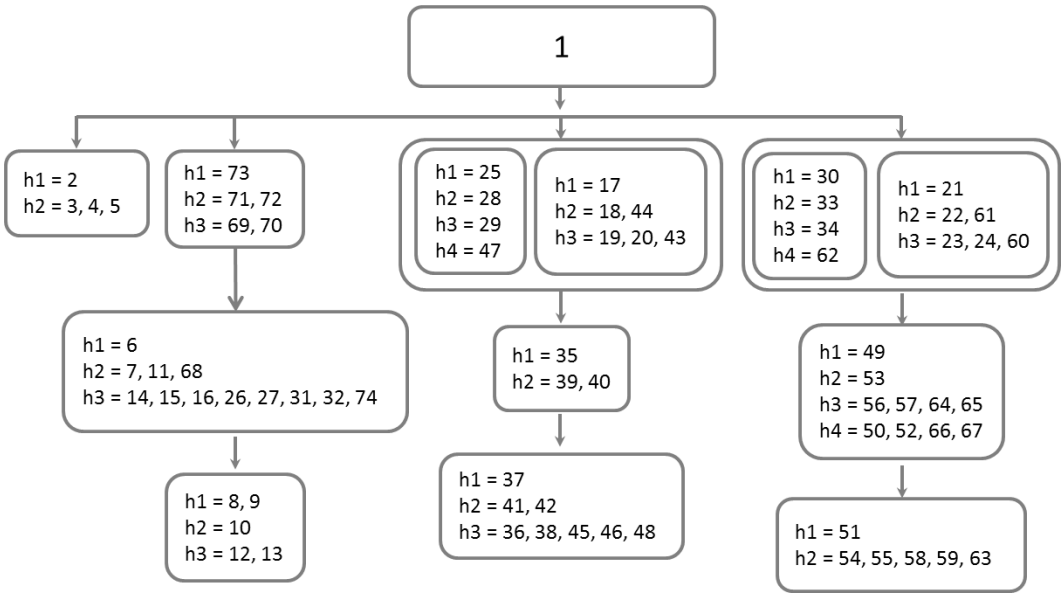


Figure 3.16: *Directed acyclic graph representation of the assembly structure for the 73-part chassis assembly.*

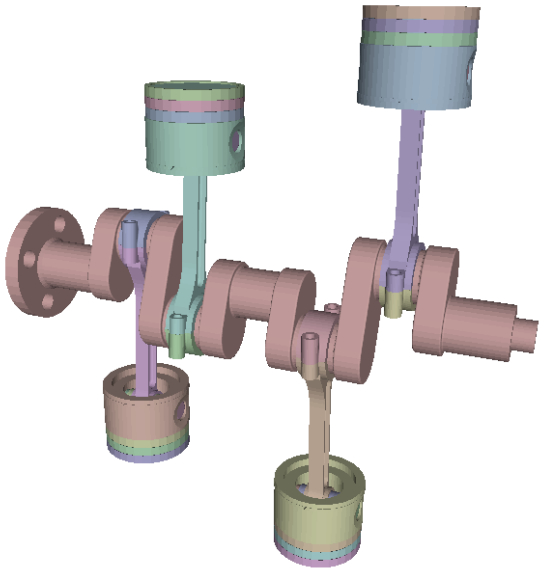


Figure 3.17: *Crank shaft assembly*

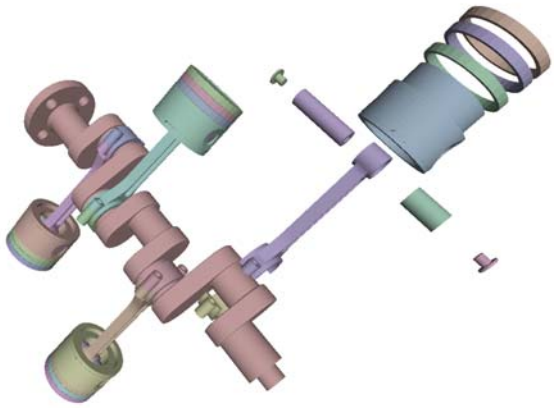


Figure 3.18: *The disassembly generation of one part interaction cluster identified by the algorithm for the crank shaft assembly.*

3.6.3 Crankshaft Assembly

The crankshaft assembly, shown in Fig. 3.17, consists of 41 assembly parts: One crank shaft, 4 shafts, 4 lower shafts, 4 inner bearings, 4 bearings, 8 pins, 12 rings, and 4 piston heads. The disassembly generation of a past-set identified by the algorithm is shown in Fig. 3.18. Figure 3.19 shows the directed acyclic graph representation of the assembly structure extracted by the algorithm. From this figure, it is clear that the algorithm correctly identifies a feasible nested configuration of part set.

3.6.4 Radial Crankshaft Assembly

The radial crankshaft assembly, shown in Fig. 3.20, consists of 50 assembly parts: One master shaft, one radial bearing, 4 shafts, 4 bolts, 5 inner bearings, 5 bearings, 10 pins, 15 rings, and 5 piston heads. The disassembly generation of a part set identified by the algorithm is shown in Fig. 3.21. Figure 3.22 shows the directed

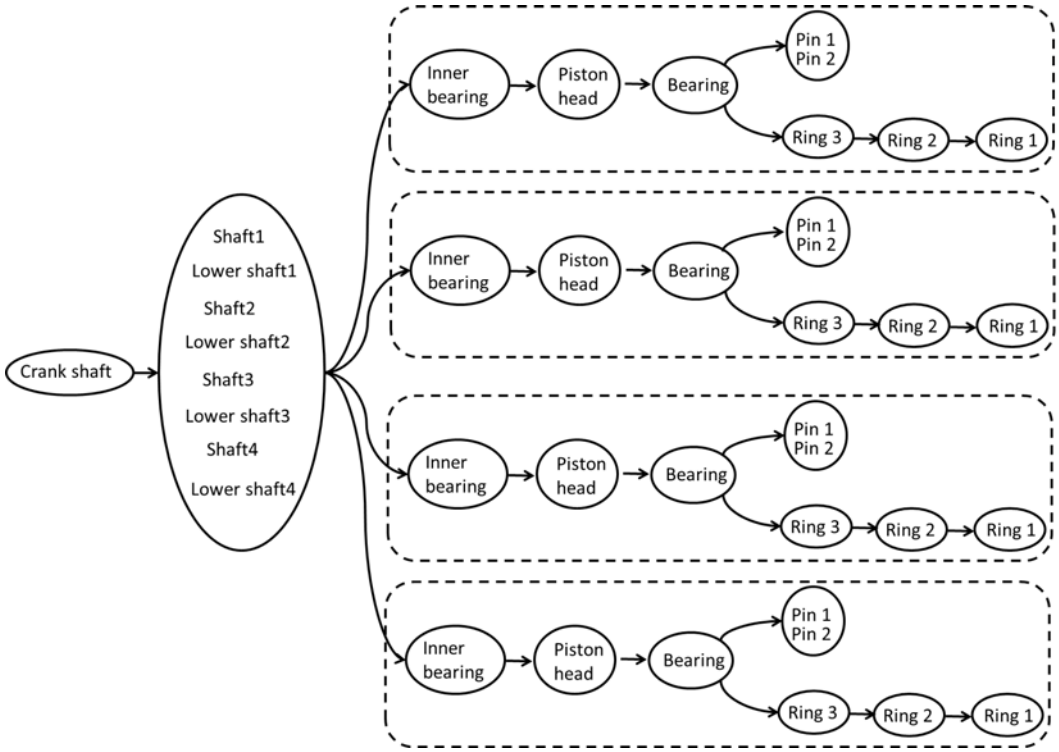


Figure 3.19: *Directed acyclic graph representation of the precedence relations between assembly parts and part sets for the crank shaft assembly.*

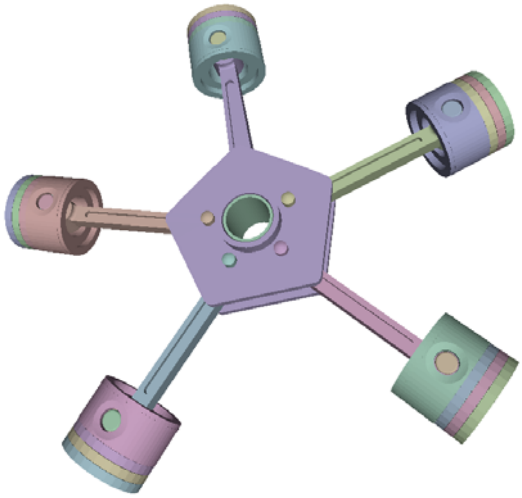


Figure 3.20: *Radial engine assembly*

Figure 3.21: *The disassembly generation of one part set identified by the algorithm for a Radial engine assembly.*

acyclic graph representation of the assembly structure extracted by the algorithm. From this figure, it is clear that the algorithm correctly identifies a feasible nested configuration of part set.

3.6.5 The Five Parts Puzzle

The five-part puzzle (Fig. 3.23) is used to illustrate the ability of the algorithm to deal with the issue of nonlinearity. In particular, this assembly example is representative of a scenario in which some parts cannot be assembled one by one in a linear order; rather, they must be simultaneously moved to their final assembly locations. The algorithm implementation can be explained in the following steps:

1. Find part sets to test feasibility with $C_{min} = 1$. No feasible partition is found with $k = 1, 2, 3$ and 4.

2. Test feasibility with the largest part + a single part containing the four remaining parts. The assembly structure passes the test. The result of this

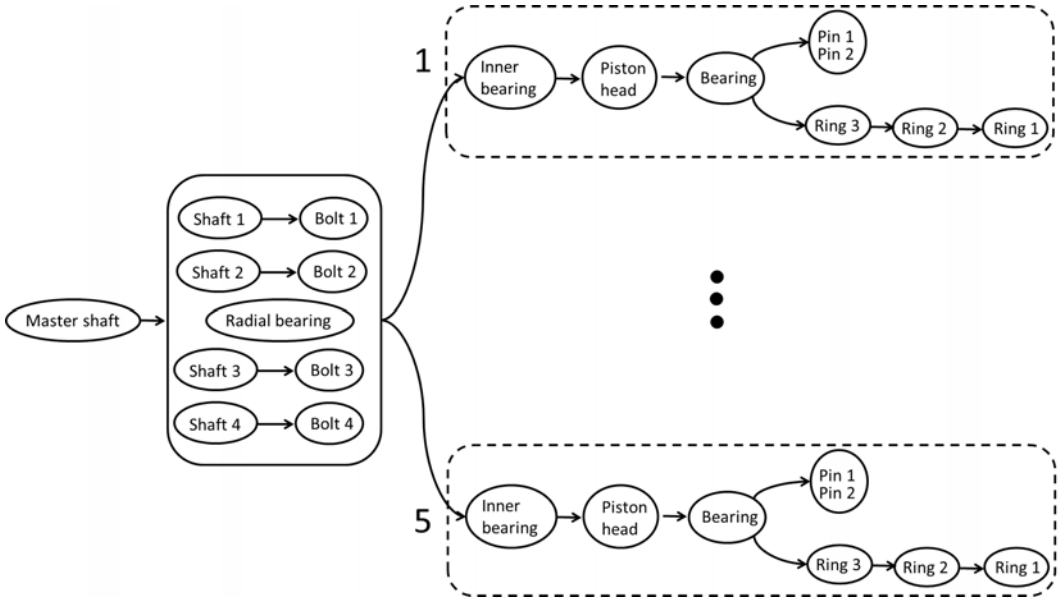


Figure 3.22: *Directed acyclic graph representation of the precedence relations between assembly parts and part sets for a Radial engine assembly.*

step is that the set of four parts must be assembled before placing in the final position.

3. Recursively partition the four-part set and test feasibility. This results in two part sets A and B.

The resulting assembly sequence is given below:

1. Assemble the two parts in part set A.

2. Assemble the two parts in part set B.

3. Assemble the previous two part sets into one single part set.

4. Assemble the resulting part set into its final position inside the largest part.

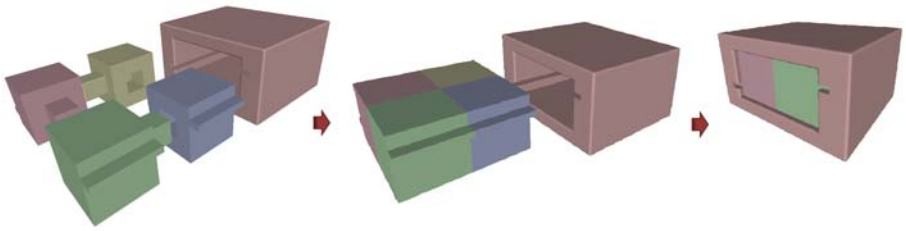


Figure 3.23: *The five-part puzzle used to illustrate how the algorithm deals with nonlinearity: Initially, the four parts form two part sets. Next, they combine to form a single part set, which is finally assembled into the largest part.*

3.7 Summary

We presented a framework that combined motion planning techniques and part interaction cluster extraction to guide the generation of feasible assembly sequences for complex mechanical assemblies. We showed that our approach can be used to handle complex assemblies in which assembly of part sets can be assigned to different stations and later, these part sets can be put together to form the final assembly. We also showed how our approach addresses the issue of nonlinearity in assemblies.

We have also demonstrated the minimization of the complexity in the assembly sequence generation by developing approximations and heuristics without significantly sacrificing the accuracy of the solutions. Combining motion planning techniques with unsupervised part interaction clustering reduces the number of candidate assembly sequences in the overall combinatorial problem.

As we showed in the previous sections, the complexity of assembly or disassembly sequencing is measured in terms of the number of parts. We demonstrated that our algorithm can generate in the worst case scenario, the same number of combinations (where not all combination of parts into subassemblies is allowed) as

the traditional sequence approach where the number of potential sequences is given by the number of permutations of parts.

Since assembly sequence planning is a NP-hard problem, it is not possible to come up with a complexity-reduction-factor between previous sequential approaches and our approach. Instead, to describe the reduction in the computational cost for our approach we can explore some simple examples. For instance, for a 5 parts assembly the number of potential sequences is 120. Many of these sequences are not valid but still the algorithm has to explore these potential candidates in order to verify the feasibility of the sequence. For the same 5 parts assembly and assuming three parts cluster boundary, the present framework reduced the number of permutations or potential sequences to 10.

Although the number of potential sequences is large for the sequential approaches, this does not mean that once the algorithm finds a feasible sequence it still keeps exploring the solution space. In many of the cases the goal is to find a feasible assembly sequence. The problem in general is: when can we find this feasible assembly sequence? This could be at the beginning of the exploration and therefore, considerably reducing the computation cost or could be at the end, hence representing a full exhaustive search. Since the precedence constraints are being discovered gradually, there is no way to predict the number of permutations before the first feasible assembly sequence is found. Our algorithm showed some variations in the number of permutations due to the unsupervised number of clusters but these variations are not as large as the previous approaches.

We have described a practical approach to generating assembly sequences from

a geometric model of the target assembly. This approach has been implemented and tested in the present framework. The system generated the assembly precedence constraints directly from 3D assembly models as it uses only information about geometric and physical constraints. In situations where part assembly involves a single translation and a single rotation, methods without using path planning are computationally more efficient than our approach. However, our approach performs better when applied to more complex assemblies that involve multiple translations and rotations. Our approach explores many different feasible assembly directions. Our approach checks for feasibility in a given assembly state. So it can handle situations where different directions become feasible based on the state of the assembly. For example, if the left direction is blocked in a given assembly state, and the right direction is checked for feasibility and selected (if appropriate).

Our approach presented in the chapter considered each part as a free-flying object. However, in a realistic scenario, as tools and human hand operations will be used during assembly, some of the precedence constraints generated by our approach may become infeasible. In order to address this issue, our approach must be augmented by incorporating motion planning for human-hand and tool models into the formulation. This will result in the generation of modified precedence constraints that cater to the spatial constraints imposed by the tools and the hands. Our technique was proposed for mechanical assemblies that are composed of finite numbers of non deformable parts. Hence, assemblies that contain flexible parts, where the flexible property of a part is exploited to fit it into the assembly, were out of the current scope of our current approach. In addition, an analysis beyond pure

geometric reasoning is required for assemblies that contain parts requiring force for placement.

Chapter 4

Toward Safe Human Robot Collaboration by using Multiple Kinects based Real-time Human Tracking

In this Chapter[1], we present a multiple Kinects based exteroceptive sensing framework to achieve safe human-robot collaboration during assembly tasks. Our approach is mainly based on a real-time replication of the human and robot movements inside a physics-based simulation of the work cell. This enables the evaluation of the human-robot separation in a 3D Euclidean space, which can be used to generate safe motion goals for the robot. For this purpose, we develop an N-Kinect system to build an explicit model of the human and a *roll-out* strategy, in which we forward-simulate the robot's trajectory into the near future. Now, we use a pre-collision strategy that allows a human to operate in close proximity with the robot, while pausing the robot's motion whenever an imminent collision between the human model and any part of the robot is detected. Whereas most previous range based methods analyzed the physical separation based on depth data pertaining to 2D projections of robot and human, our approach evaluates the separation in a 3D space based on an explicit human model and a forward physical simulation of the robot. Real-time behavior ($\approx$ 30 Hz) observed during experiments with a 5 DOF articulated robot and a human safely collaborating to perform an assembly task

[1] The work in this chapter is partially derived from the published work [25] and [26].

validate our approach.

4.1 Introduction

Robots excel at performing tasks−welding, component soldering, bolting, packaging−requiring speed, repeatability, and high payload capabilities. However, humans are better at manipulation of a wide range of parts without using special fixtures; they also have a natural ability to handle unexpected situations on the shop floor. Therefore, collaborative frameworks in which humans and robots share the workspace and closely work together to perform manufacturing tasks can lead to increased levels of productivity.

Safety is one of the primary challenges encountered while trying to introduce robots into anthropic environments [102, 159, 160]. Traditionally, safety in work cells is ensured by caging a robot with either a physical [161] or virtual [162] barrier and sequencing the roles of the robot and the human; that is, the robot is rendered inoperative whenever a human enters the robot's work cell to perform his/her task. However, this segregation paradigm leaves no scope to realize the proposed benefits of human-robot collaboration (HRC).

Strategies to achieve safe HRC can be broadly divided into two categories: Pre-collision [94, 95, 96, 97, 98, 99, 100] and post-collision [102, 103, 163]. The former problem deals with devising controllers that allow the robot to prevent imminent collisions with a human. However, the latter aims to reduce the impact/injury after an unexpected human-robot collision has occurred. One example is a human-

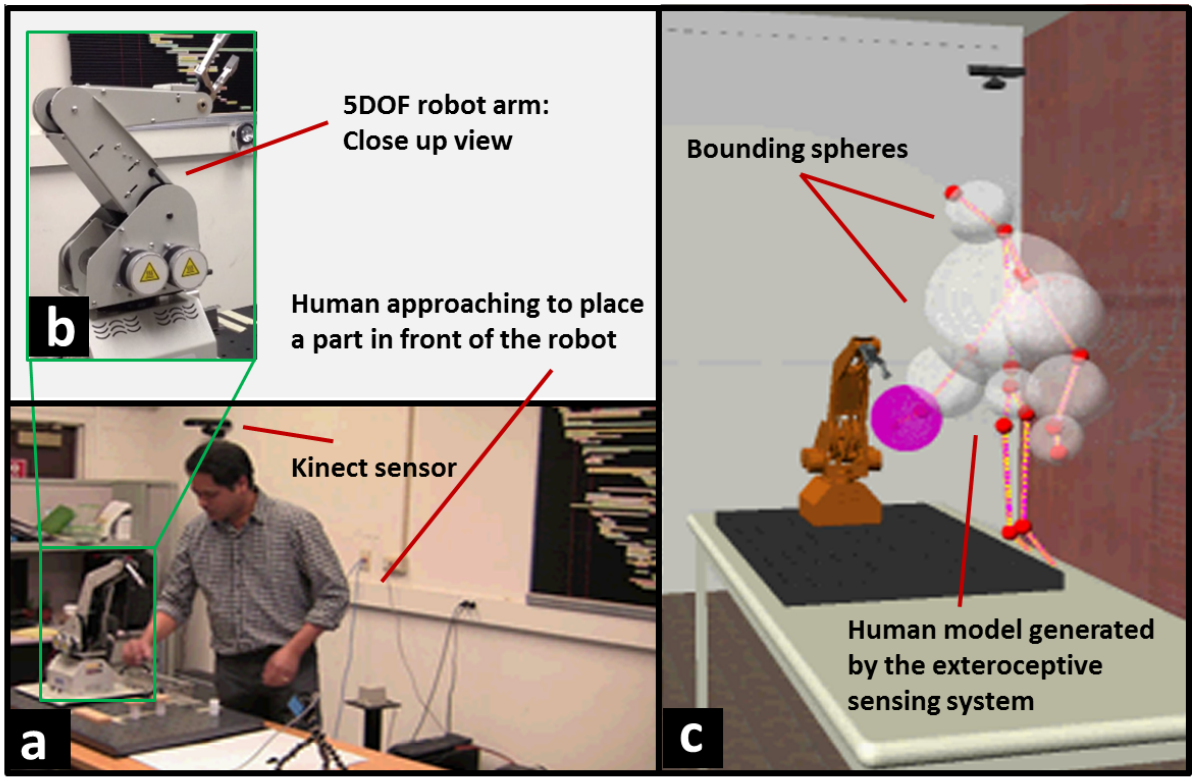


Figure 4.1: *Overall system overview: (a) Work cell used to evaluate human-robot collaboration. (b) 5 DOF robot used for the experiments. (C) Physical simulation used to evaluate the interference between the human and the robot in real-time.*

friendly robot designed by Shin et al. [163]. In this chapter, we limit the scope of our literature review to pre-collision strategies as the methods presented in this chapter belong to this category.

The underlying principle of most pre-collision methods consists of calculating the physical separation between the robot and the human, tracking the changes in the separation, and enabling the robot to take preventive actions whenever the separation is below a specified threshold. Separation monitoring in shared workspaces has been identified as one of the important aspects for which performance metrics are appearing in the recent literature [164]. Successful deployment of human robot collaboration systems involves a proper integration between low-loop control loops and high-level planners [165]. In this context, separation monitoring also provides the perceptual feedback required to implement expressive temporal planners [166], which integrate sharable resource management into plan generation. This feedback can also be used in conjunction with assembly planners [82, 25] and instruction visualization tools [130, 135, 27] to modify assembly plans and assembly instructions appropriately.

Note that the separation of interest is not a simple Euclidean distance between two points, since a collision can occur between any part of the human and any part of the robot during a collaborative task. Moreover, certain parts of the human (robot) have more probability of collision with the robot (human) than certain others. For example, consider a human and a desktop-robot manipulator working in close proximity w.r.t. each other. In this HRC scenario, the human's hands are exposed to the arms of the robot with a higher frequency than that of his/her

trunk; the human's legs never come into contact with the robot as they always operate below the surface of the assembly table.

All these issues raise important questions of how to model the robot-human separation, how to design sensing methods to accurately measure the model variables, and how to incorporate the resulting data into robot control for ensuring safety in the work cell. Previous approaches that address these challenges mainly differ from one another depending on (a) how the human's motion is accounted for; that is, whether the human is tracked by using an explicit 3D human model or he/she is treated as equivalent to other obstacles in the work cell, (b) if a human model is used, then what sensing method is used to build the model, and (c) what control algorithm is used to prevent collisions between the human and the robot during the course of their collaborations.

Recent advances in computer game interfaces have enabled their use as tools for interaction with robots. For example, Smith and Christensen [167] presented a method to use wiimote controller to track human input based on human motion models. Similarly, Kinect is another low-cost sensing device that is recently being used for HRC applications [96].

In this chapter, we present a multiple Kinects based exteroceptive sensing framework to achieve safe human-robot collaboration during assembly tasks. An overview of the overall system is shown in Fig. 4.1. A preliminary implementation of the system was presented in [26]. Our approach consists of a real-time replication of the human and robot movements inside a physics-based simulation of the work cell. This enables the evaluation of the interactions between them in a three

dimensional Euclidean space, which can be used to generate safe motion goals for the robot. First, we develop an N-Kinect based framework that builds an explicit model of the human in near real-time. In particular, the sensing system consists of multiple Microsoft Kinects mounted at various points on the periphery of the work cell. Usage of multiple Kinects accounts for problems caused by occlusion. Each Kinect monitors the human and outputs a 20-joint human model. Data acquired from all the Kinects are fused in a filtering scheme to obtain a refined estimate of the human's motion. Second, the generated human model is augmented by approximating pairs of neighboring joints with dynamic bounding spheres that move as a function of the movements performed by the human in real-time. Third, we implement a *roll-out* strategy in a physics-based engine, where we forward-simulate the robot's trajectory into the near future, creating a temporal set of robot's postures for the next few seconds; now, we check whether any of these postures collides with one of the bounding spheres of the human model. Fourth, we use a pre-collision strategy that allows a human to operate in close proximity with the robot, while pausing the robot's motion whenever an imminent collision between the human and any part of the robot is detected. Whereas most previous range based methods analyzed the physical separation based on depth data pertaining to 2D projections of robot and human, our approach is one of the first successful attempts to evaluate human-robot interference in a three dimensional Euclidean space based on an explicit human model and a forward physical simulation of the robot. Real-time behavior ($\approx$ 30 Hz) observed during experiments with a 5 DOF articulated robot and a human safely interacting to perform a shared assembly task validate the ef-

fectiveness of our approach.

4.1.1 Kinect Sensor

Human motion estimation is an active field of research with a diversity of approaches. Multi-camera systems are among the most traditional methods for 3D tracking. One of the typical approaches for motion reconstruction of freely moving humans employs shape-from-silhouette for estimating shape from multiple structures. Another family of motion extraction methods relies on efficient descriptor-based tracking. These descriptors can be visual or body attached sensors such as magnetic, mechanical or passive markers.

Recently with the introduction of Microsoft Kinect sensor, a lot of attention has been focused on depth sensors/cameras. Kinect sensor captures motion in real time (approximately 30 fps) and releases 2.5D data of resolution 640 x 480 accompanied with registered RGB data. One of the major advantages of the Kinect sensor is its ability to infer human motion by extracting human silhouettes in skeletal structures. Several researchers have used Kinect-based tracking for human activity recognition applications. However, most researchers use single sensor solutions due to problems encountered with sensor interference with each other. Multiple Kinect sensor interference can result in erroneous or missing depth estimates. As a consequence, self-occluded body parts or conditions where parts of the body are occluded by other objects cannot be handled.

The human motion tracking framework presented in this work is designed for

a hybrid assembly cell where one human operator interacts with one robot in a 3D environment to perform an assembly operation. In this design, the human operator has complete freedom of his/her motion. Human operator activity is captured by the Kinect sensors that reproduce the operator's location and movements virtually in the form of a simplified animated skeleton for the system that also controls the robot. The human motion tracking framework presented in this work uses multiple Kinect sensors to cover a large work space and overcome the problem of occlusion and self-occlusion and produce reliable skeletons and 3D reconstructions.

4.1.2 Comparative effectiveness of Kinect-based systems with existing systems

Modern motion capture systems have taken a variety of approaches to solving the problem of accurately tracking human motion. Generally, these systems track motion using two different mechanisms, optical and non-optical capturing. Each tracking mechanism has advantages and disadvantages when applied to different specified problem domains. Bellow we present a comparison between the different motion capture technologies and their application in the specific domain of tracking a human operator in a very diverse and crowded environment, the assembly cell.

The following list below represents the comparative effectiveness of Kinect-based motion tracking systems with other types of systems.

- **Mechanical**
  - *Description*: Human operator wears a human-shaped set of straight

metal pieces (like a very basic skeleton) that is hooked onto the human's back; as the human operator moves, this exoskeleton is forced to move as well and sensors in each joint perceives the rotations.

- *Advantages*: No interference from light or magnetic fields that can be present in the assembly cell.
- *Disadvantages*:
  1. Technology has no awareness of ground, so there can be no jumping; moreover, feet data tends to slide.
  2. The equipment or exoskeleton must be calibrated often.
  3. Unless there is some other type of sensor in place, it does not know which way the human's body is pointing.
  4. Absolute positions in the assembly cell are not known but can be calculated from the rotations.
  5. In an environment where the human performance is critical the additional exoskeleton limits the motion of the human.
- *Precision and Frequency*: 0.36 degrees - 120Hz.
- *Cost*: Low cost US$ 10000.

- **Optical**
  - *Description*:
    * Human operator wears reflective descriptors that are followed by several cameras and the information is triangulated between them.

    * Markers are either reflective, such as a system manufactured by VICON or Motion Analysis, or infra-red emitting.

  – *Advantages*:

    1. Human operator has the freedom to move in the assembly cell due to no cables connecting the body to the equipment.
    2. Very clean, and detailed data at higher rates are available.
    3. Rotations of body parts must be solved for and are not absolute.
    4. Human operator must wear a suit with descriptors and balls (20-30 for body), which may be uncomfortable.
    5. Information has to be post-processed or $\ddot{\text{t}}\text{racked}\ddot{\text{b}}$efore viewing so human operator cannot see his/her image and so cannot have the instant feedback to identify potential.

  – *Disadvantages*:

    1. It is prone to light interference.
    2. Reflective descriptors can be blocked by the robot or other structures, causing loss of data, or occlusion.

  – *Precision and Frequency*: Passive descriptors 0.1mm-240Hz. Active descriptors 0.07mm- 30-480Hz.

  – *Cost*: Higher cost than magnetic US$ 50,000 to 250,000.

- **Magnetic**

  – *Description*: Human operator wears an array of magnetic receivers which

track location with respect to a static magnetic transmitter. One of the first uses of this technology was by the military that used such systems to track head movements of pilots. Often this type of motion capture system is layered with animation from other input devices.

- *Advantages*:
  1. Positions are absolute, rotations are measured absolutely; Orientation in space can be determined, which is very useful.
  2. Can be real-time, which allows immediate broadcast as well as the opportunity for performers to puppeteer themselves with instantaneous feedback (more spontaneity in the performance).
- *Disadvantages*:
  1. Magnetic distortion occurs as distance to the origin increases.
  2. Data can be noisy and it is not as good as optical,
  3. Prone to interference from magnetic fields. Cement floors usually contain metal, so stages must be built.
  4. Human operator wears cables connecting them to a computer, which limits their freedom in the assembly cell.
- *Precision and Frequency*: 0.76mm-240Hz.
- *Cost*: Relatively cheaper than optical price under US$40,000 for a typical system.

- **Kinect**

- *Description*: Descriptorless motion capture systems use advanced computer vision technology to identify and track subjects without the need for any special suits or descriptors. Without the aid of descriptors to provide hints to the image processing software, advanced algorithms are required, especially when the goal is to track motion in real-time.
- *Advantages*:
    1. There are many clear advantages to descriptorless motion capture. Because no special suits, descriptors or equipment are required, human operators can simply step into the assembly cell to begin tracking.
    2. Because there is no special setup required, it is easier to track the motion of the human operator in the assembly cell in whose condition makes the application of special suits difficult.
    3. Provides a cost effective solution.
- *Disadvantages*: Implementing accurate tracking algorithms that perform well enough for real-time use, without the aid of descriptors to provide hints to the software is difficult.
- *Precision and Frequency*: 3mm-30Hz
- *Cost*: Under US$2000

Part of what makes human motion capture in the assembly cell such a big challenge is the speed at which everything must occur; for example, within 1/30th

of a second, the length of one frame of video, motion must be sampled, data must be applied to a digital scene representing various body parts of a character, and a scene must be rendered into a virtual environment. Depending on the system used, interference of the signals can impede accurate collection of data.

Even though, Kinect sensor (depth camera) offers the lower values in precision and frequency compared with mechanical, magnetic and descriptor-based motion systems, the characteristics of comfort, easy deployment, and low cost make this technology very suitable for our application.

Comfort: In a realistic scenario, human operators have to work 8 hours a day and 40 hours a week in a regular shift in a factory. Most of the work that they perform is physical work and requires their full mobility. This implies an inability to wear heavy suits that can limit the human operator motion.

Easy deployment: Optical motion capture systems require an exhaustive calibration not only of the hardware but also of the human. In addition, the operator has to wear the descriptors in order to be tracked. Therefore, all the operators that have interaction with the robot have to wear the descriptors (passive or active) or switch suits every single time that they enter the assembly cell. The same applies to magnetic and mechanical motion tracking wearing devices. Moreover, descriptors can be "confused" with other material in the assembly cell with similar color and intensity. A Kinect motion capture system has minimum illumination requirements. In order to track the human operator in the assembly cell, Kinect motion capture system does not require the human to wear any special suit or additional device.

Low cost: The very accessible price of the Kinect sensor and the possibilities to

integrate and refine its output data makes the Kinect-based human motion capture a very suitable solution.

4.2 Real-time Human Motion Tracking

Tracking of the human inside the work cell is achieved by generating a skeleton-like model of the human and by estimating the 3D positions of its joints in order to determine the human's movements. For this purpose, we use an N-Kinect based exteroceptive sensing system, which consists of multiple Kinects mounted at various points on the periphery of the work cell. Each Kinect monitors the human and outputs a 20-joint human model (Fig. 4.2) in its local reference frame. Positional data from all the Kinects are fused in a filtering scheme in order to obtain a refined human model in the global frame of reference.

Instead of processing the entire depth map, our sensing system works with a 20 DOF human model. This limited number of joints used to describe the human pose ensure the real-time operation of the framework, the scalability, and the latency free sensor fusion by reducing the number of variables to be processed and by reducing the amount of data to be transferred. Unlike previous gesture-based human tracking systems, usage of the Kinect doesn't require the human to wear any sensing-related devices. The specifications of the Kinect are shown in Table 4.1.

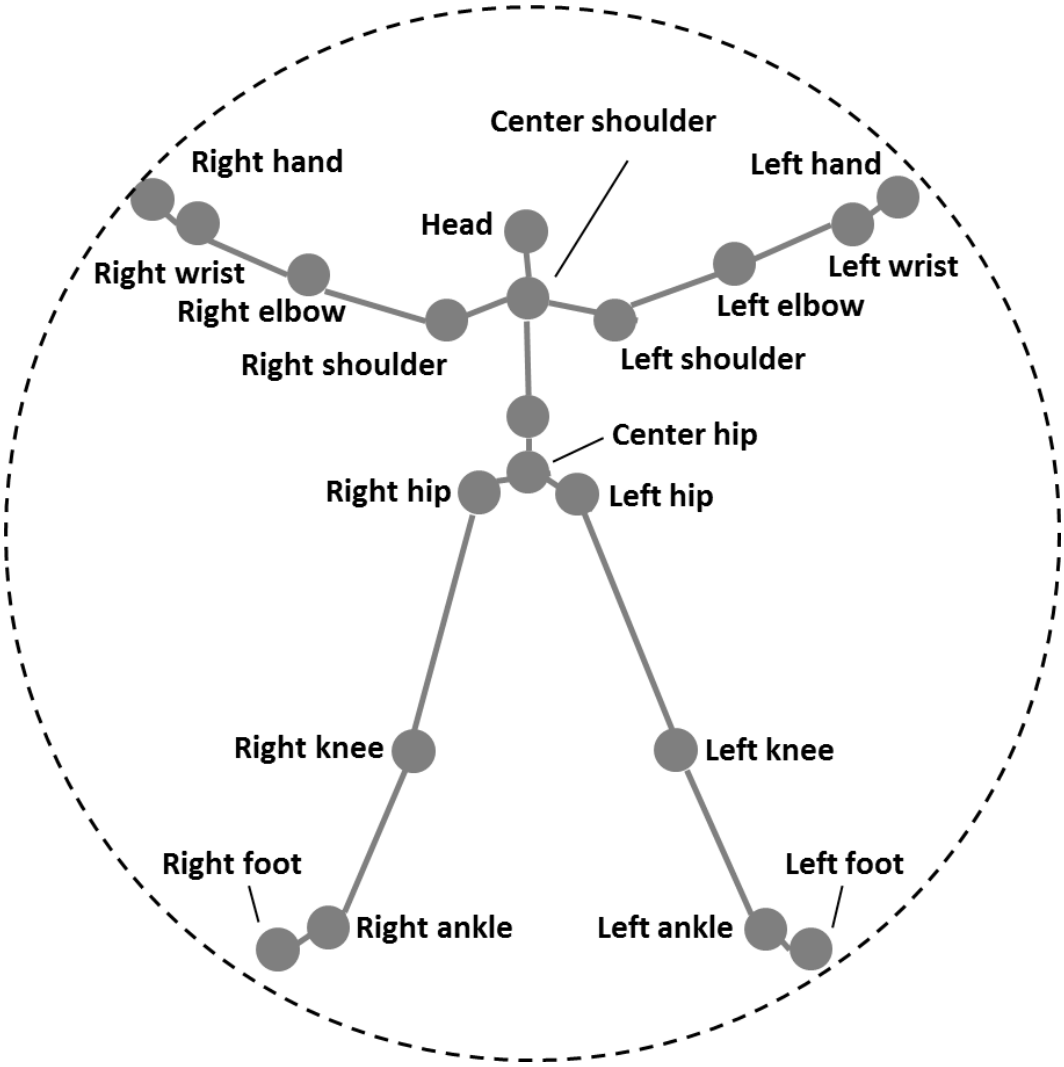


Figure 4.2: *The Microsoft Kinect directly outputs a 20-joint model of a human observed in a 3D scene*

Parameter	Specifications
Output	20-joint human skeleton model; 3D position coordinates for each joint given in meters
Operating range	0.8 to 3.5 m
Horizontal field of view	57^o
Vertical field of view	43^o
Spatial resolution	0.003 m
Depth resolution	0.01 m
Kinect SDK Version	V1.6

Table 4.1: *Kinect specifications used in the sensing design*

4.2.1 Exteroceptive Sensing Configuration

Design of the sensing configuration, given the work volume shared by the robot and the human, is mainly influenced by factors like shape of the workspace, number of sensors, placement of sensors, and presence of dead zones. We carry out a systematic experimental analysis of these factors in order to characterize the performance of the sensing system. In general, our objective is to achieve a coverage of the workspace by maximizing the number of fully tracked joints, while minimizing the number of sensors used.

4.2.1.1 Workspace Analysis

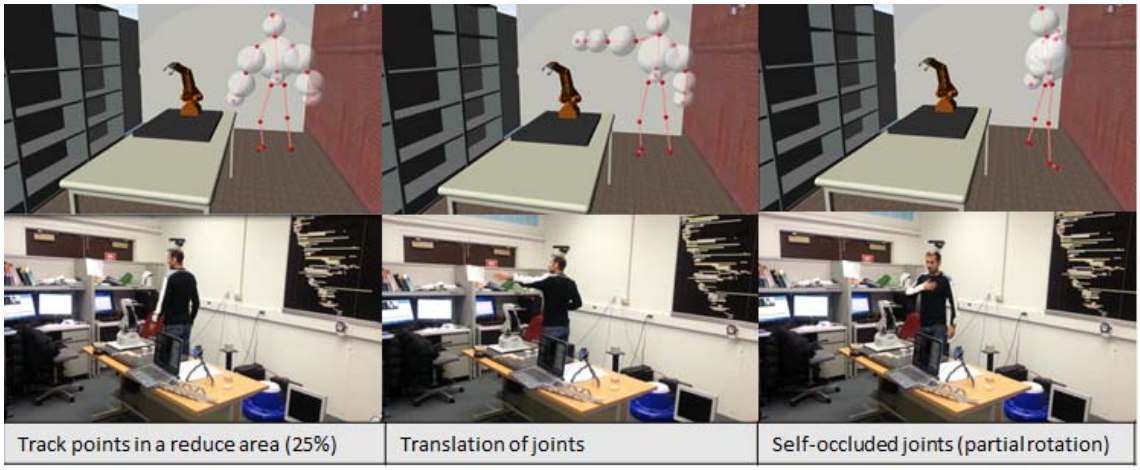


Figure 4.3: *1 Kinect sensor:*

Our framework considers a N-Kinect based sensing system, where N is the number of Kinects required to fully cover the work volume. The shape of the work volume considered in the experiments is cylindrical by nature. Therefore, there is no need for a Kinect to be placed directly above the robot. However, there is a need for multiple Kinects to be placed radially surrounding the periphery of the work cell. The exact placement of each Kinect in the radial direction and the angular separation between two neighboring Kiencts is guided by the operating range and the horizontal field of view of the Kinect (Table 4.1) and the dimensions of the work cell (4.72 m $\times$ 3.2 m $\times$ 2.7 m). The height and the pitch[2] (= -20^o) of each Kinect are selected such that a human with hands in a upright position is within the vertical field of view of the Kinect.

[2] Angle between the sensor axis and the horizontal plane

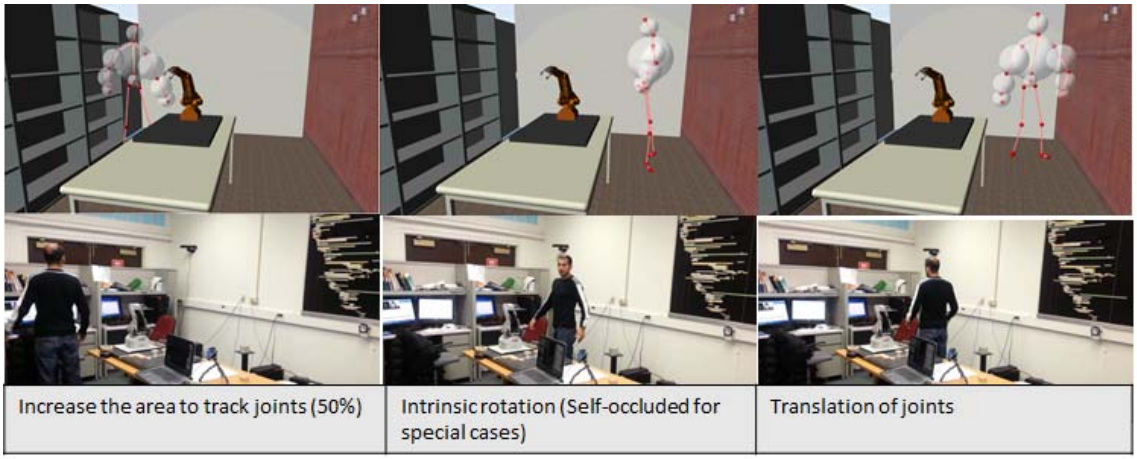


Figure 4.4: *2 Kinect sensors:*

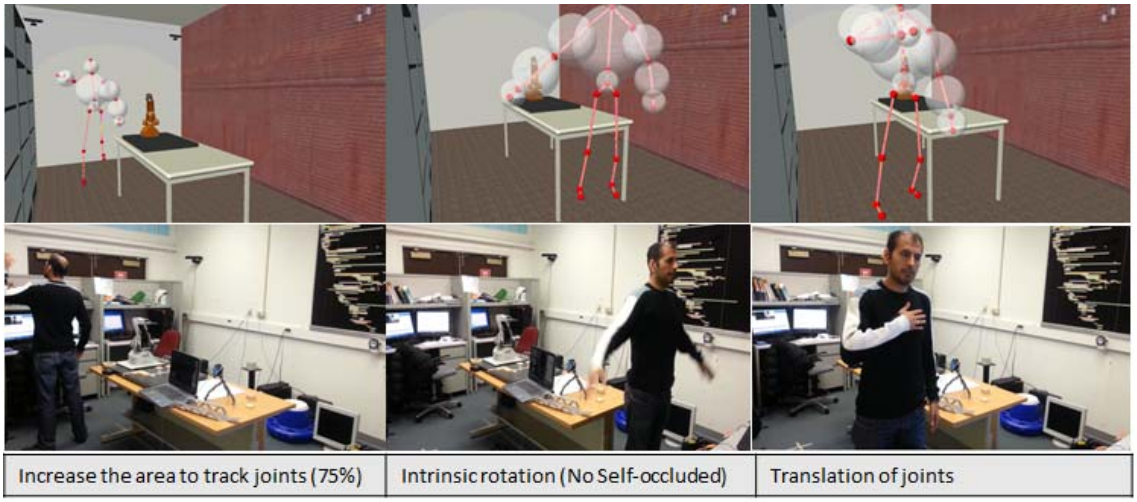
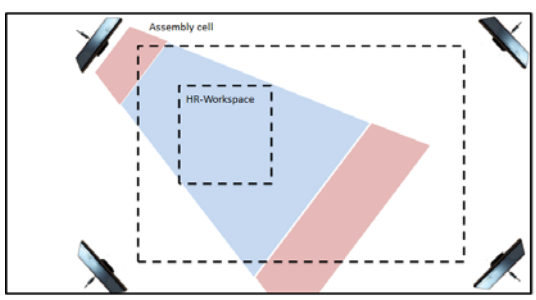
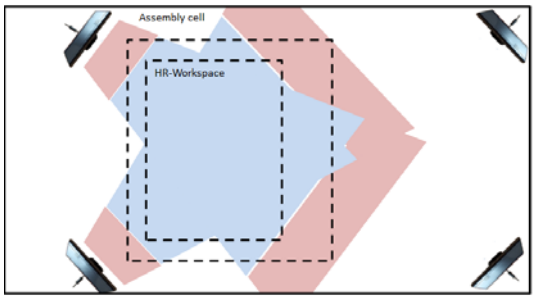


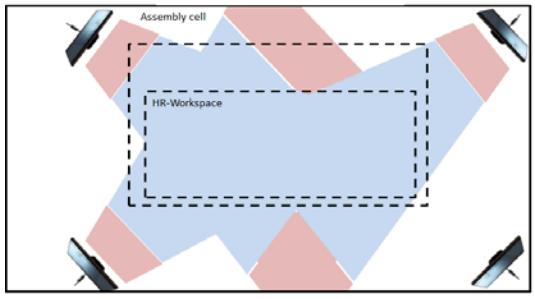
Figure 4.5: *3 Kinect sensors:*



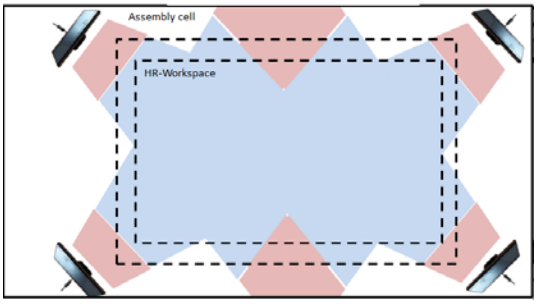
(a) *Human motion tracking with one Kinect Sensor*



(b) *Human motion tracking with two Kinect Sensors*



(c) *Human motion tracking with three Kinect Sensors*



(d) *Human motion tracking with four Kinect Sensors*

Figure 4.7: *Sensors placement an area of converge. Area in blue represent the recommended tracking area based on Kinect technical specification (for each sensor: 0.8m x 3.5 m x 2.1m). Area in red represent the additional area in sensor range*

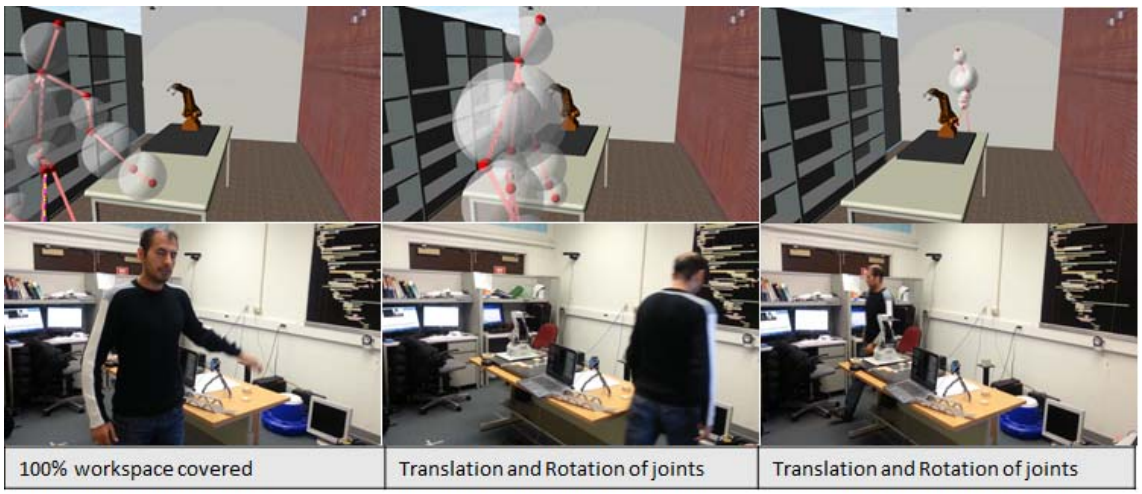


Figure 4.6: *4 Kinect sensors:*

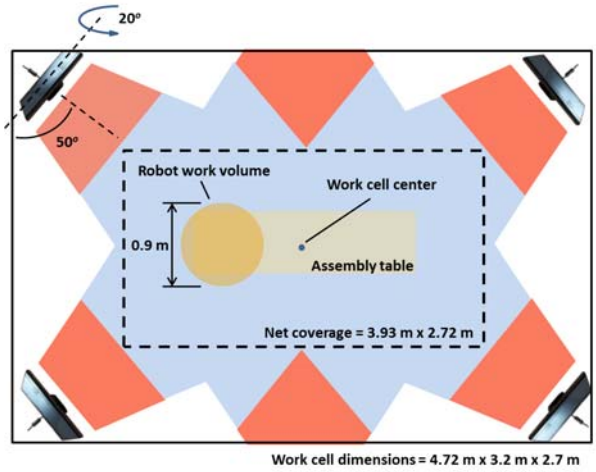


Figure 4.8: *Coverage (horizontal projection) obtained by using four Kinect sensors. The blue-color regions are fully covered. Red- and white-colored regions represent the dead regions of the work cell.*

4.2.1.2 Number of Kinects

Intuitively, coverage increases with an increase in the number of Kinects. However, the signals from multiple Kinects tend to interfere with each other. In particular, the infrared-ray pattern generated by the Kinect is not modulated in a way that the Kinect can recognize its own pattern; thereby, one Kinect could cast a ray that another Kinect defines as its own and hence incorrectly estimates the distance. Therefore, the number of Kinects must be chosen such that the coverage is maximized, while the interference between two neighboring Kinects is minimized.

We studied these effects by conducting the following experiment: We placed one Kinect at an appropriate distance to the center of the work cell and logged the values of metrics like workspace coverage[3], assembly cell coverage[4], implicit rotation, and the number of fully tracked human joints. Next, we incrementally added a new

[3] Ratio of area covered by the Kinect and total area of the workspace
[4] Ratio of workspace coverage and the total area of the work cell

Number of Kinects	1	2	3	4	5	6
Assembly cell coverage ($\approx$ %)	20	35	55	85	88	90
Workspace coverage ($\approx$ %)	25	50	75	100	100	100
Implicit rotation (degrees)	90	270	360	360	360	360
Number of fully tracked joints	4	8	20	20	20	20

Table 4.2: *Coverage as a function of number of Kinects*

Kinect at some angular separation to the previous Kinect (but at the same distance to the work cell center as that of the previous one) and recorded the readings again. A typical sensor arrangement with multiple Kinects mounted on the periphery of the work cell is shown in Fig. 4.8. The yaw[5] ($= 50^o$) of each Kinect is fixed at an angle such that the Kinect axis makes a small offset with the nearest diagonal of the work cell. This reduces the overlap with the Kinect facing diametrically opposite to it, thereby, increasing the net coverage due to the two Kinects.

Table 4.2 shows how the values of the metrics mentioned above varied as a function of the angle between two neighboring Kinects and the number of Kinects used up to the current step. From these experiments, we find that four Kinects mounted on the corners of the work cell are sufficient to cover the workspace. Note that there is no additional benefit in using more than four Kinects for the given work cell.

4.2.1.3 Dead zones

Dead zones correspond to regions which have either poor or no coverage. With respect to each Kinect (Fig. 4.8), the blue-colored region is fully covered and the red-colored region is poorly covered. Accordingly, from Fig. 4.8, the red- and

[5] Angle between the Kinect axis and the side wall

white-colored regions are the dead zones of the work cell. These sensing failures are handled by choosing the number of Kinects and their postures such that the workspace shared by the robot and the human is a proper subset of the union of the volumes covered by all the Kinects. From Fig. 4.8, note that the workspace marked as the dotted rectangle completely falls within the net coverage of all the Kinects.

4.2.2 Human Model Estimation

Each joint position of the human model generated by a Kinect p_{ij} (where i and j are the Kinect and joint indices, respectvely) is estimated by using a separate discrete Kalman filter. This results in a set of twenty local filters corresponding to twenty joints for each Kinect. Next, the resulting estimates of each joint j from all Kinects are used as inputs to a particle filter. This results in a set of twenty particle filters used to obtain improved estimates of all twenty joints.

The Kinect software cannot handle data from multiple Kinects. Therefore, individual models obtained from different Kinects are integrated via a communication architecture based on User Datagram Protocol (UDP). A client computer reads the positional data of the human model from each Kinect and transforms it into global coordinates. Next, the joint-position estimates from all 20×4 local filters are sent to the server, in which the particle filters are implemented.

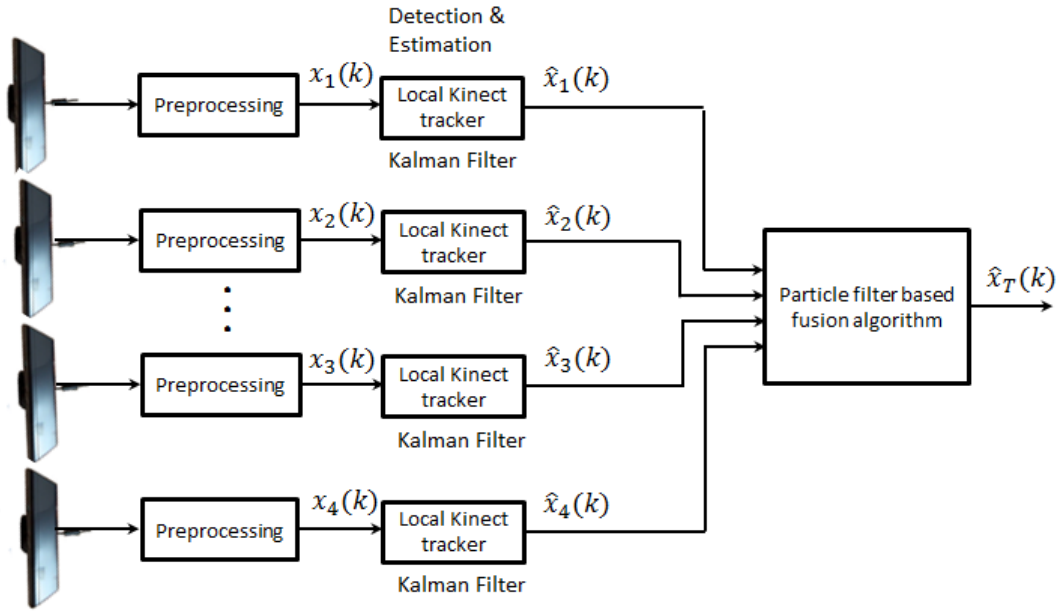


Figure 4.9: *Fusion architecture in which local tracks are generated at local Kinect sensor sites and then communicated to the central fusion unit.*

4.2.2.1 Local filter

We derive an approximate model of human motion as follows. Let $p_j = (x_j, y_j, z_j)$, $\dot{p_j} = (\dot{x_j}, \dot{y_j}, \dot{z_j})$, and $\ddot{p_j} = (\ddot{x_j}, \ddot{y_j}, \ddot{z_j})$ represent the position, velocity, and acceleration of each joint j. Writing the Taylor series expansion for position and velocity along the $x-$axis, we have

$$\begin{aligned} x_j(k+1) &= x_j(k) + \Delta T \dot{x}_j(k) + \frac{\Delta T^2}{2!}\ddot{x}_j(k) + \cdots \\ \dot{x}_j(k+1) &= \dot{x}_j(k) + \Delta T \ddot{x}_j(k) + \frac{\Delta T^2}{2!}\dddot{x}_j(k) + \cdots \end{aligned} \tag{4.1}$$

where k is a discrete time index and ΔT is the sampling time.

Similarly, we write the series expansions for position and velocity along the

other two orthogonal axes. Now, by neglecting the higher order terms, we obtain an approximate linear state model for each joint j as below:

$$\begin{aligned}
x_j(k+1) &= x_j(k) + \Delta T \dot{x_j}(k) + w_1(k) \\
\dot{x_j}(k+1) &= \dot{x_j}(k) + w_2(k) \\
y_j(k+1) &= y_j(k) + \Delta T \dot{y_j}(k) + w_3(k) \\
\dot{y_j}(k+1) &= \dot{y_j}(k) + w_4(k) \\
z_j(k+1) &= z_j(k) + \Delta T \dot{z_j}(k) + w_5(k) \\
\dot{z_j}(k+1) &= \dot{z_j}(k) + w_6(k)
\end{aligned} \tag{4.2}$$

where $w_i(k)$ is the noise in each state.

From (4.2), we obtain an approximate linear state model for each joint j as below:

$$X_j(k+1) = \mathbf{F} X_j(k) + W(k) \tag{4.3}$$

where

$$X_j(k) = \begin{bmatrix} x_j(k) \\ \dot{x_j}(k) \\ y_j(k) \\ \dot{y_j}(k) \\ z_j(k) \\ \dot{z_j}(k) \end{bmatrix}, \mathbf{F} = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{and}$$

$W(k) = \begin{bmatrix} w_1(k) & w_2(k) & w_3(k) & w_4(k) & w_5(k) & w_6(k) \end{bmatrix}^T$ is the system disturbance with a covariance matrix $\mathbf{Q}(k)$. If we assume $w_i(k) = 0$ for all k, then the acceleration and higher order derivatives are zero. This implies that the joint is moving at a constant velocity, which is not reflective of the actual motion of the human. Accordingly, we expect that the filter may not work well. Therefore, we address the question whether we can make it to work sufficiently well by assuming that each $w_i(k)$ is a zero-mean white random process and choosing the values of $\mathbf{Q}(\mathbf{k})$ appropriately. In particular, we model the process covariance terms using the formulation from [168]:

$$\begin{aligned} \mathbf{Q}(k) &= E[W(k)W(k)^T] \\ &= q \int_{t_k}^{t_{k+1}} F(t_{k+1}, \tau) F^T(t_{k+1}, \tau) d\tau \\ &\approx \end{aligned}$$

$$q\Delta T \begin{bmatrix} 1+\Delta T^2 & \Delta T & 0 & 0 & 0 & 0 \\ \Delta T & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+\Delta T^2 & \Delta T & 0 & 0 \\ 0 & 0 & \Delta T & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\Delta T^2 & \Delta T \\ 0 & 0 & 0 & 0 & \Delta T & 1 \end{bmatrix} \tag{4.4}$$

where q is the strength of the noise.

Note that we obtain only the joint position measurements from each Kinect. Consequently, let $Y_j(k+1) = \begin{bmatrix} x_j^m(k+1) & y_j^m(k+1) & z_j^m(k+1) \end{bmatrix}^T$ represent the position measurement[6] for joint j. Now, the measurement model for each joint j is given by:

$$Y_j(k+1) = \mathbf{H}X_j(k+1) + V(k+1) \tag{4.5}$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } V(k+1) = \begin{bmatrix} v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{bmatrix}$$

$V(k+1)$ is the measurement noise with a covariance matrix $\mathbf{R}(k+1)$.

[6] the Kinect index is omitted for brevity.

We make the following assumptions with respect to the various noise related variables: (a) Each $w_i(k)$, $(i = 1, 2, \ldots, 6)$ is a zero-mean white random process. (b) $\mathbf{v}_x(k+1)$, $\mathbf{v}_y(k+1)$, and $\mathbf{v}_z(k+1)$ are independent, zero-mean, and Gaussian noises with variances of $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 0.06$ m^2. (c) $W(k)$ and $V(l)$ are uncorrelated for all $k \geq 0$ and $l \geq 0$ [169]. (d) $W(k)$ and $V(k)$ are uncorrelated with the initial state $X(0)$ as their respective sources are different.

Let $X_j^-(k)$ represent the state prediction for k^{th} time step, $X_j'(k)$ represent the corrected state estimate after the measurement is made available, and $K_j(k)$ represent the Kalman gain. Let $\mathbf{P}_j^-(k)$ and $\mathbf{P}_j(k)$ represent the predicted and estimated error covariances in the state, respectively. Now, we implement the distributed discrete Kalman filter to estimate the state for each joint j by using Algorithm 5.

Algorithm 5 Kalman filter implementation for joint j

1: $k = 0$;
2: $X_j'(0) = E[X_j(0)] = \begin{bmatrix} x_j^m(0) & 0 & y_j^m(0) & 0 & z_j^m(0 & 0) \end{bmatrix}^T$;
3: $\mathbf{P}_j(0) = \mathbf{P}_0$;
4: $k \leftarrow k + 1$;
5: $X_j^-(k) = \mathbf{F}X_j'(k-1)$;
6: $\mathbf{P}_j^-(k) = \mathbf{F}\mathbf{P}_j(k-1)\mathbf{F}^T + \mathbf{Q}$;
7: $K_j(k) = \mathbf{P}_j^-(k)\mathbf{H}^T(\mathbf{H}\mathbf{P}_j^-(k)\mathbf{H}^T + \mathbf{R})^(-1)$;
8: $X_j'(k) = X_j^-(k) + K_j(k)(Y_j(k) - \mathbf{H}X_j^-(k))$;
9: $\mathbf{P}_j(k) = (\mathbf{I} - K_j(k)\mathbf{H})\mathbf{P}_j^-(k)$;
10: Go to Step 4;

4.2.2.2 Data fusion

As mentioned earlier, the position estimates of each joint j obtained from all the four Kinects are used as inputs to a particle filter [170]. The same state model derived in (4.3) is used here. For each joint j, the median of the state estimates

$\{X'_{1j}(k), X'_{2j}(k), X'_{3j}(k), X'_{4j}(k)\}$, represented by $X_j^M(k)$, is used as the input to the j^{th} particle filter at time step k. We assume that the measurement noise in each state ψ_i follows a Gaussian distribution with zero mean and variance σ_ψ^2. We can write the measurement model as below:

$$Y_j(k+1) \quad = \quad X_j(k+1) + V(k+1) \tag{4.6}$$

Now, we implement a particle filter for joint j using the pseudocode in Algorithm 6.

Algorithm 6 Particle filter implementation for joint j

$k = 0$;
$X_j^M(0) = \text{Median}[X'_{1j}(0), X'_{2j}(0), X'_{3j}(0), X'_{4j}(0)]$;
$Y_j(0) = X'_j(0) = X_j(0) = X_j^M(0)$;
Initialize N particles $\{\phi_{ij}(0) : i = 1, 2, \ldots, N\}$ from a Gaussian distribution $\mathcal{N}(X'_j(0), \mathbf{Q})$;
$k \leftarrow k + 1$;
$Y_j(k) = \text{Median}[X'_{1j}(k), X'_{2j}(k), X'_{3j}(k), X'_{4j}(k)]$;
$\omega_j(k) = 0$;
for $i = 1 : N$ **do**
    $\phi_i(k) = F\phi_i(k-1) + GW(k-1)$;
    $Y_{ij}(k) = \phi_i(k) + V(k)$;
    $\omega_{ij}(k) = \frac{1}{(2\pi)^3|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(Y_{ij}(k) - Y_j(k))^T \Sigma^{-1} (Y_{ij}(k) - Y_j(k))\right)$;
    $\omega_j(k) \leftarrow \omega_j(k) + \omega_{ij}(k)$;
end for
Generate a CDF Ω from the set of p.m.f.s assigned to the particles $\{\frac{\omega_{ij}(k)}{\omega_j(k)} : i = 1, 2, \ldots N\}$;
Resample the N particles $\{\phi_{ij}(k) : i = 1, 2, \ldots, N\}$ from Ω;
$X'_j(k) = (\frac{1}{N}) \sum_{i=1}^{N} \phi_{ij}(k)$;
Go to Step 5;

4.2.2.3 Estimation performance

The tracking performance of the filter is tested by conducting the following experiment: A human moves his wrist from one known point to another known point in the work cell and the measurements from all Kinects are collected and processed

using the filtering scheme built around Algorithm 5 and Algorithm 6. The tracking performance along x, y, and z axes are shown in Figures 4.10, 4.11, and 4.12, respectively. Each plot includes the ground truth values of initial and final positions, local measurements of the wrist-joint from one of the Kinects, the corresponding local Kalman filter output, the median of all the four Kalman filter outputs, and the particle filter output that provides the final estimate of the wrist-joint motion. Note from Fig. 4.10 and 4.12 that the scales used to plot the x and z graphs are different. Therefore, the margins between the measured and estimated values appear to be different in these graphs; but they are indeed similar to each other in reality. A 3D plot of this tracking data is shown in Fig. 4.13. Note that the particle filter acts upon the median output and provides a more refined estimation of the joint motion.

We test the estimation accuracy of the overall sensing system in the following way: A human is made to stand at different randomly selected known positions in the work cell. By assuming different postures at each position, ground truth data for a total of 15 postures are collected for the neck, shoulder, elbow, and wrist joints. Now, we compare this ground truth data to corresponding estimates provided by the sensing system. For illustration purpose, we use six out of these postures that are shown in Fig. 4.14. The discrepancy between the ground truth and estimated values are shown via projections of the joint positions on the XY plane (Fig. 4.15) and YZ plane (Fig. 4.16). In these figures, for each posture, a red-colored $*$ and a green-colored $*$ represent the ground truth and the estimated values for the neck-joint, respectively. Figure 4.17 shows the discrepancy values for each joint averaged

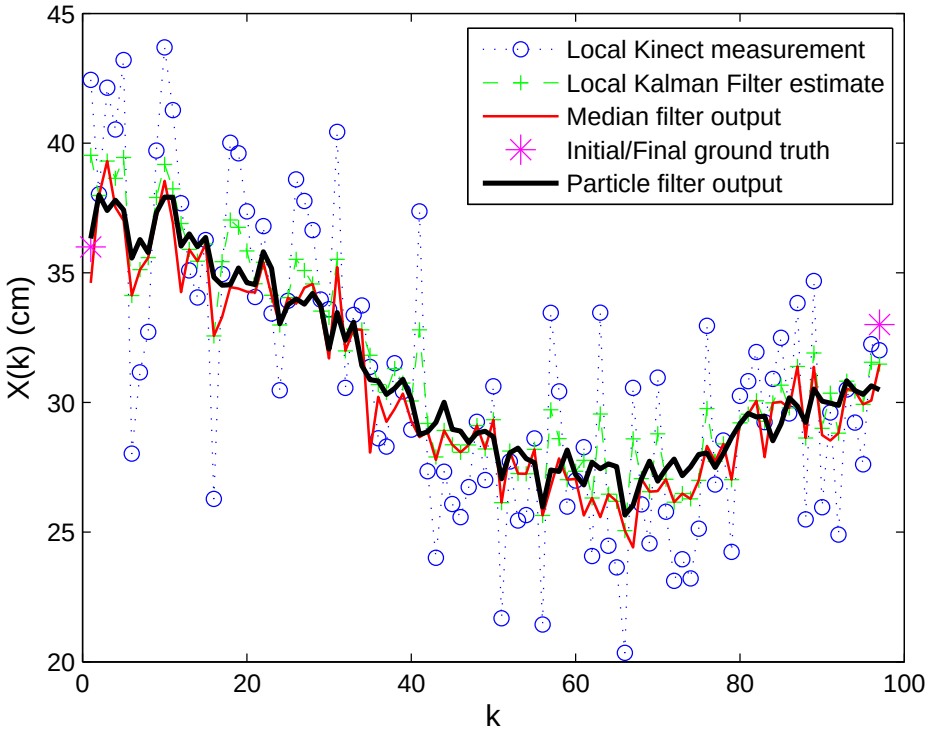


Figure 4.10: *Filter tracking performance along the x coordinate axis*

over all the 15 postures. Note that the estimated values match with the ground truth within a margin of $\approx 4{-}5$ cm.

4.3 Pre-collision Strategy to achieve Safe HRC

The problem of ensuring safety based on separation monitoring is related to the traditional robot collision avoidance problem. However, the properties of physical interaction scenarios in shared work cells significantly differ from classical settings. For example, safety cannot be guaranteed always, if the robot responds to a detected imminent collision by using movements along alternative paths. This is mainly due to the inherently random nature of human motion, which is difficult to predict, and

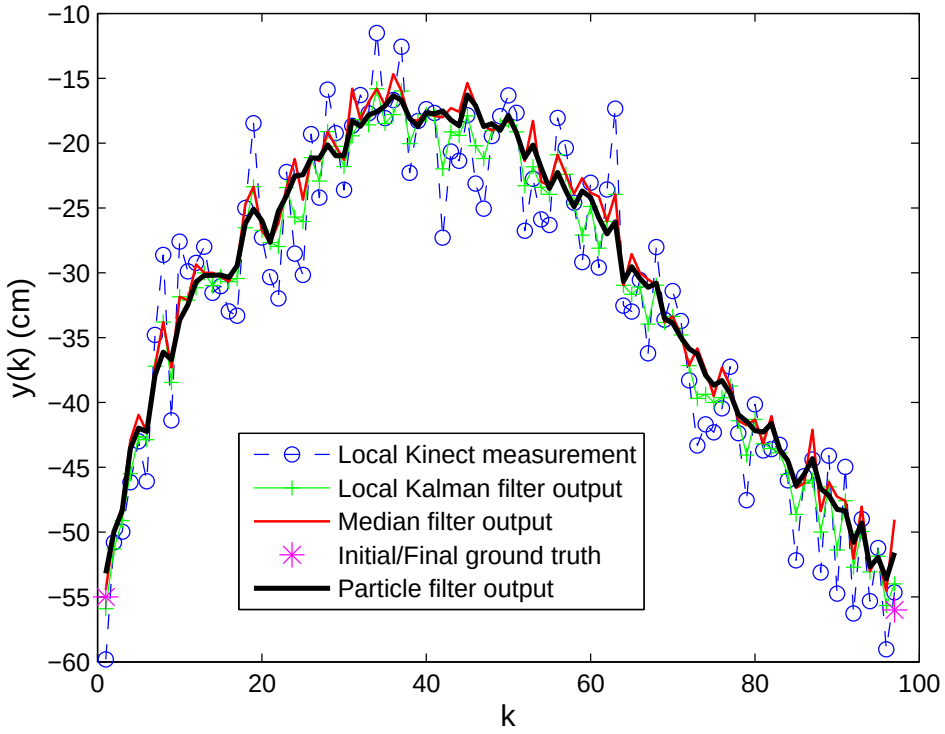


Figure 4.11: *Filter tracking performance along the y coordinate axis*

the dynamic nature of the robot implementing such a collision avoidance strategy. In addition, these methods may increase the computational overhead as the system must try to find collision-free paths in real-time. Velocity-scaling based methods [171] address these issues by operating the robot in a tri-modal state. In particular, the robot operates in a *clear* (normal functioning) state when the human is far away from it. When the separation between them is less than a specified threshold, the robot transitions into a *slow* state, in which it continues to move in the same path, but at a reduced speed. When the separation is less than a second threshold (whose value is smaller than that of the first one), the robot enters a *pause* state, in which it comes to a safe, controlled stop.

Our approach to ensuring safety while a human and robot collaborate in close

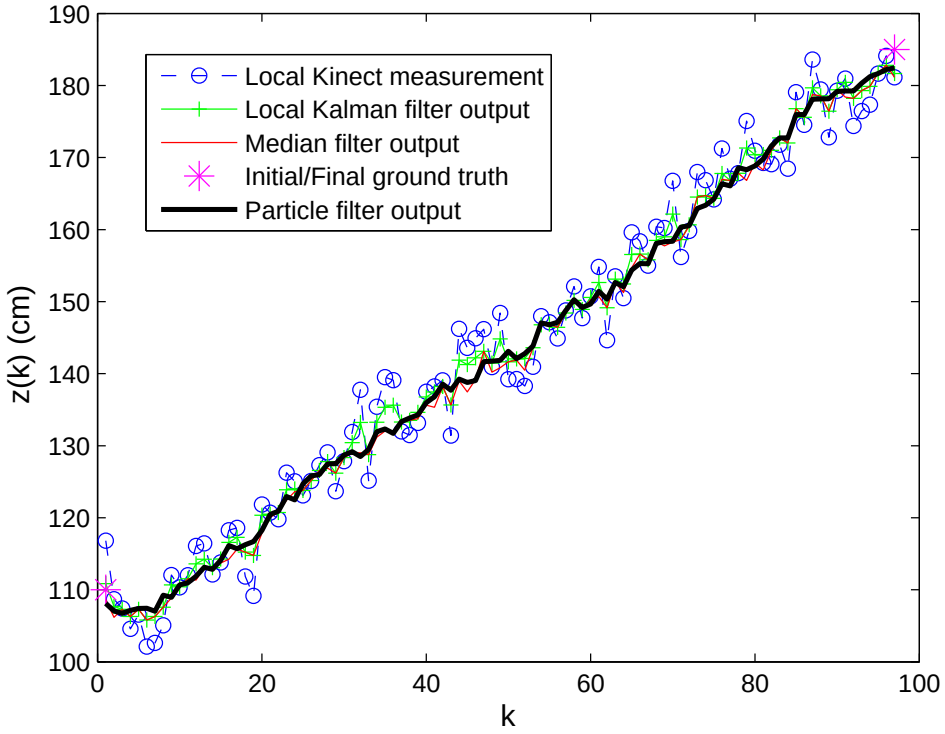


Figure 4.12: *Filter tracking performance along the z coordinate axis*

proximity with each other consists of pausing the robot's motion whenever an imminent collision between them is detected by the system. This is similar to a simpler bi-modal control strategy, in which the robot directly transitions from *clear* to *pause* when the estimated separation is below a threshold distance. This *stop-go* approach to safety is in line with the recommendations put forward by the ISO standard 10218 [172, 173].

In order to track the physical separation, the 20-joint human model generated by the exteroceptive sensing system (described in the previous section) is augmented by approximating all pairs of neighboring joints by dynamic bounding spheres that move in a 3D space as a function of the movements performed by the human in real-time. Now, we use a *roll-out* strategy, in which we pre-compute the robot's

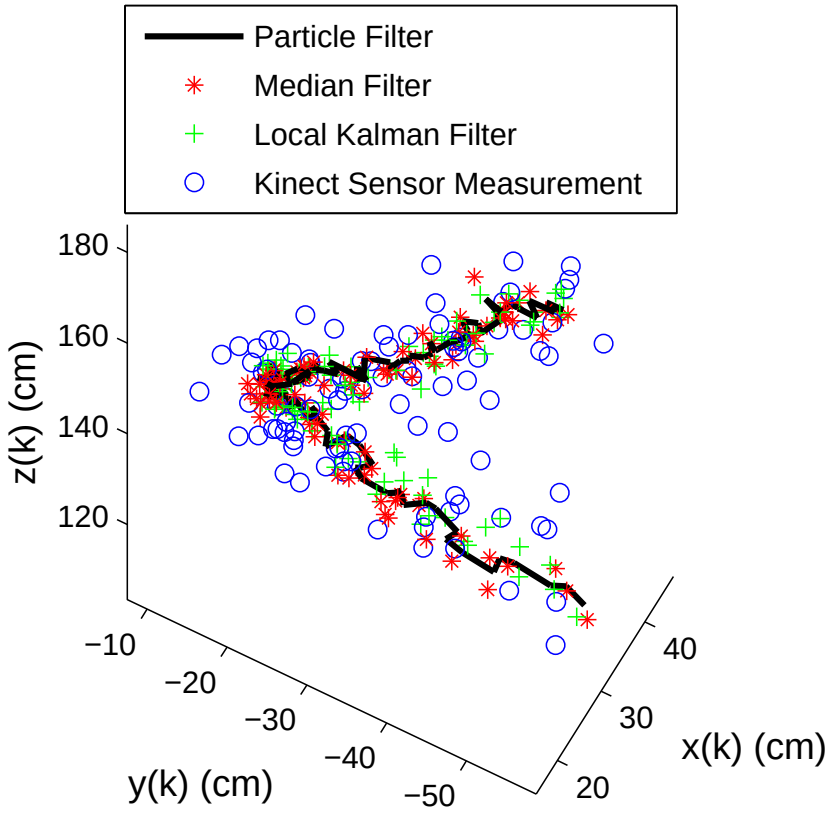


Figure 4.13: *Filter tracking performance in 3D*

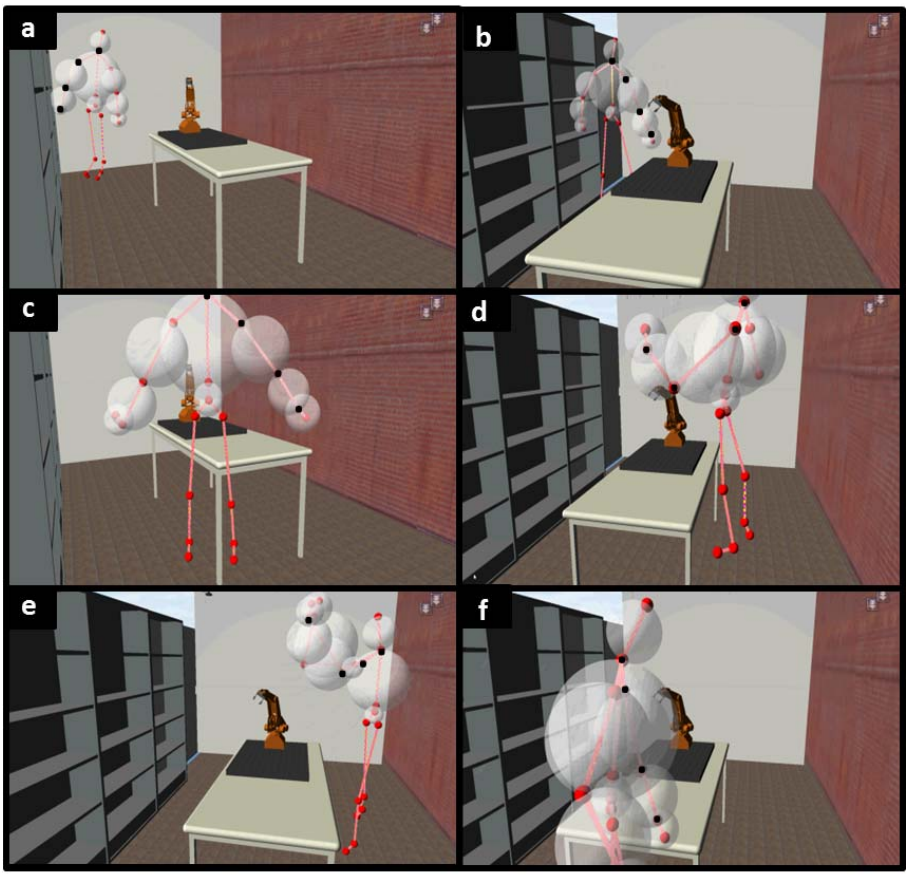


Figure 4.14: *Postures used to test the estimation accuracy of the overall system*

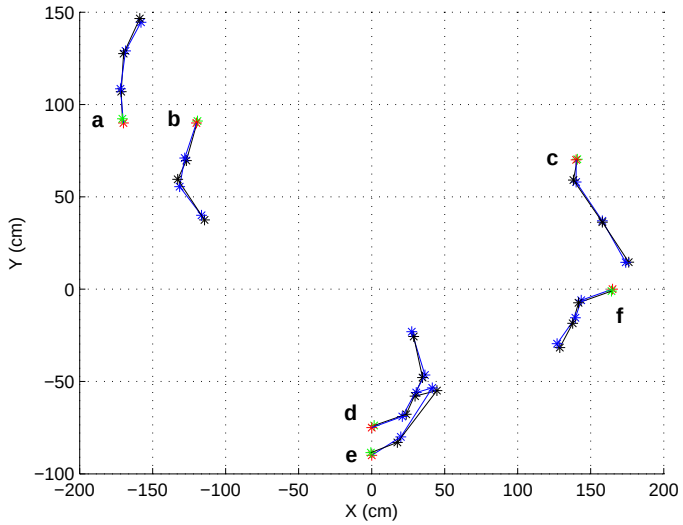


Figure 4.15: *Discrepancy between projections of ground truth and estimated values on the XY plane*

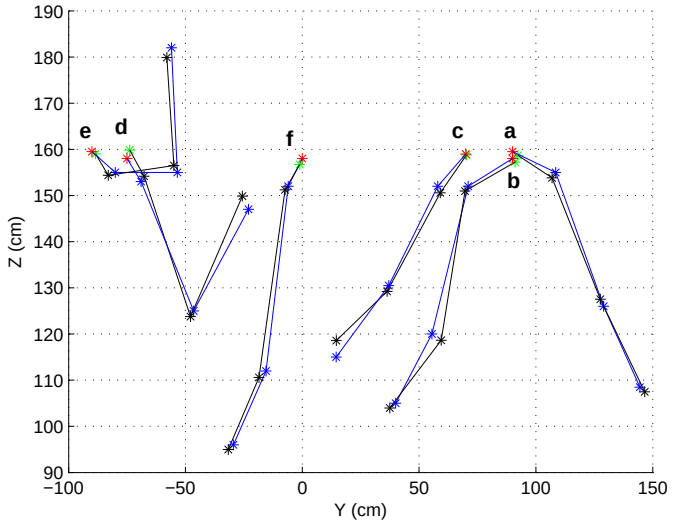


Figure 4.16: *Discrepancy between projections of ground truth and estimated values on the YZ plane*

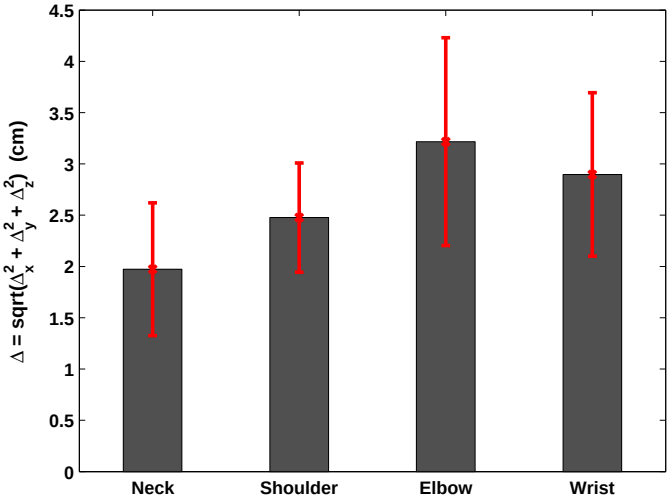


Figure 4.17: *Discrepancy between ground truth and estimated values for each joint averaged over 15 locations.*

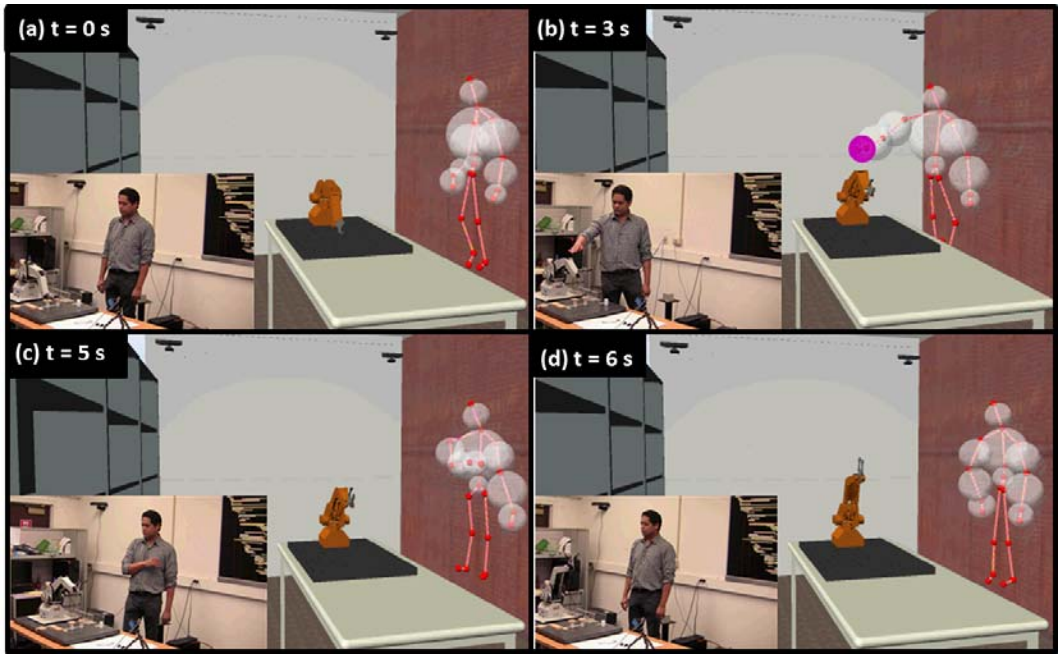


Figure 4.18: *Illustration of pre-collision strategy: (a) Human is far away from the robot. As the distance between the spheres is significant, robot performs its intended task. (b) An imminent collision is detected by the system; therefore, the robot is paused and a visual alarm is raised (bounding spheres change color). (c, d) Human returns to a safety zone; therefore, the robot resumes its motion.*

trajectory into the near future in order to create a temporal set of robot's postures for the next few seconds and check whether anyone of the postures in this set collides with one of the bounding spheres of the human model. This pre-collision strategy is implemented in a virtual simulation engine that is developed based on Tundra software.

First, a simulated robot, with a configuration and dimensions that are identical to the physical robot, is instantiated within the virtual environment. The simulated robot replicates the motion of the physical robot in real-time by using the same motor commands that drive the physical robot. The robot's motion plan is assumed to be known beforehand. Therefore. at time $t = 0$, we generate a set of 10 robot's postures by using this information, a sampling time of 0.3 sec, and a roll-out parameter of 3 sec. This set is updated at control-sampling frequency, according to a FIFO method, by removing the robot's current posture from the set and adding its future posture after 3 sec to the set.

Second, a simulated human model, with degrees of freedom identical to the one given by the Kinect, is built and instantiated within the same virtual environment. The simulated human model replicates the motion of the refined human model generated by the exteroceptive system by accessing the instantaneous positions of all the 20 joints. Since the joints below the hip do not interfere with the robot during any part of the interaction, they are not considered in the computation of the bounding spheres for the human model.

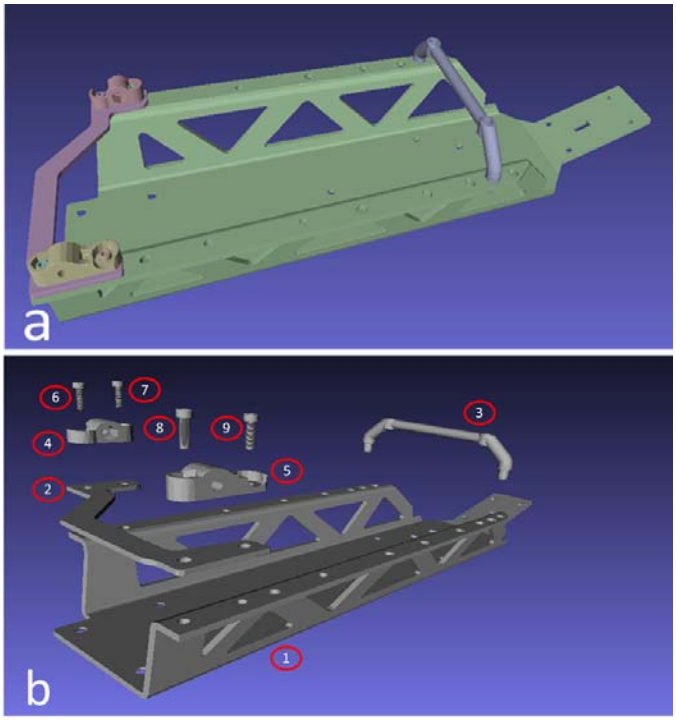
Figure 4.18 illustrates the pre-collision strategy based on the movement of the bounding spheres. From Fig. 4.18(a), the human is in front of the robot when it has

just started lifting a part at $t = 0$ sec. As there is no intersection between its current set of roll-out postures and the human model, the robot continues its intended task of lifting the part from the table surface. However, at $t = 3$ sec (Fig. 4.18(b)), note that the human's hand reaches a state in which a collision is imminent. The roll-out strategy enables the system to detect this condition and pause the robot's motion immediately. It also raises a visual alarm (the sphere changes color from white to red as seen in the figure), which is displayed on a monitor and an audio alarm to alert the human. After $t = 5$ sec (Fig. 4.18(c) and 4.18(d)), the robot automatically resumes its task as the human's hand is retrieved into a safety zone.

4.4 Results

We report results from an experimental scenario, in which a real robot and a human perform a shared assembly task. The physical robot used for the experiments is Lab-Volt 5150 5 DOF manipulator. The task consists of assembling the parts of a simplified chassis assembly consisting of the following parts: Main chassis, a center roll bar, a rear brace, two radio boxes, and four screws. An assembly planning system developed in our earlier work [24] takes a 3D CAD model of the assembly and automatically generates an assembly sequence that drives the task sequence of the robot.

We assign the roles of the human and the robot as follows: Whereas the human picks each part to be assembled and places it in front of the robot, the robot attempts to pick a part available in front of it and proceeds to place it in its intended



Linear Assembly Sequence

1	2	3	4	5	6	7	8	9
Main_ chassis	Rear_ brace	cenroll barset	4_radio	8_radio	SHCS 4_1	SHCS 4_2	SHCS 8_1	SHCS 8_2

Figure 4.19: *(a) CAD model of the simple chassis assembly used in the experiments. (b) Assembly sequence generated by the assembly planner.*

location in the assembly. The robot motion is kept asynchronous with respect to that of the human on purpose. That is, the robot doesn't wait to reach the part until the human finishes placing it in front of the robot. This thereby sets up an interaction scenario for possible collisions between the human and robot. Figure 4.20 shows how the robot and human collaborate to assemble one of the parts onto the main chassis. From Fig. 4.20(c), note that the robot pauses its motion when human intervenes to place the part in front of it and resumes its motion when the human turns away Fig. 4.20(f). Similar real-time behavior is observed as the robot and the human collaborate to assemble the remaining parts.

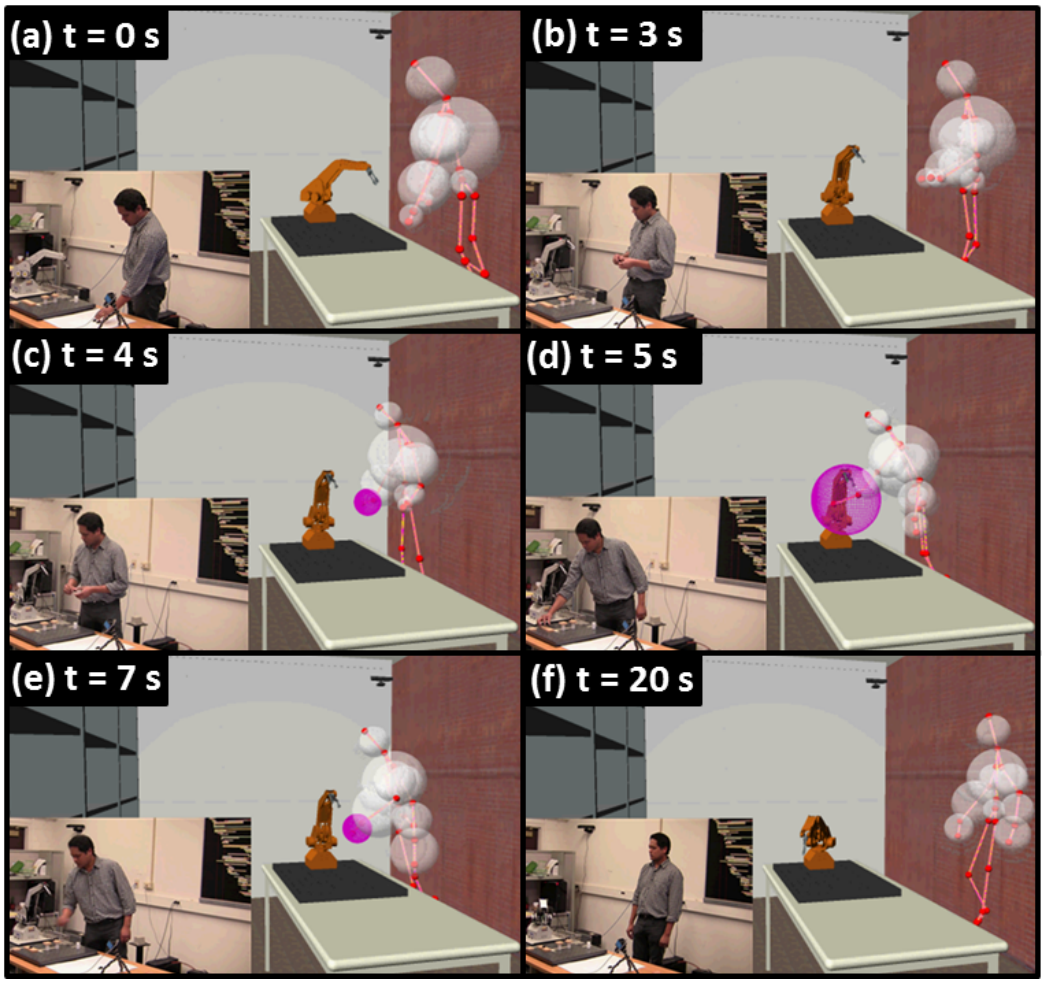


Figure 4.20: *Robot and human collaborate to assemble the third part (radio box) onto the chassis*

4.5 Summary

We presented a separation monitoring framework that allows a robot and human to safely collaborate on shared tasks in assembly cells. The main contributions of this chapter can be summarized as:

1. Design of an $N-$Kinect framework to generate a 3D model of human's movements in real-time.

2. Experimental procedure for placement of multiple Kinects in the work cell.

3. Technique to rapidly evaluate human-robot interference in 3D Euclidean space by using a physics-based simulation engine.

4. Pre-collision strategy to achieve safe HRC

In order to verify the accuracy of the framework, an error analysis was performed. The error analysis showed a maximum variation of +/-2 cm of each estimated joint with respect to the ground-truth. The pre-collision strategy uses these data to trigger the "stop" order to the robot if it detects the possibility of human-robot collision. The achieved level of accuracy was reasonable and offered several other advantages. Because no special suits, descriptors or equipment were required, human operators simply stepped into the assembly cell to begin the work. This means that the framework captured more realistic motion data in less time, and for a much lower total cost. It also made the interaction with the robot much more practical by reducing the use of additional hardware for the interaction. Because there is no special setup required, the Kinect framework shows flexibility to track

the motion of human operators with different heights or workers whose condition makes the application of special suits difficult.

In the current work, the pre-collision strategy consisted of bringing the robot to a complete stop whenever the system detected an intersection between the bounding spheres of the robot and the human. However, the human model based prediction of the human movement can be easily extended to derive better motion goals for the robot, which cater for safety as well as productivity. For example, a tri-modal control strategy, in which the robot transitions into an intermediate slow-speed state before coming to a complete stop can be easily implemented by incorporating the velocity estimates of the human model into the robot control algorithm. For this purpose, the current Taylor series based model can be extended to more practical dynamic models without the constant velocity assumption. Using real data obtained from extensive experiments, we demonstrated that our collaborative framework is robust and accurate. However, a more exhaustive evaluation of the accuracy of the tracking system can be made by comparing it with other tracking and motion capture systems.

Chapter 5

A Framework for Hybrid Cells that Support Safe and Efficient Human-Robot Collaboration in Assembly Operations

In this Chapter[1], we present a framework to build hybrid cells that support safe and efficient human-robot collaboration during assembly operations. Our approach considers a representative one-robot one-human model in which a human and a robot asynchronously work toward assembling a product. Whereas the human retrieves parts from a bin and brings them into the robot workspace, the robot picks up parts from its workspace and assembles them into the product. Using this collaboration model, we explicate the design details of the overall framework comprising three modules−plan generation, system state monitoring, and contingency handling. We provide details of the virtual cell and the physical cell used to implement our framework. Finally, we report results from human-robot collaboration experiments to illustrate our approach.

5.1 Introduction

Assembly operations are an integral part of the overall industrial manufacturing process. After parts are manufactured, they must be assembled together to impart the desired functionality to products. Pick-and-place, fastening, riveting, welding,

[1] The work in this chapter is partially derived from the published work [26, 27, 29] and [28].

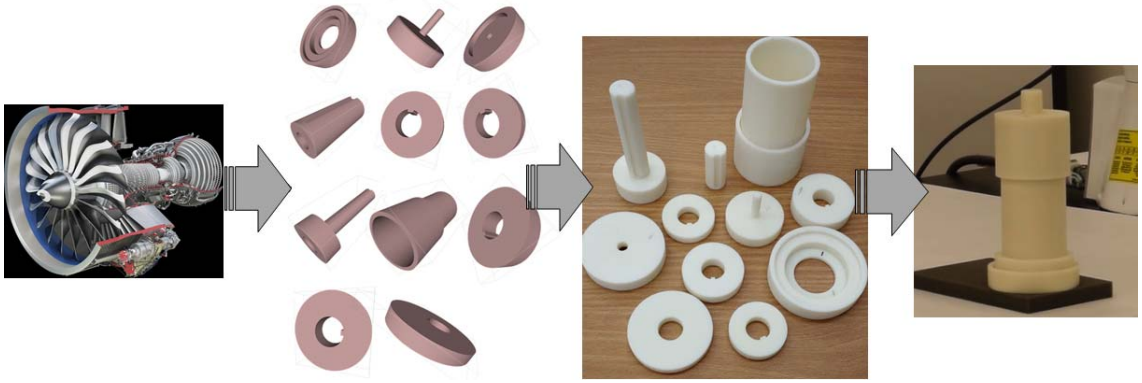


Figure 5.1: *3D printed replica of a jet engine*

soldering, brazing, adhesive bonding, and snap fitting constitute representative examples of industrial assembly tasks [18].

Humans and robots share complementary strengths in performing assembly tasks (Fig. 5.4). Humans offer the capabilities of versatility, dexterity, performing in-process inspection, handling contingencies, and recovering from errors. However, they have limitations w.r.t. factors of consistency, labor cost, payload size/weight, and operational speed. In contrast, robots can perform tasks at high speeds, while maintaining precision and repeatability, operate for long periods of times, and can handle high payloads. However, currently robots have the limitations of high capital cost, long programming times, and limited dexterity. Owing to these reasons, small batch and custom production operations predominantly use manual assembly. However, in mass production lines, robots are often utilized to overcome the limitations of human workers.

In contrast, robots are very good at pick and place operations and highly repeatable in placement tasks. Robots can perform tasks at high speeds and still maintain precision in their operations. Robots can also operate for long periods of

Figure 5.2: *Simplified 3D printed replica of a jet engine from Fig. 5.1*

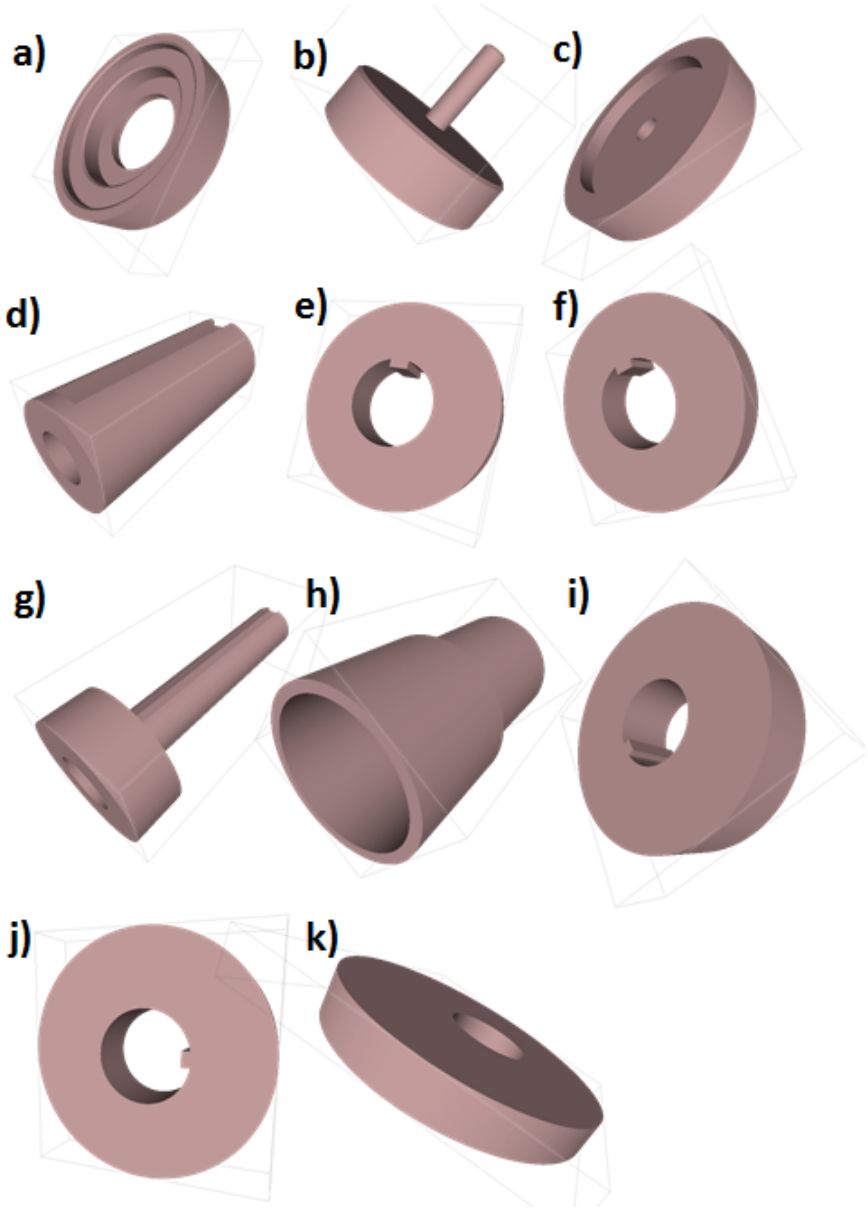


Figure 5.3: *Simplified 3D printed jet engine replica: (a) Front Shroud Safety (b) Main Fan (c) Shroud (d) Front Shaft (e) First Compressor (f) Second Compressor (g) Rear Shaft (h) Shell (i) Rear Bearing (j) Exhaust Turbine (k) Cover.*

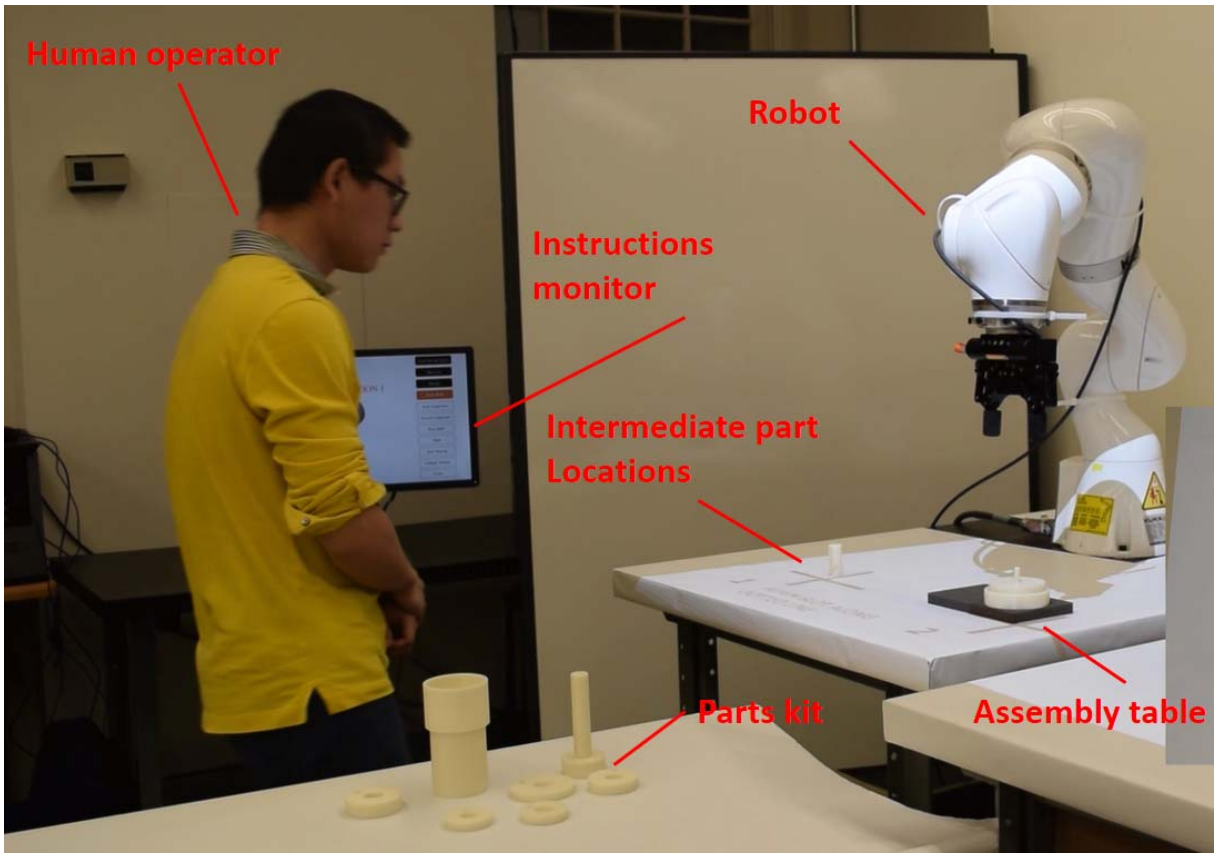


Figure 5.4: *Hybrid assembly cell: Human assisting the robot in resolving a pose estimation problem during an assembly task. Human pick a part and place the part in a known pose and location. Robot recognize the location and pick up the part and perform the task*

times. Robots are also very good at applying high forces and torques. They have become popular for several different assembly operations.

Purely robotic cells are not the solution as they do not provide the needed flexibility. These reasons, along with short production cycles and customized product demands, set SMMs as primary candidates to benefit from hybrid cells that support human-robot collaborations. However, currently shop floors install robots in cages. During robot operation, the cage door is locked and elaborate safety protocol is followed in order to ensure that no human is present in the cage. This makes it very difficult to design assembly cells where humans and robots can collaborate effectively.

In this chapter, we present a framework for hybrid cells that enable safe and efficient human-robot collaboration (HRC) during industrial assembly tasks. Advent of safer industrial robots [20, 21, 22] and exteroceptive safety systems [26, 25] in the recent years are creating a potential for hybrid cells where humans and robots can work side-by-side, without being separated from each other by physical cages. The main idea behind hybrid cells is to decompose assembly operations into tasks such that humans and robots can collaborate by performing tasks that are suitable for them. In fact, task decomposition between the human and robot (who does what?) has been identified as one of the four major problems in the field of human robot collaboration [174].

We consider a representative one-robot one-human model in which a human and a robot asynchronously work toward assembling a product. The model exploits complimentary strengths of either agents: Whereas the robot performs a pick-and-

place task and subsequently assembles each picked-up part to form the product, the human assists the robot in critical situations by performing dexterous fine manipulation tasks required during part-placing. Moreover, a system state monitoring allow the Hybrid Assembly cell to have a "knowledge" about the development of the assembly tasks, and provide additional information to the human operator if needed.

Whereas robots can repetitiously perform routine pick-and-place operations without any fatigue, humans excel at their perception and prediction capabilities in unstructured environments. Their sensory and mental-rehearsal capabilities enable humans to respond to unexpected situations, quite often with very little information. We exploit these complementary strengths of either agents in order to design a deficit-compensation model that overcomes the primary perception and decision-making problems associated with a sequence of assembly tasks. An overview of the hybrid cell is shown in Fig. 5.4.

We explicate the design details of our overall framework comprising three modules: plan generation, system state monitoring, and contingency handling. In order to prove our approach and because grasping techniques are out of this dissertation scope, we used a simplified 3D printed jet engine as a mechical assembly (Fig. 5.2 and Fig. 5.3).

5.2 System Overview

Our approach to hybrid cells considers a representative *one-human one-robot* model, in which a human and a robot will collaborate to assemble a product. In particular, the cell will operate in the following manner:

1. The cell planner will generate a plan that will provide instructions for the human and the robot in the cell.

2. Instructions for the human operator will be displayed on a screen in the assembly cell.

3. The human will be responsible for retrieving parts from bins or random locations and bringing them within the robot workspace.

4. The robot will pick up parts from its workspace and assemble them into the product.

5. If needed, the human will perform the dexterous fine manipulation to secure the part in place in the product.

6. The human and robot operations will be asynchronous.

7. The cell will be able to track the human, the locations of parts, and the robot at all time.

8. If the human operator makes a mistake in following an assembly instruction, re-planning will be performed to recover from that mistake. As a part of the

re-planning process, appropriate warnings and error messages will be displayed in the cell.

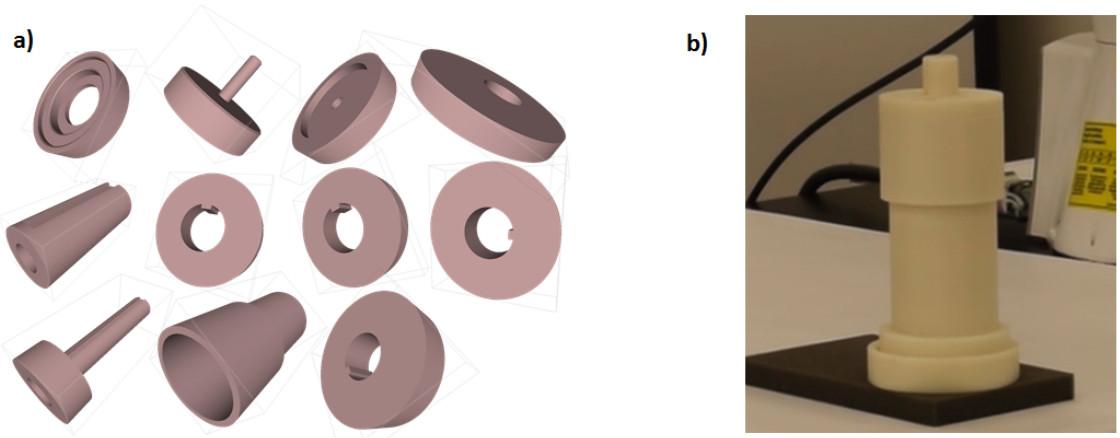
9. If the human comes too close to the robot to cause a collision, the robot will perform a collision avoidance strategy.

The overall framework used to achieve the above list of hybrid cell operations consists of the following three modules:

Plan generation. In Section 5.3, we present methods for automatically generating plans for the operation of hybrid cells. This will address both assembly complexity and issues related to motion constraints.

System state monitoring. In order to ensure smooth and error-free operation of the cell, we will need to monitor the state of the assembly operations in the cell. Accordingly, we present methods for real-time tracking of the human operator, the parts, and the robot in Section 5.4.

Contingency handling. We consider three types of contingency handling – *collision avoidance between robot and human*, *replanning*, and *warning generation*. A critical issue that is hampering the entry of humans into traditional robotic environments is safety. The cooperation between humans and robots in the assembly cell will only be practical if human safety can be ensured during the assembly tasks that require collaboration between humans and robots. Accordingly, in Section 5.5.1, we describe how the state information wr.t. the human and the robot obtained in the previous module is used to take appropriate measures to ensure human safety when the planned move by the robot may compromise the safety of



c)	Linear Assembly Sequence									
1	2	3	4	5	6	7	8	9	10	11
Front Shroud Safety	Main Fan	Shroud	Front Shaft	First Compressor	Second Compressor	Rear Shaft	Shell	Rear Bearing	Exhaust Turbine	Cover

Figure 5.5: *(a) Assembly CAD parts from a simplified Jet engine. (b)A simple jet engine assembly. (c) Feasible assembly sequence generated by the algorithm.*

the human. In the envisioned hybrid cell, we will be relying on human operators to bring the part into the cell. If the human operator makes an error in selecting the part or placing it correctly, the robot will be unable to correctly perform the task assigned to it. If the error goes undetected, it can lead to a defective product and inefficiencies in the cell operation. The reason for human error can be either confusion due to poor quality instructions or human operator not paying adequate attention to the instructions. Accordingly, in Section 5.5.2, we describe how the part tracking information obtained in the previous module is used to automatically generate instructions for taking corrective actions if a human operator deviates from the selected plan. Potential corrective actions may involve re-planning if it is possible to continue assembly from the current state. Corrective actions may also involve issuing warning and generating instructions to undo the current task.

5.3 Plan Generation

5.3.1 Assembly Sequence Generation

Careful planning is required to assemble complex products ([30, 34, 35, 50, 54, 81, 82, 55, 58, 50, 43]). Variations in part shapes and part sizes restrict the number of sequences in which assembly operations can be performed. Therefore, we must find precedence constraints among assembly operations and use them to guide the generation of feasible assembly sequences. In order to address this problem, we utilize a method developed and described in *Chapter* 3 that automatically detects *part interaction clusters* that reveal the hierarchical structure in a product. This thereby allows the assembly sequencing problem to be applied to part sets at multiple levels of hierarchy.

The assembly model used to illustrate the concepts developed in this chaper is a simple chassis assembly as shown in Fig. 5.5(a)-(b). The result of applying the above method on this assembly model is a feasible assembly sequence as shown in Fig. 5.5(c).

5.3.2 Instruction Generation

The human worker inside the hybrid cell follows a list of instructions to perform assembly operations. However, poor instructions lead to the human committing mistakes related to the assembly. We address this issue by utilizing an instruction generation system developed in our previous work [27] that creates effective and easy-to-follow assembly instructions for humans.

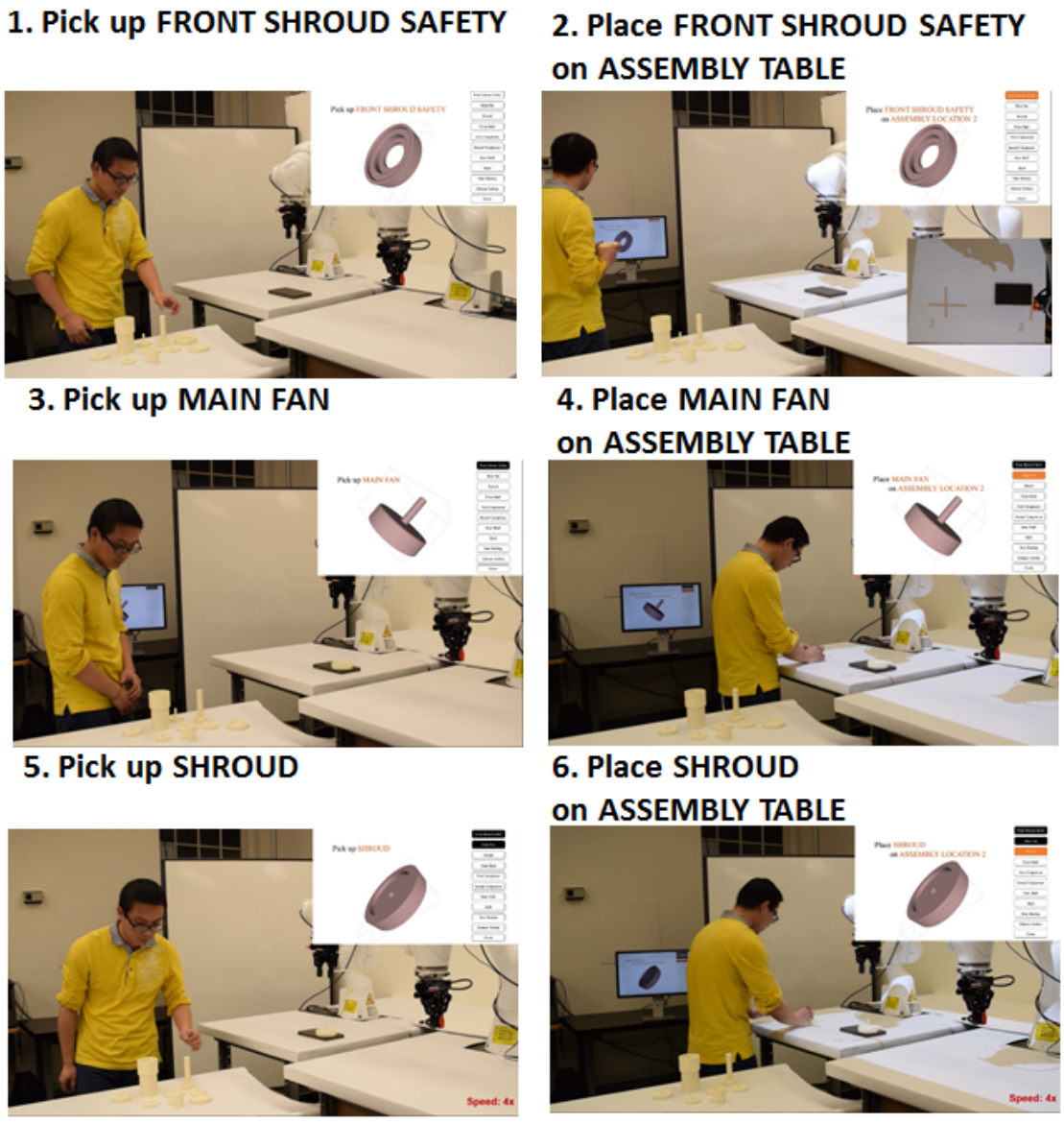


Figure 5.6: *Generation of instructions for chassis assembly (1-6)*

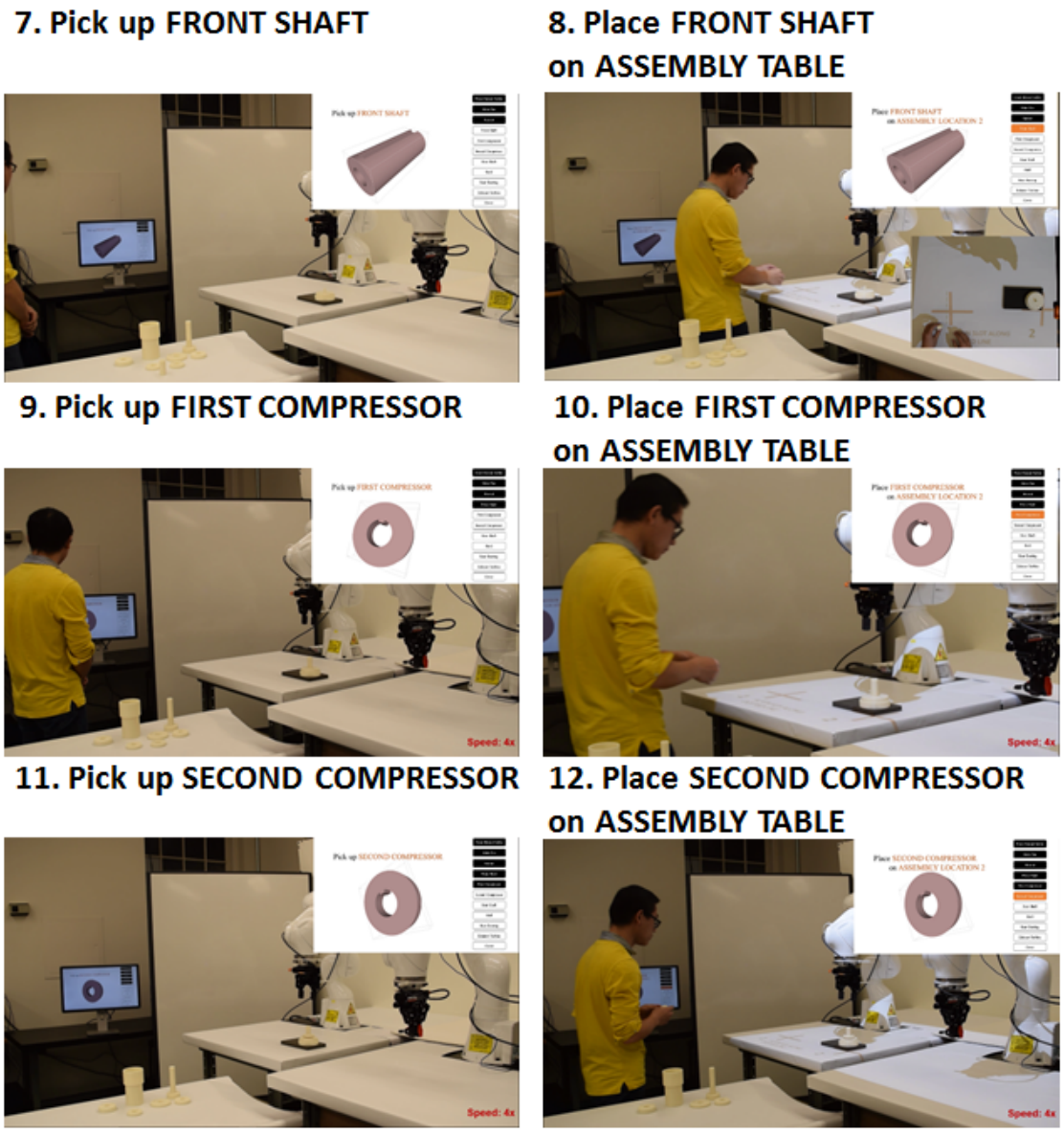


Figure 5.7: *Generation of instructions for chassis assembly (7-12)*

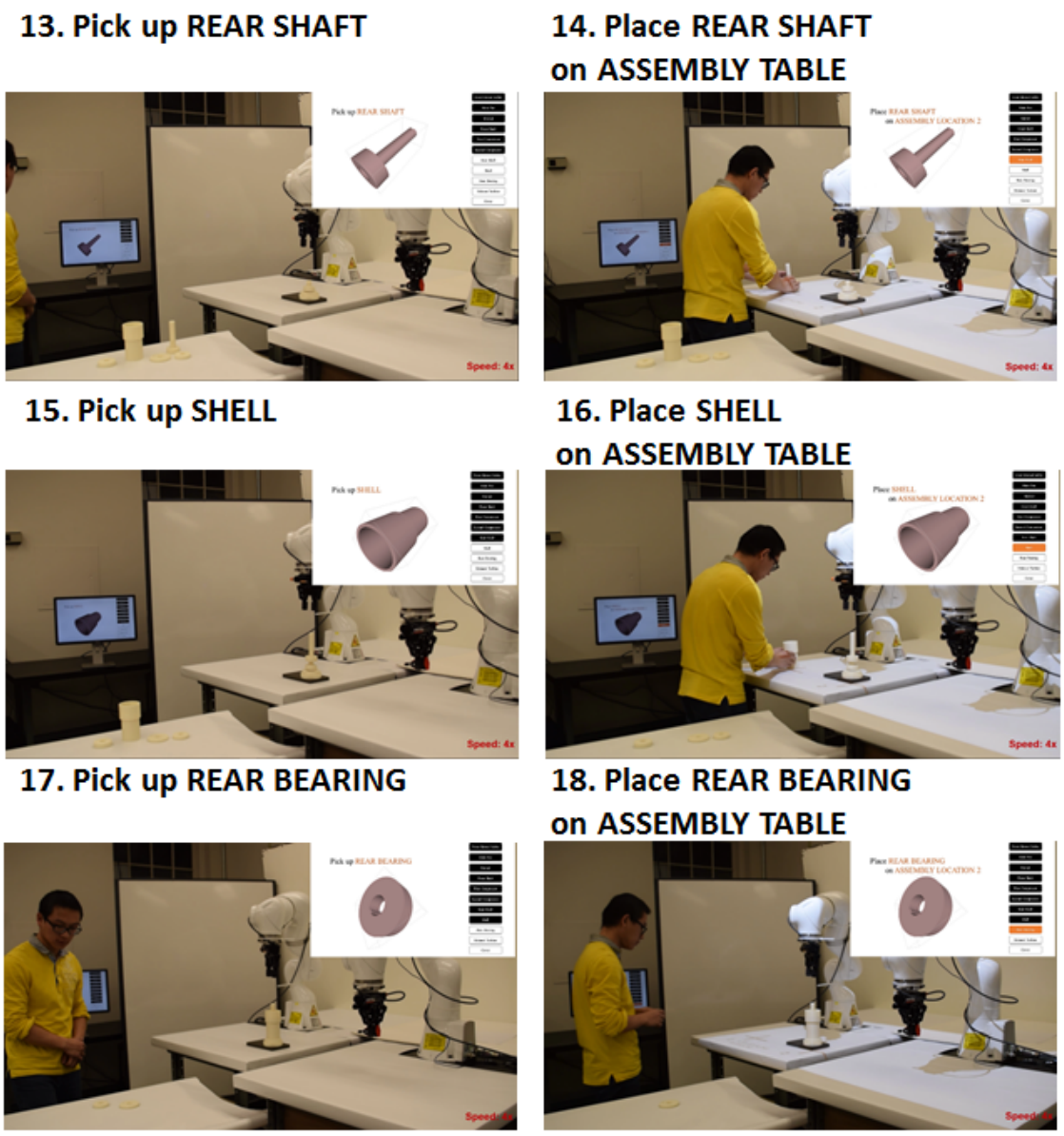


Figure 5.8: *Generation of instructions for chassis assembly (13-18)*

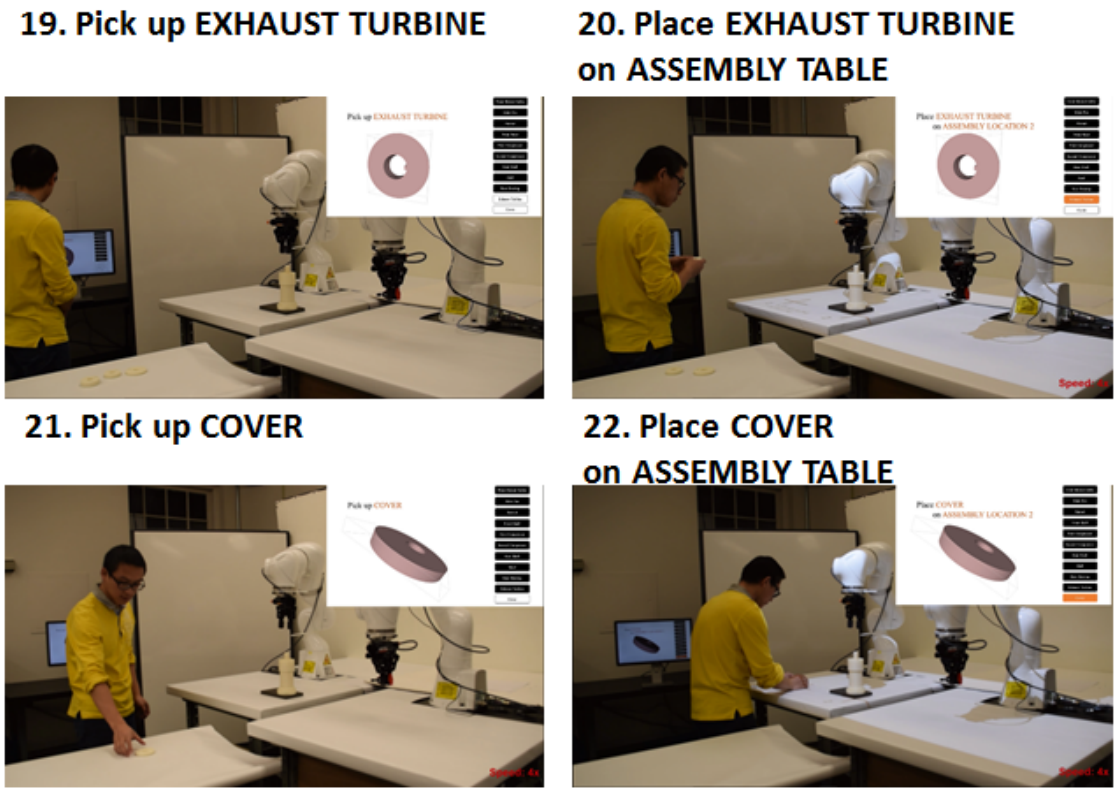


Figure 5.9: *Generation of instructions for chassis assembly (19-22)*

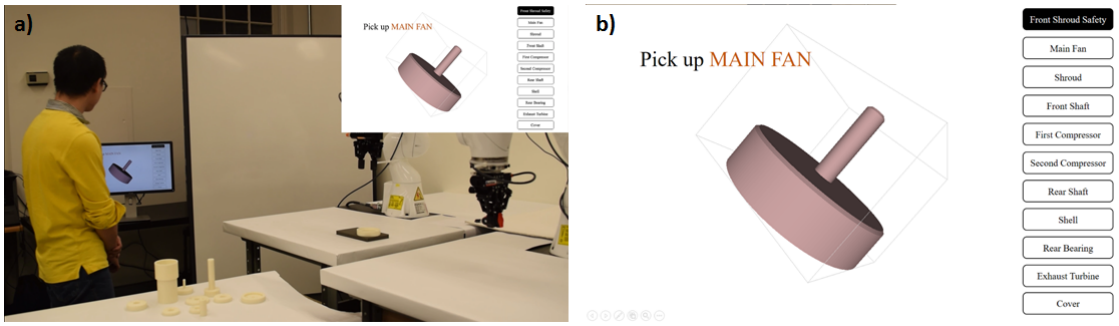


Figure 5.10: *(a) Human operator viewing an Assembly Instruction. (b) Human Implementing the Viewed Instruction.*

A linearly ordered assembly sequence (result of the previous section) is given as input to the system. The output is a set of multimodal instructions (text, graphical annotations, and 3D animations). Instructions are displayed on a big monitor located at a suitable distance from the human. Text instructions are composed using simple verbs such as *Pick, Place, Position, Attach,* etc. Examples of grammatical constructs for the text instructions include:

1. **Pick** PART?
2. **Place** PART? **on** LOCATION?
3. **Position** PART? **so that** FEATURE-A? **aligns with** FEATURE-B?

As mentioned in Section 5.3.1, we compute a feasible assembly sequence directly from the given 3D CAD model of the chassis assembly. Therefore, the following assembly sequence is input to the instruction generation system:

1. Pick up FRONT SHROUD SAFETY
2. Place FRONT SHROUD SAFETY on ASSEMBLY TABLE
3. Pick up MAIN FAN
4. Place MAIN FAN on ASSEMBLY TABLE
5. Pick up SHROUD
6. Place SHROUD on ASSEMBLY TABLE
7. Pick up FRONT SHAFT

8. Place FRONT SHAFT on ASSEMBLY TABLE
9. Pick up FIRST COMPRESSOR
10. Place FIRST COMPRESSOR on ASSEMBLY TABLE
11. Pick up SECOND COMPRESSOR
12. Place SECOND COMPRESSOR on ASSEMBLY TABLE
13. Pick up REAR SHAFT
14. Place REAR SHAFT on ASSEMBLY TABLE
15. Pick up SHELL
16. Place SHELL on ASSEMBLY TABLE
17. Pick up REAR BEARING
18. Place REAR BEARING on ASSEMBLY TABLE
19. Pick up EXHAUST TURBINE
20. Place EXHAUST TURBINE on ASSEMBLY TABLE
21. Pick up COVER
22. Place COVER on ASSEMBLY TABLE

Figure 5.6, Figure 5.7, Figure 5.8 and Figure 5.9 show the instructions used by the system for some of the assembly steps.

5.4 System State Monitoring

5.4.1 Human Tracking

The human tracking system used here is based on our N-Kinect based sensing framework presented in *Chapter* 4. The system is capable of building an explicit model of the human in near real-time. As it was mentioned previously, it is designed for a hybrid assembly cell where one human interacts with one robot in a 3D environment to perform assembly operations. In this design, the human has complete freedom of his/her motion. Human activity is captured by the Kinect sensors that reproduce the human's location and movements virtually in the form of a simplified animated skeleton.

5.4.2 Part Tracking

The assembly cell state monitoring uses a discrete state-to-state part monitoring system that was designed to be robust and decrease any possible robot motion errors. A failure in correctly recognizing the part and estimating its pose can lead to significant errors in the system. To ensure that such errors do not occur, the monitoring system is designed based on 3D mesh matching with two control points−the first control point detects the part selected by the human and the second control point detects the part's spatial transformation when it is placed in the robot's workspace. The detection of the selected part in the first control point helps the system to track the changes introduced by the human in real-time and trigger the assembly re-planning and the robot motion re-planning based on the new sequence. Moreover,

the detection of the posture of the assembly part related to the robot in the second control point sends a feedback to the robot with the "pick and place" or "wait" flag.

The 3D mesh matching algorithm uses a real-time 3D part registration and a 3D mesh interactive refinement [175]. In order to register the assembly part in 3D format, multiple acquisitions of the surface are necessary given that a single acquisition is not sufficient to describe the object. These views are obtained by the Kinect sensors and represented as dense point clouds. The point clouds are refined in real-time by a dense projective data association and a point-plane iterative closes point ICP all embedded in KinectFusion [176, 177, 178, 179]. KinecFusion is used to acquire refined point-clouds from both control points and for every single assembly part.

In order to perform a 3D mesh-to-mesh matching, an interactive refinement revises the transformations composed of scale, rotation, and translation. Such transformations are needed to minimize the distance between the refined point cloud in a time t_i and the refined point cloud at the origin t_0 also called mesh model. Point correspondences were extracted from both meshes using a variation of Procrustes Analysis [180, 181, 182] and then compared with an iterative closest point algorithm [183].

5.4.2.1 3D Mesh Matching Algorithm

3D vision measurements produce 3D coordinates of the relevant object or scene with respect to a local coordinate system. 3D Point cloud registration transform multiple

data sets into the same coordinate system to overlap and align the components of the point clouds. There is no an standard method for the registration problem and the performance of the algorithms are often related to preliminary assumptions.

Consider a point cloud representation of a rigid object with n points $x_1, ..., x_n$ in $\mathbb{R}^3$ that is subject to an orthogonal rotation $R \in \mathbb{R}^{3x3}$ and a translation $t \in \mathbb{R}^3$ then the problem is fitting the points $x_1, ..., x_n$ into a given point cloud representation of the same object or scene with m points $y_1, ..., y_m$ under choice of an unknown rotation R, an unknown translation t, and an unknown scale factor s.

We can represent several configurations of the same object in a common space by minimizing the goodness of fit criterion. We do this with the aid of 3 high-level transformations:

$Translation$: move the centroids of each configuration to a common origin.

$Isotropic\ scaling$: shrink or stretch each configuration isotropically to make them as similar as possible.

$Rotation/Reflection$: turn or flip the configurations in order to align the point clouds.

Given $x_1, ..., x_n$ in $\mathbb{R}^3$ and $y_1, ..., y_n$ in $\mathbb{R}^3$ and assuming that there are no translation and scale transformations then, the rotation R can be computed by solving:

$$\min_R \|RX - Y\|_F^2 \qquad subject\ to\ R^T R = I_3, det(R) = 1$$

Therefore, the solution of the orthogonal procrustes problem is given by using

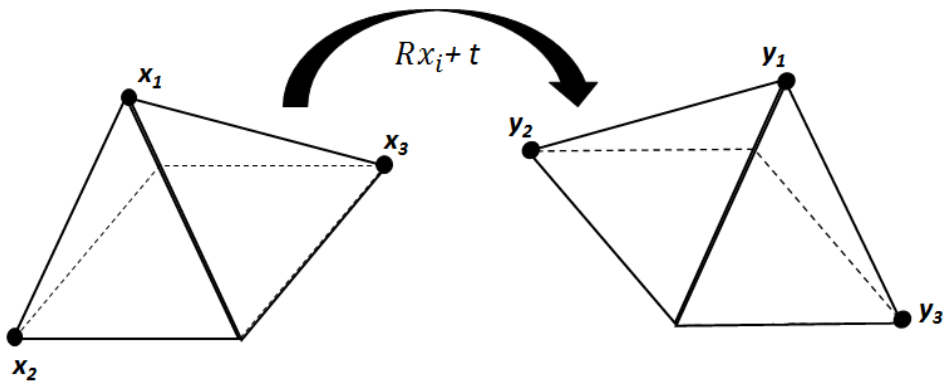


Figure 5.11: *Rigid body transformations (Rotation and Translation).*

SVD of XY^T. Where $X = (x_1 - \bar{x}, ..., x_n - \bar{x})$ and $Y = (y_1 - \bar{y}, ..., y_n - \bar{y})$ with the variables $\bar{x}$ and $\bar{y}$ representing the mean value vectors of x_i and y_i. If XY^T is nonsingular the solution is unique.

Theorem 4. *Let $X \in \mathbb{R}^{nxn}$ and $Y \in \mathbb{R}^{mxn}$ be known matrices with Rank(X) = n and Rank(Y) = n. Then the solution $\hat{R}$ of the orthogonal procrustes problem*

$$\min_R \|RX - Y\|_F^2 \qquad subject\ to\ R^T R = I_n, det(R) = 1$$

is $\hat{R} = V I_{m,n} U^T$ where U and V are the orthogonal matrices given by the singular value decomposition $U\Sigma V^T = XY^T$

Proof. Since

$$\|RX - Y\|_F^2 = tr((RX - Y)^T (RX - Y))$$

$$tr((RX-Y)^T(RX-Y)) = tr((RX)^T(RX))+tr((Y)^T(Y))-tr((RX-Y)^T Y)-tr(Y^T(RX))$$

$$= \|X\|_F^2 + \|Y\|_F^2 - 2tr(Y^T RX)$$

$$\|RX - Y\|_F^2 = \|X\|_F^2 + \|Y\|_F^2 - 2tr(Y^T RX)$$

then, the orthogonal procrustes problem is equivalent to

$$\max tr(Y^T RX), \qquad subject\ to\ R^T R = I_n$$

where $\mathrm{tr}(Y^T RX) = \mathrm{tr}(XY^T R)$.

Let $U\Sigma V^T = XY^T$ be a singular value decomposition. Then, we define the *consensus matrix* $Z \in \mathbb{R}^{mxn}$ where $Z = V^T RU$ and we obtain

$$tr(XY^T R) = tr(\Sigma V^T RU)$$

where $V^T RU$ are embedded in Z therefore,

$$= tr(\Sigma Z)$$
$$= \sum_{i=1}^{n} \sigma_i z_{i,i}$$

Because $V^T RU$ has orthonormal columns, the upper bound of $\max tr(Y^T RX)$ is given by having $V^T RU = I_{m,n}$ (identity consensus matrix $z_{i,i} = 1$). Therefore, the solution of the orthogonal procrustes problem is $R = V I_{m,n} U^T$ □

Procrustes Analysis removes the effects of level, range and interpretation from each individual point cloud by applying 3 transformations: translation to common mean, isotropic scaling (stretch or shrink) and rotation/reflection.

Then, the set of transformations of the rigid object can be represented by $sx_iR+jt^T = y_i$ where $j = 1$ is $1xn$ unit vector. The optimization problem of finding R, t, and s that minimized the fitting error is often called Extended Orthogonal Procrustes Analysis.

We cast our matching/registration problem as a Weighted Extended Orthogonal Procrustes Analysis (WEOPA).

$$\min_{sRt} \|sXR + jt^T - Y\|_F^2 \qquad subject\ to\ R^TR = I_3, det(R) = 1$$

where $\|.\|_F$ is the Frobenious matrix norm. By using matrix-trace we have the following formulation:

$$\min_{sRt} tr(sXR + jt^T - Y)^T(sXR + jt^T - Y) \qquad subject\ to\ R^TR = I_3, det(R) = 1$$

By introducing the Lagrangian function, we have

$$E = tr((sXR + jt^T - Y)^T(sXR + jt^T - Y)) + tr(L(R^TR - I_3))$$

$$\begin{aligned} E = tr(Y^TY) + s^2tr(R^TX^TXR) \ + j^Tjt^Tt - 2str(Y^TXR) \\ -2tr(Y^Tjt^T) + 2str(R^TX^Tjt^T) \ + tr(L(R^TR - I_3)) \end{aligned}$$

Finding the derivatives respect to the transformations:

$$\frac{\partial E}{\partial R} = 2s^2X^TXR - 2sX^TY + 2sX^Tjt^T + R(L + L^T)$$

$$\frac{\partial E}{\partial t} = 2j^Tjt - 2Y^Tj + 2sR^TX^Tj$$

$$\frac{\partial E}{\partial s} = 2str(R^TX^TXR) - 2tr(Y^TXR) + 2tr(R^TX^Tjt^T)$$

Assume that the point clouds are dominated by noise. Then by constructing a diagonal matrix W, we can give these point clouds a low weight as $(sXR+jt^T-Y)W$. W then can also be decomposed in two weights W_n and W_3.

$$\|sXR + jt^T - Y\|_F^2 = tr(sXR + jt^T - Y)^TW_n(sXR + jt^T - Y)W_3,$$

$$subject\ to\ R^TR = I_3, det(R) = 1$$

A direct solution can be computed if $W_3 = I$, otherwise there is not direct solution to the two view problem. An Iterative optimization algorithm is given in Algorithm 7 [184] to compute a solution to the above equation. Moreover, a WEOPA can have several minima, which leads to the problem of deciding if a computed solution is the "best" one. Hence a solution, computed by some iterative method, is not necessarily a global optimum.

Algorithm 7 Weighted Extended Orthogonal Procrustes Analysis Algorithm.

Input:

$X = \{x_1, x_2, \ldots, x_n\}$ (point cloud reference)
$Y = \{y_1, y_2, \ldots, y_m\}$

Output:

$R = \in \mathbb{R}^{3x3}$ (Rotation)
$t \in \mathbb{R}^3$; (translation)
$s = \in \mathbb{R}$; (scale)

Procedure WEOPA Fitting$(X, Y) \rightarrow (R, t, s)$

Compute initial transformation values R_0, T_0, s_0;
$k = 0$, $\Delta = 10^{-9}$, $\Delta_k = \Delta + 1$;
while $\Delta_k > \Delta$ **do**
  if $(J^T J + H)$ is positive definite **then**
    Compute a Newton search direction;
  else
    Compute a Gauss-Newton search direction;
  end if
  Update R, t and s
  $k = k + 1$
  Update Δ
end while
Return R, t, s

Where the Gauss-Newton search direction is given by

$$s_{GN} = -J^{+}(sXR + jt^T - Y)$$

the search direction corresponding to Newton method is

$$s_N = -(J^T J + H)^{-1}(J^T(sXR + jt^T - Y))$$

and the Δ update is given by

$$\Delta_{k+1} = \frac{\|J^T(sXR + jt^T - Y)\|_2}{\|J\|_2\|(sXR + jt^T - Y)\|_2}$$

The algorithm depends of a good R_0, t_0, s_0 initialization, therefore the al-

gorithm is not stable. In order to solve the stability problem, a heuristic method was designed [184]. We called the heuristic Iterative-WEOPA. The heuristic uses a random initialization of R_0, t_0, s_0, combined with the WEOPA fitting algorithm, to compute and store additional minimums. When no new minimum is found, after 150 random R_0, t_0, s_0 initialization have been used, the algorithm is terminated. Later the total number of minimums are used in order to draw conclusions. Moreover,

Algorithm 8 Weighted Extended Orthogonal Procrustes Analysis Algorithm.

Procedure I-WEOPA HEURISTIC Search of minima $(X, Y) \rightarrow (R, t, s)$
  Compute initial transformation values R, T, s;
  $k = 0$;
  while $k < number\ of\ iterations$ **do**
    R_0 = Random orthogonal matrix with $R^T R = I$ and $det(R) = 1$;
    t_0 = Random translation vector;
    s_0 = Random scale unit;
    $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$, $\hat{\mathbf{s}}$:= Computed minimum with R_0, t_0, s_0 as initial values for the WEOPA Fitting algorithm.
    if $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$, $\hat{\mathbf{s}}$ is a new minimum **then**
      Store $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$, $\hat{\mathbf{s}}$;
      $k = 0$;
    end if
    $k = k + 1$
  end while
  Return R, t, s

experimentation showed that in most of the cases the algorithm found the minimum in less than 35 initialization parameters.

5.4.2.2 Results

We have created a 3D printed jet engine replica. The jet engine is composed by eleven assembly parts but we have selected five representative parts(Fig. 5.12) that afford different recognition complexities to illustrate various challenges encountered during an assembly task.

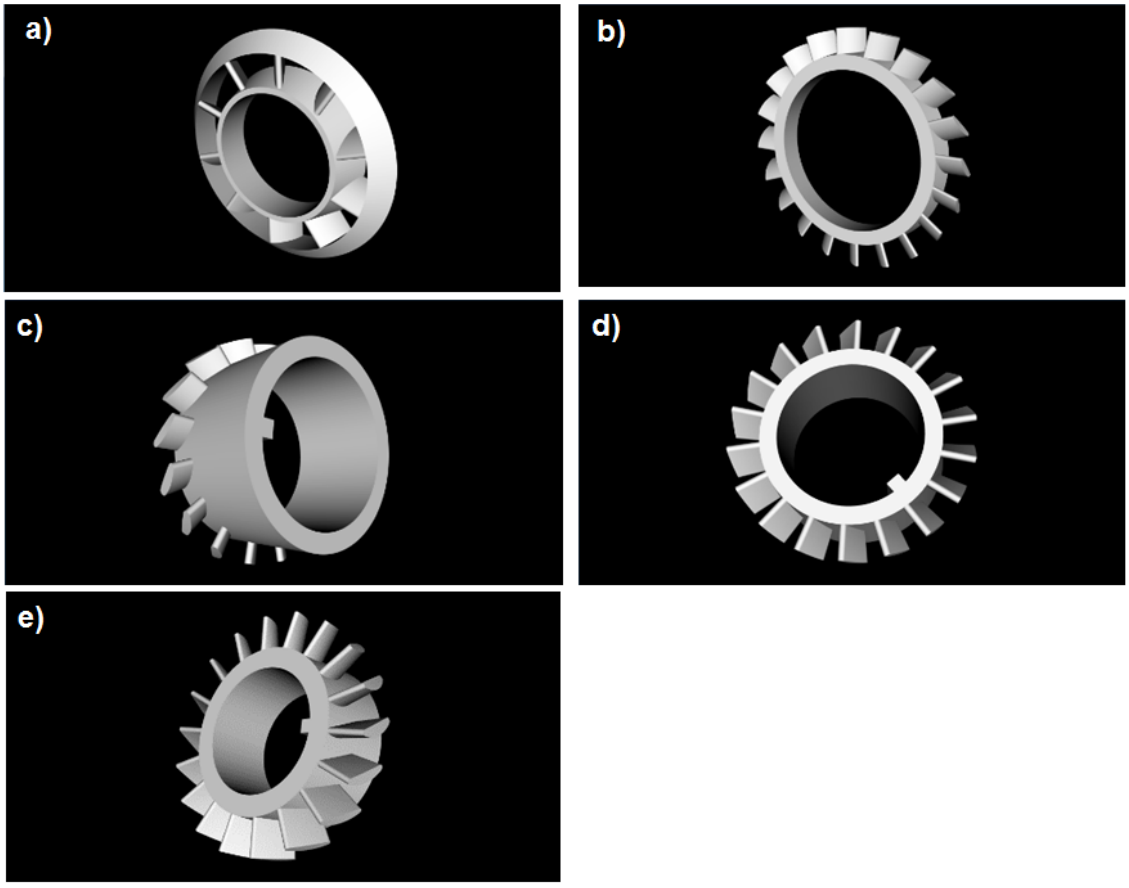


Figure 5.12: *3D printed jet engine replica with representative assembly parts affording different recognition complexities: (a) Rear Bearing (b) Exhaust Turbine (c) Third Compressor (d) Second Compressor (e) First Compressor.*

The scenario shown in Fig. 5.14 represents a case where the human operator is tasked with picking Part 1 through 5 in sequence. The parts are out of the robot workspace. A non-prerecorded initial configuration represents a complex task for the robot because of the difficulty of detecting random part poses. Therefore, human pick the parts one at the time following the assembly plan Fig. 5.14.

Because of this, and in order to track the changes in the scene, the first step is to perform segmentation on the point cloud in order to retrieve all assembly parts. In this case we performed a plane segmentation to find any table in the scene, and

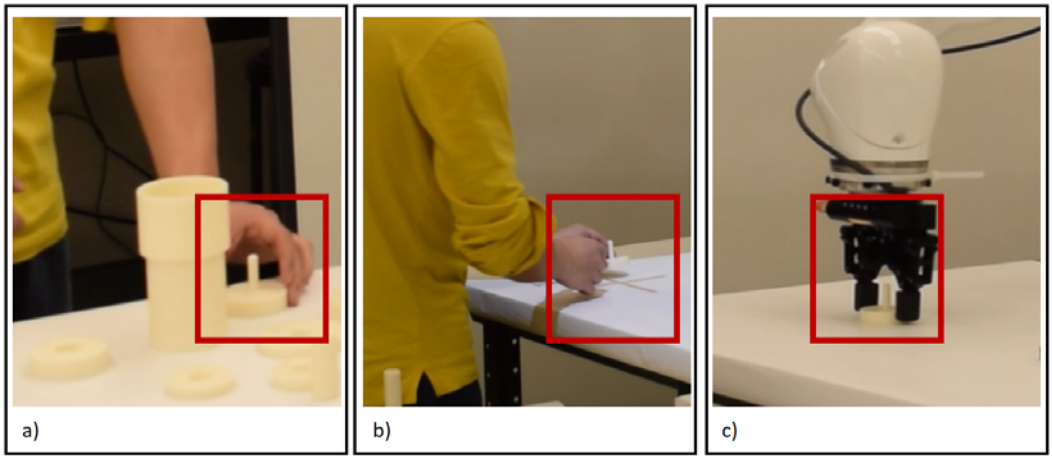
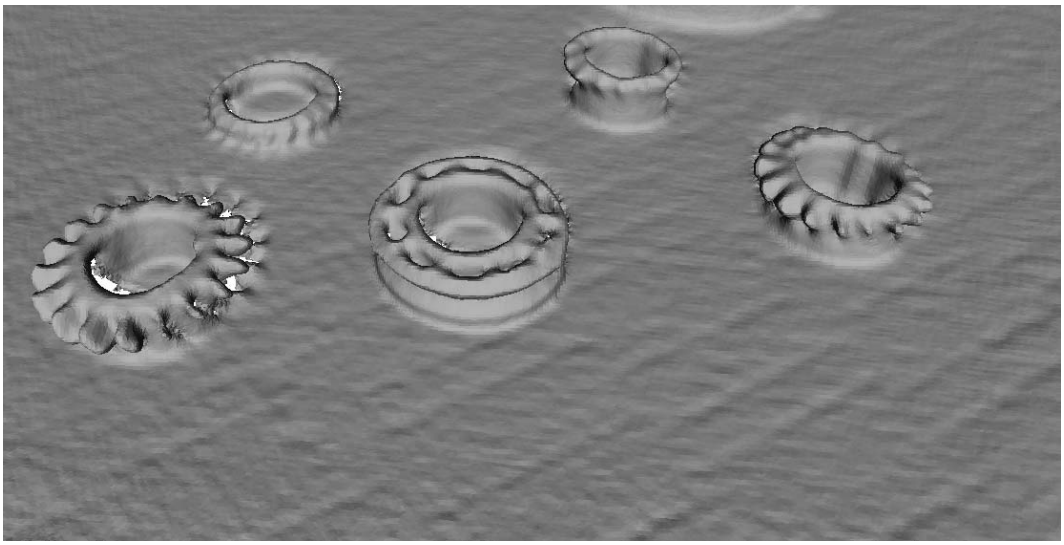


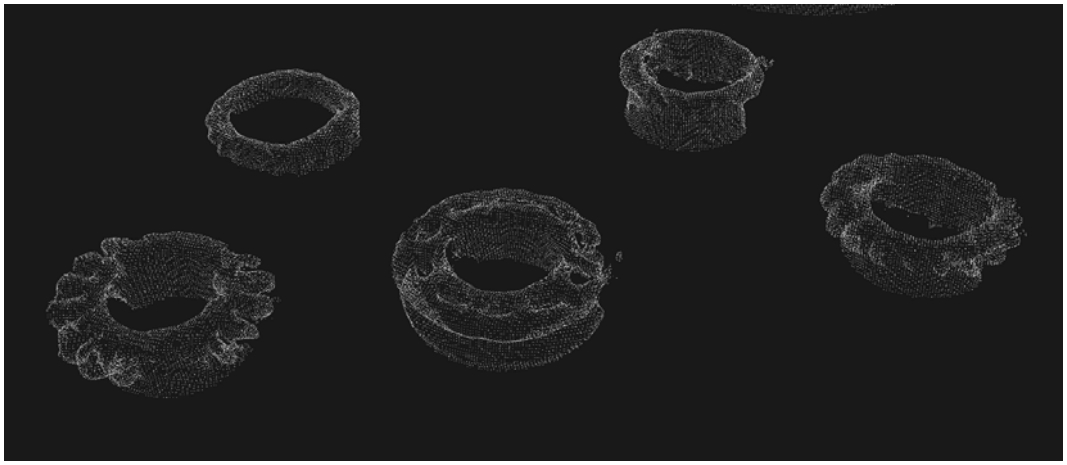
Figure 5.13: *The state-state discrete monitoring system has two control points: (a) Initial location: Parts are located out of the robot workspace in a random configuration. Human pic the parts one by one. (b) Intermediate location: Human place the parts at the robot workspace in an specific configuration. (c) Robot successfully picking up the part from the assembly table and perform the task.*

consider only clusters sitting on it. Later, we removed all clusters that are too small or too big in order to reduce the number of cluster and therefore the noise in the scene.

After human places the part, the part is now ready to be picked by the robot. Uncertainties related to pose estimation are reduced to a small variations in the final location. That is, any attempt by the robot to pick up the part results in a successful grasping (Fig. 5.13(c)).

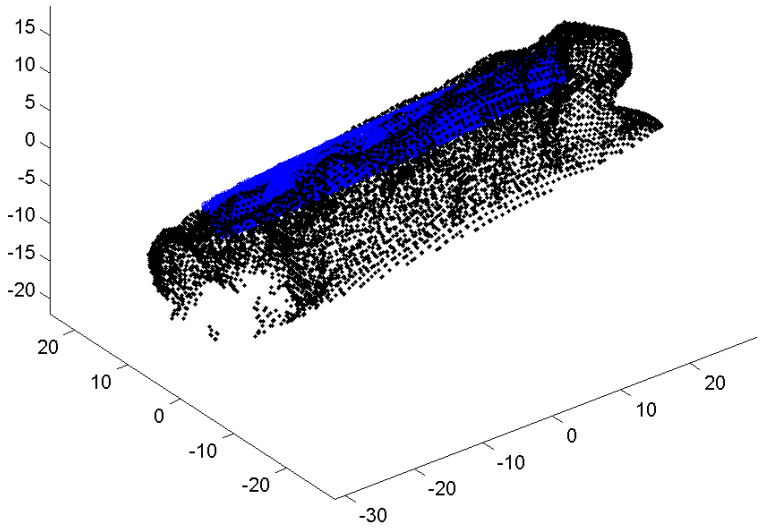


(a) *Scene scanned*

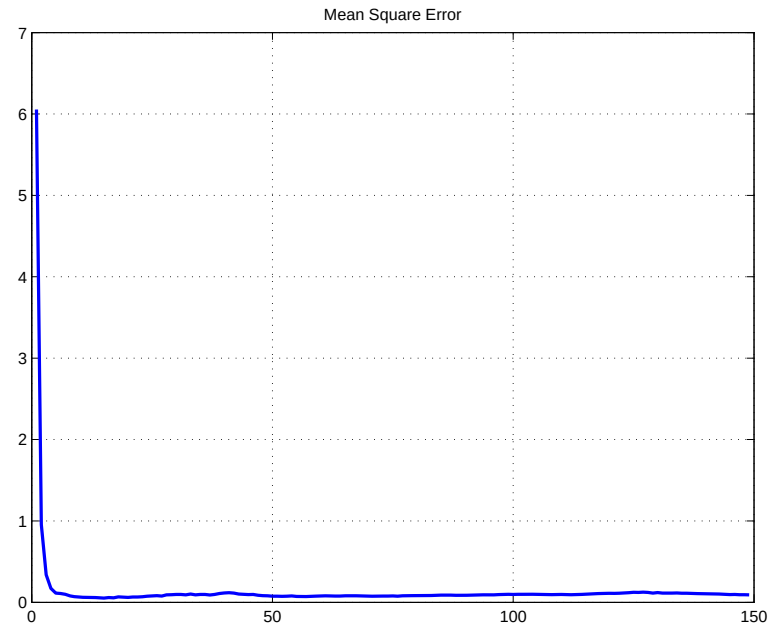


(b) *Assemply parts detected and segmented from the background*

Figure 5.14: *Parts are in a predefined initial location but their poses are random. Human solve the pose estimation problem in an intuitive manner.*

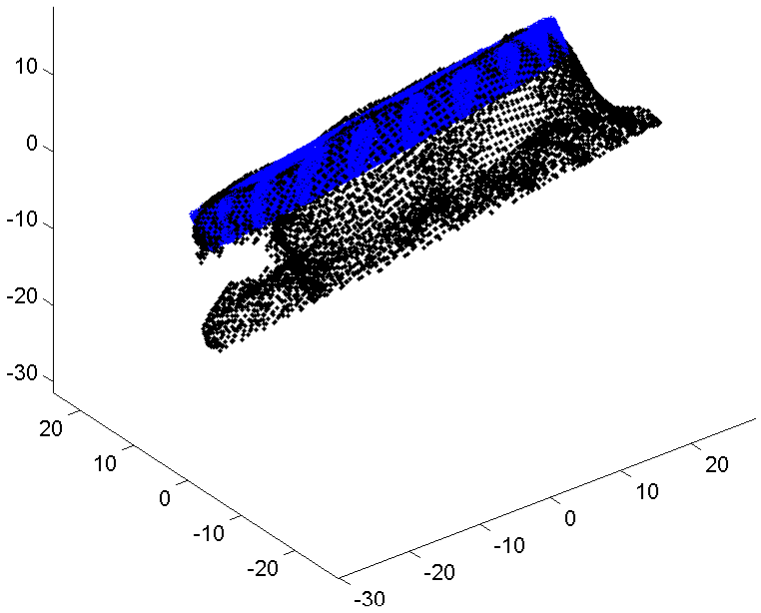


(a) *Exhaust Turbine vs Cluster 1*

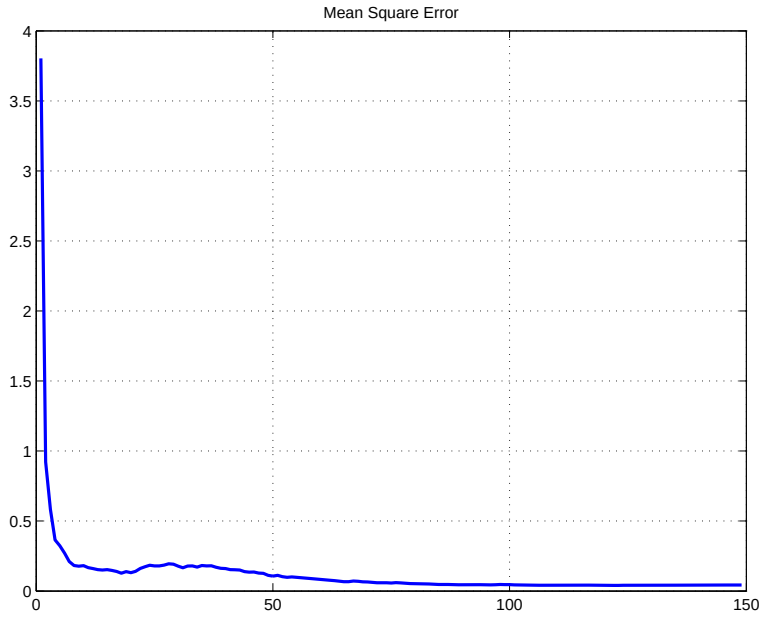


(b) *Part Fitting error*

Figure 5.15: *1-N part alignment and registration for part recognition. Exhaust Turbine is compared against Cluster 1 extracted from the scene.*

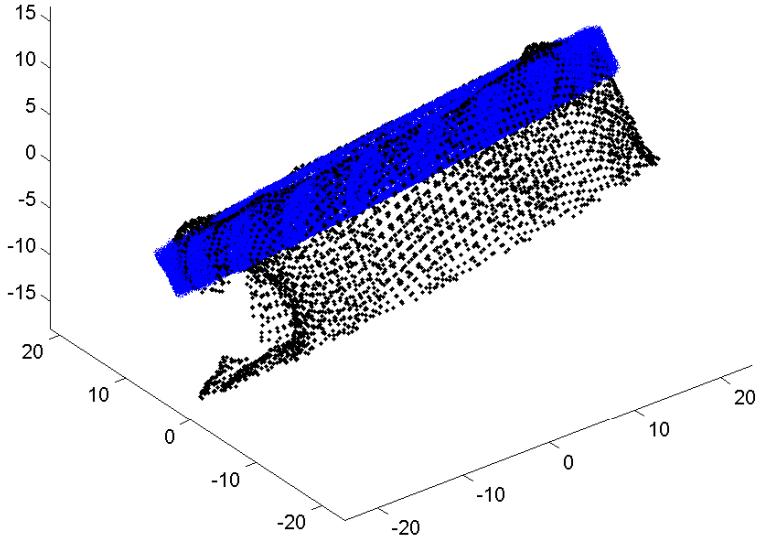


(a) *Exhaust Turbine vs Cluster 2*

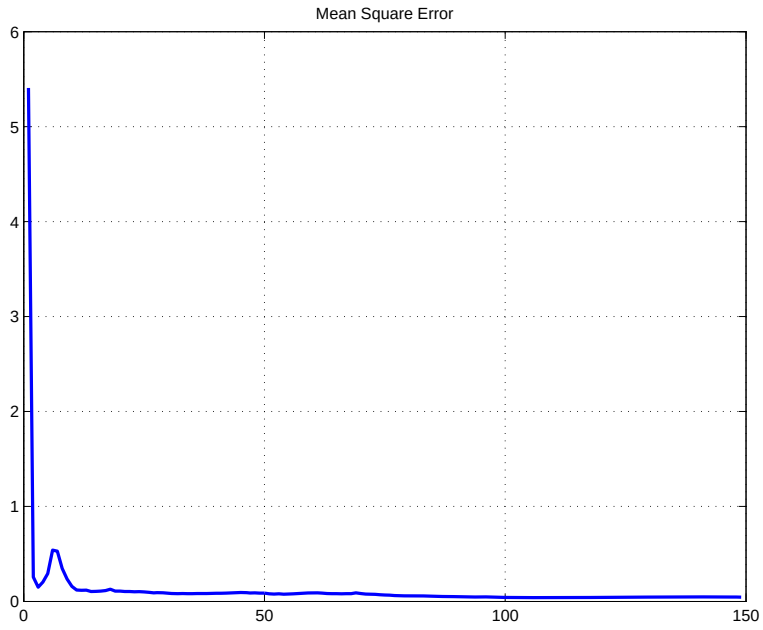


(b) *Part Fitting error*

Figure 5.16: *1-N part alignment and registration for part recognition. Exhaust Turbine is compared against Cluster2 extracted from the scene.*

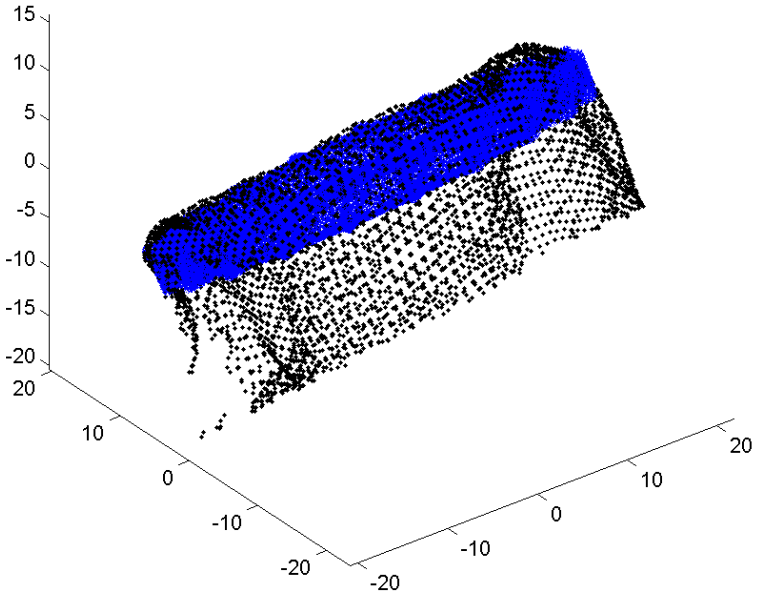


(a) *Exhaust Turbine vs Cluster 3*

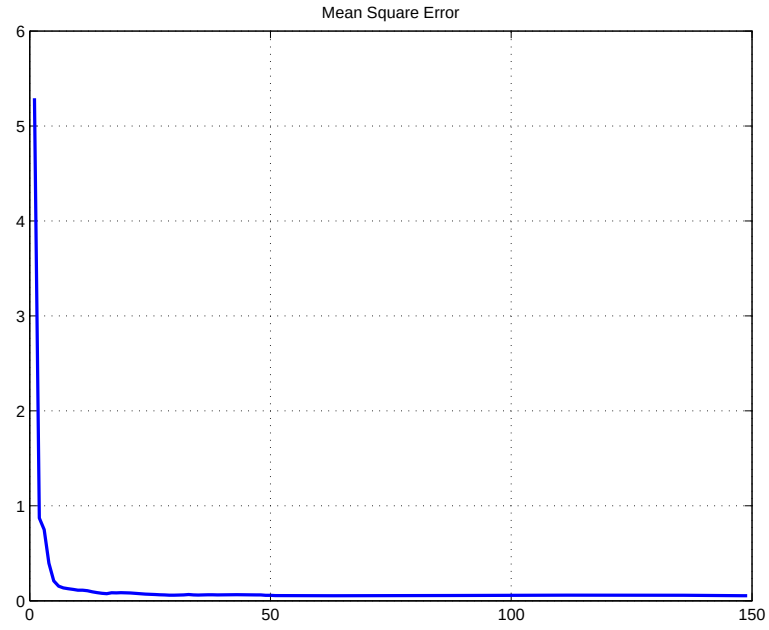


(b) *Part Fitting error*

Figure 5.17: *1-N part alignment and registration for part recognition. Exhaust Turbine is compared against Cluster 3 extracted from the scene.*

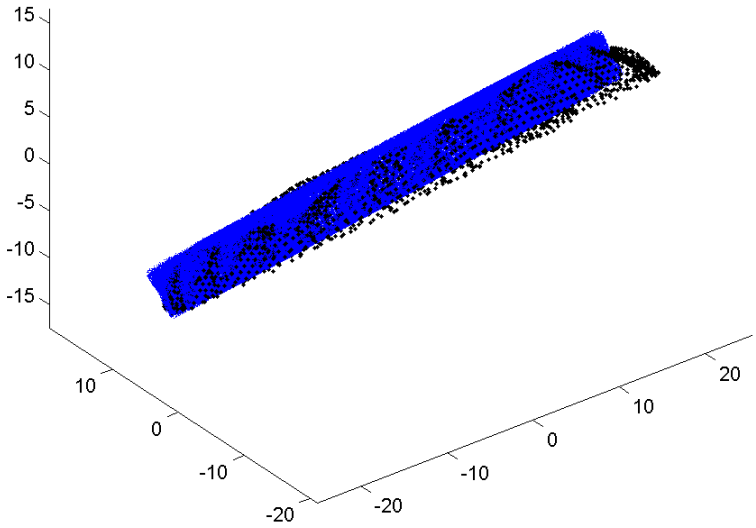


(a) *Exhaust Turbine vs Cluster 4*

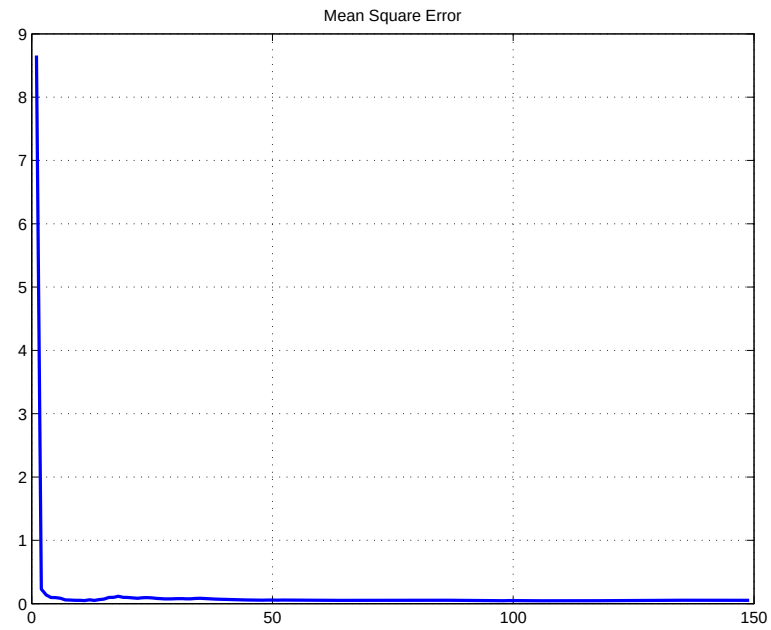


(b) *Part Fitting error*

Figure 5.18: *1-N part alignment and registration for part recognition. Exhaust Turbine is compared against Cluster 4 extracted from the scene.*



(a) *Exhaust Turbine vs Cluster 5*



(b) *Part Fitting error*

Figure 5.19: *1-N part alignment and registration for part recognition. Exhaust Turbine is compared against Cluster 5 extracted from the scene. Cluster 5 is recognized as a Exhaust Turbine*

Regardless the control point, the algorithm use the point cloud generated from

the 3D CAD model as a target and compare this target against the N point clouds or clusters extracted from the scanned scene. This approach allow the system to evaluate the alignment error for each assembly part detected under the assumption that the minimum error belongs to the matching cluster. (Fig. 5.15, 5.16, 5.17, 5.18 and 5.19)

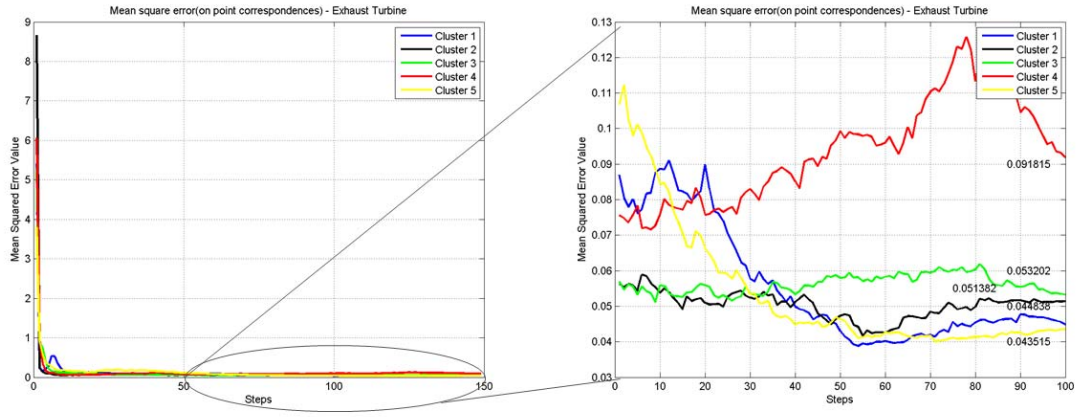


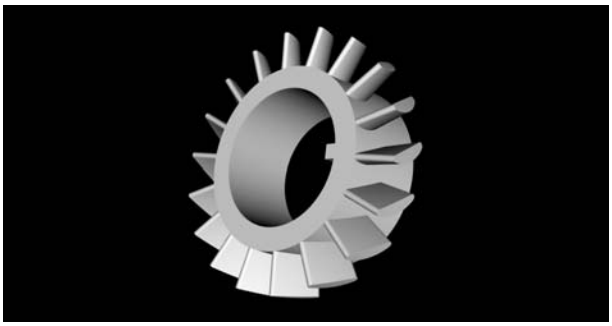
Figure 5.20: *Error analysis: Exhaust Turbine is selected as a target Assemply part and then compared against all the extracted clusters.*

Once this 1 to N analysis is over as shown in Fig. 5.20 the system identify the cluster that represents the best matching cluster and therefore the cluster now is recognized. In Fig. 5.20 the target point cloud is Exhaust Turbine and after the analysis the best matching cluster is Cluster 5. Therefore Cluster 5 now is identify as an Exhaust Turbine .

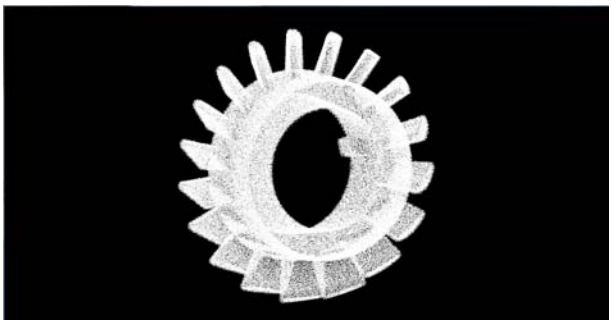
Experimentation showed that our $Iterative-WEOPA$ algorithm successfully detected the corresponding point cloud matching between point clouds obtained from scanning and point clouds generated from 3D CAD models. Clusters identification and scene labeling provide to the system an accurate tracking mechanism to

detect changes in the scene and report such a changes.

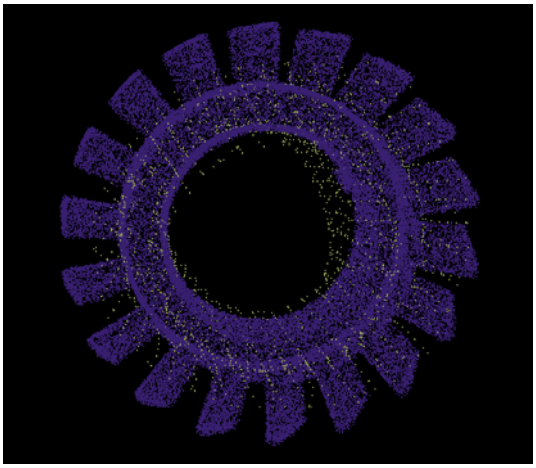
The matching results in the first control point (*Initial location*) are illustrated in Fig. 5.21, Fig. 5.22 and Fig. 5.23.



(a) *First Compressor: CAD model*

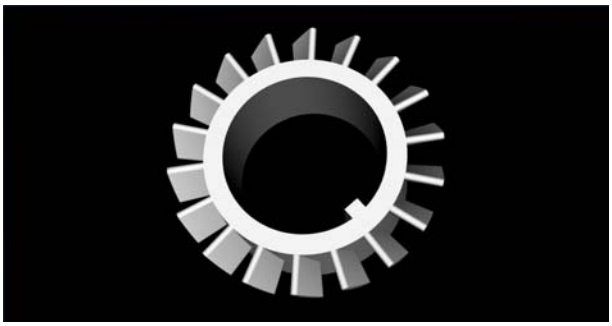


(b) *Point Cloud*

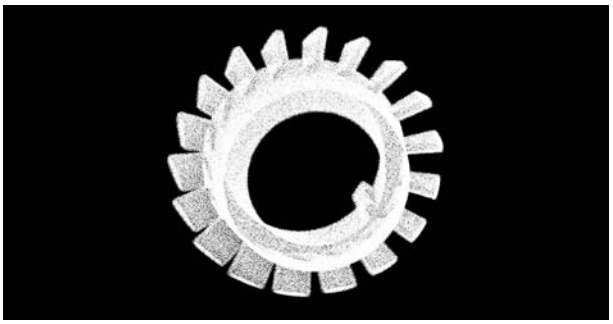


(c) *Alignment and detection*

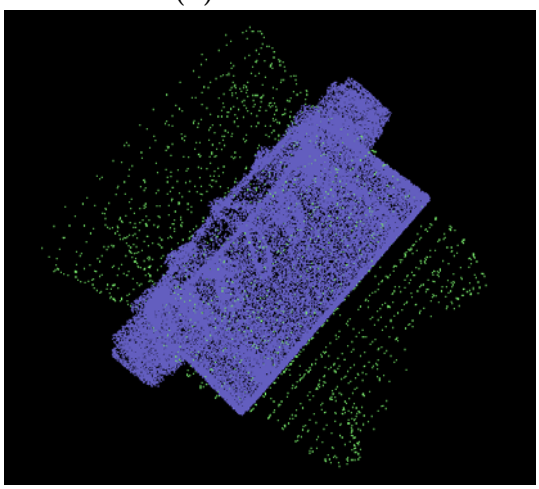
Figure 5.21: *3D CAD models for each assembly part are provided to the system. These CAD models are used to generate a point cloud targets. Then, several point clouds are extracted (clustering) from the 3D scence descriving a single part extraction. Fig. (a) shows the CAD model. Fig. (b) shows the conversion from CAD to point clouds using ray-tracing algorithm. Fig. (c) shows the assembly part detection as a correspondence of the reference point cloud and the state-point cloud.*



(a) *Second Compressor: CAD model*

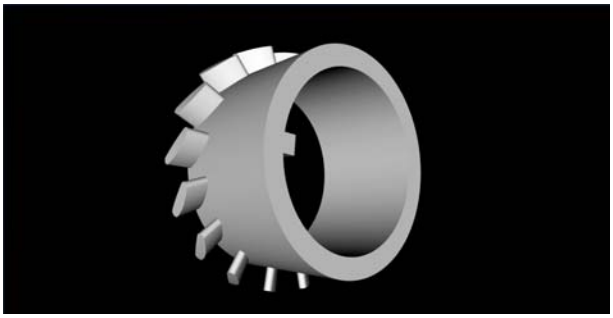


(b) *Point Cloud*

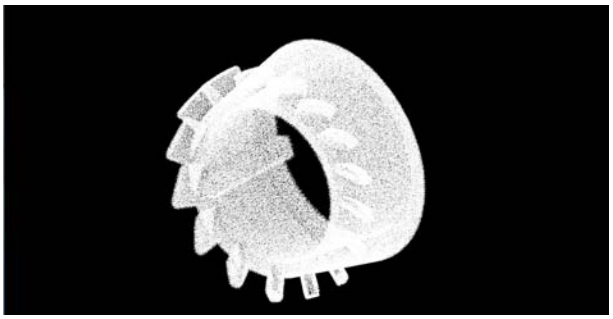


(c) *Alignment and detection*

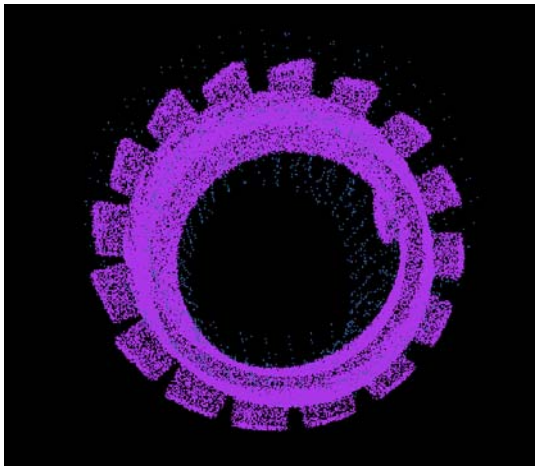
Figure 5.22: *3D CAD models for each assembly part are provided to the system. These CAD models are used to generate a point cloud targets. Then, several point clouds are extracted (clustering) from the 3D scence descriving a single part extraction. Fig. (a) shows the CAD model. Fig. (b) shows the conversion from CAD to point clouds using ray-tracing algorithm. Fig. (c) shows the assembly part detection as a correspondence of the reference point cloud and the state-point cloud.*



(a) *Third Compressor: CAD model*



(b) *Point Cloud*

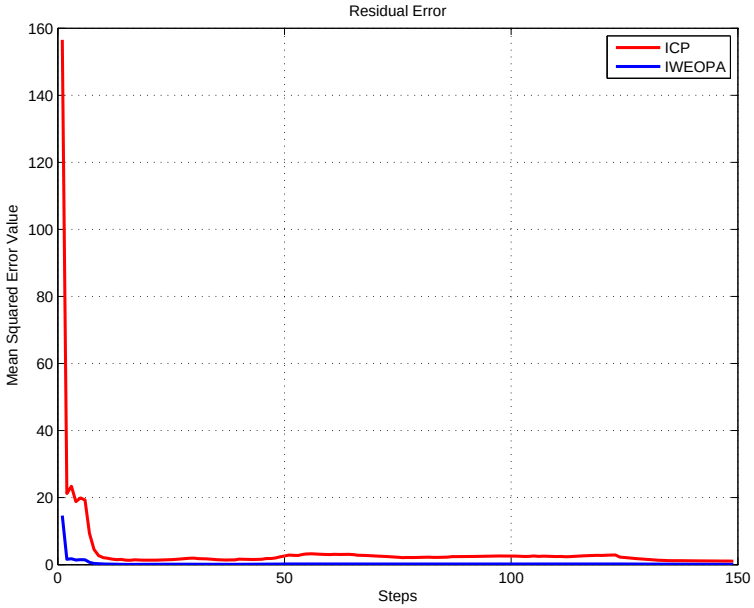


(c) *Alignment and detection*

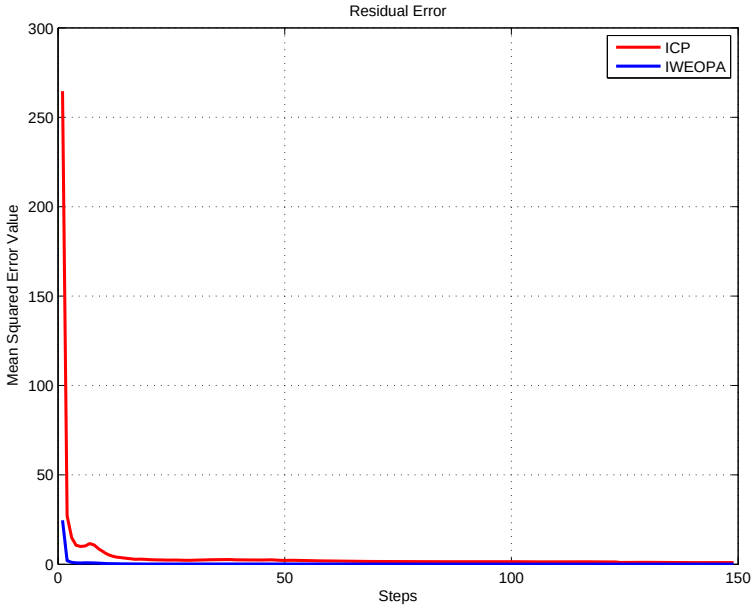
Figure 5.23: *3D CAD models for each assembly part are provided to the system. These CAD models are used to generate a point cloud targets. Then, several point clouds are extracted (clustering) from the 3D scence descriving a single part extraction. Fig. (a) shows the CAD model. Fig. (b) shows the conversion from CAD to point clouds using ray-tracing algorithm. Fig. (c) shows the assembly part detection as a correspondence of the reference point cloud and the state-point cloud.*

Moreover, we compared the results with the classical ICP algorithm. The proposed algorithm, perform better in both scenarios and for every part (Fig. 5.24,

5.25 and 5.26). In some cases ICP showed a degenerative behavior. In order to evaluate and compare performance of our approach, a residual error was computed at each step. The residual was computed as the mean square distance between the points of the current mesh and the model mesh and their closest point.

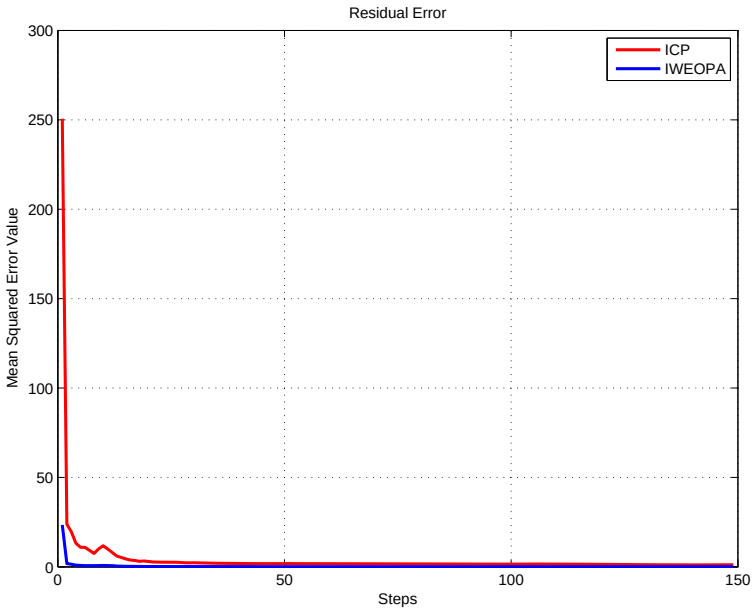


(a) *Rear Bearing*

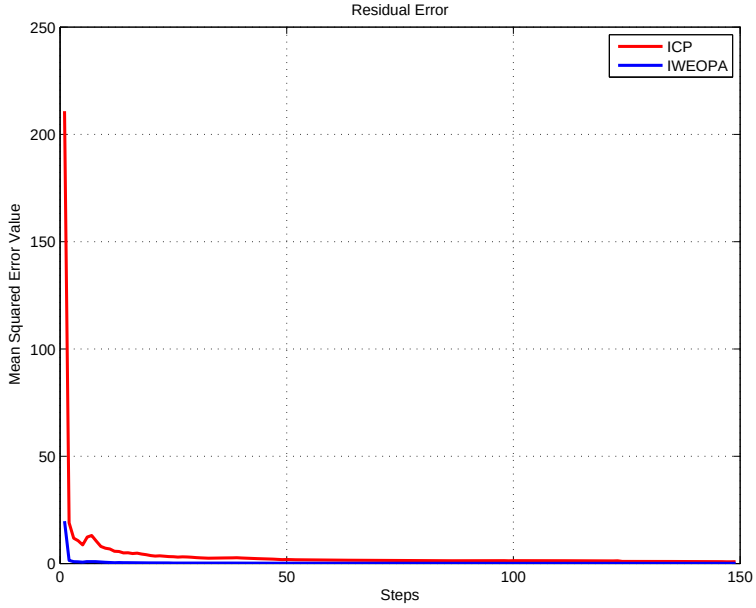


(b) *First Compressor*

Figure 5.24: *Residual errors at each step of the algorithm for each assembly part.*

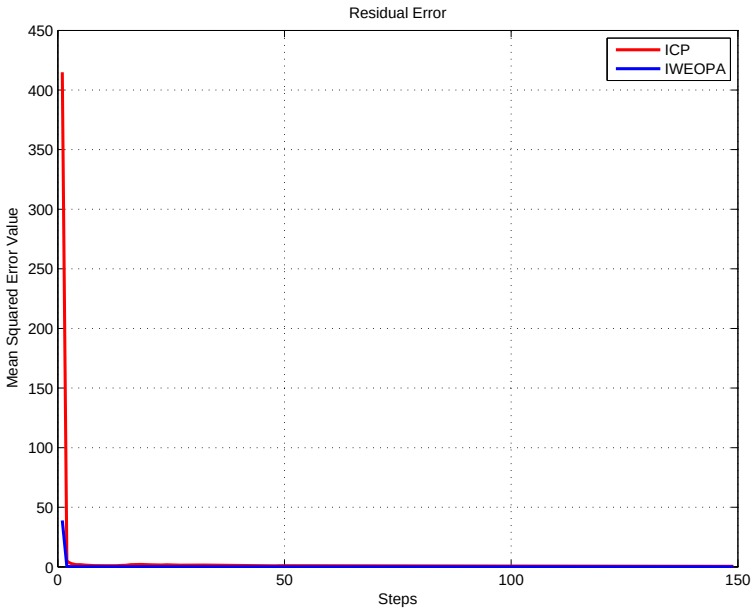


(a) *Second Compressor*



(b) *Third Compressor*

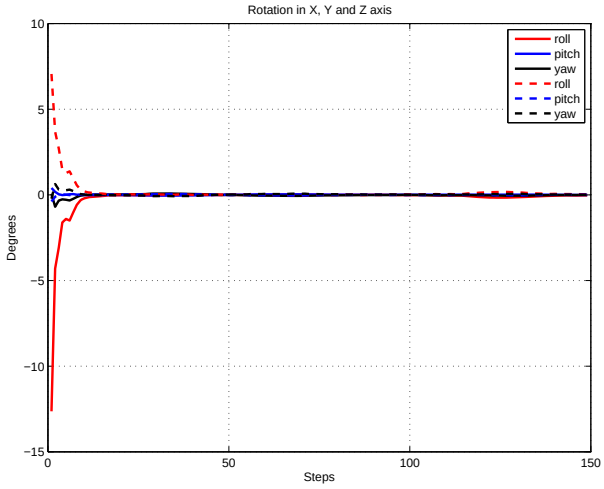
Figure 5.25: *Residual errors at each step of the algorithm for each assembly part.*



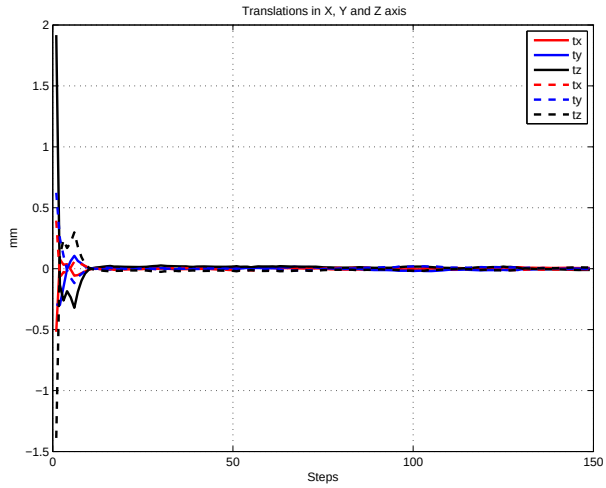
(a) *Exhaust Turbine*

Figure 5.26: *Residual errors at each step of the algorithm for each assembly part.*

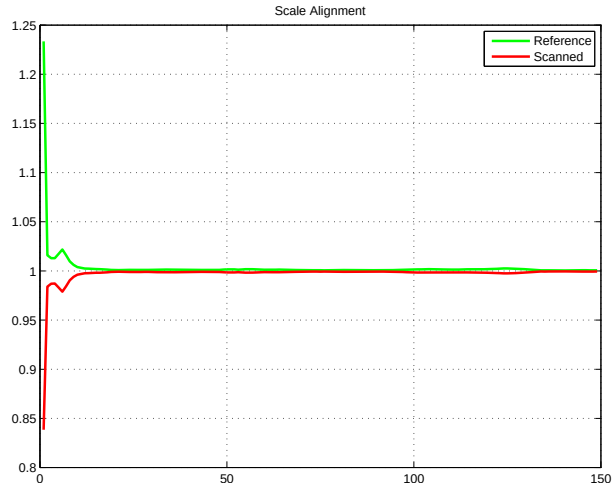
In order to define the number of steps required for the algorithm to converge, we analyzed 3D rotations, translations and scale changes in each step (Fig. 5.27, 5.28, 5.29, 5.30 and 5.31). After 100 steps very small changes were observed in terms of rotation, translation and scale. Therefore, we set 150 as a fix number of cycles for this specific experiment. More iterations do not have any impact in the numerical solutions but it start having an impact in the processing time.



(a) *Rotations*

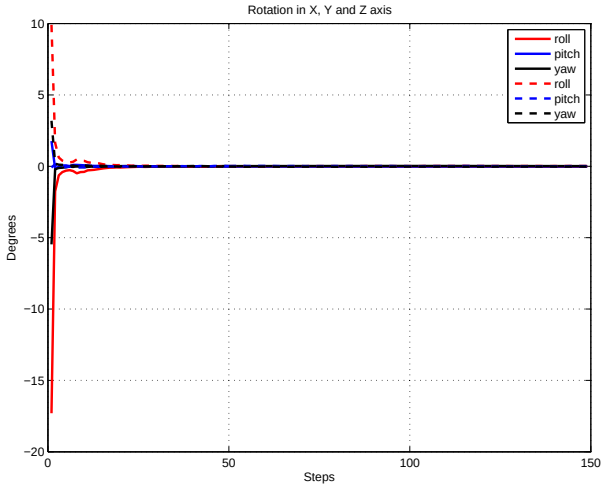


(b) *Translations*

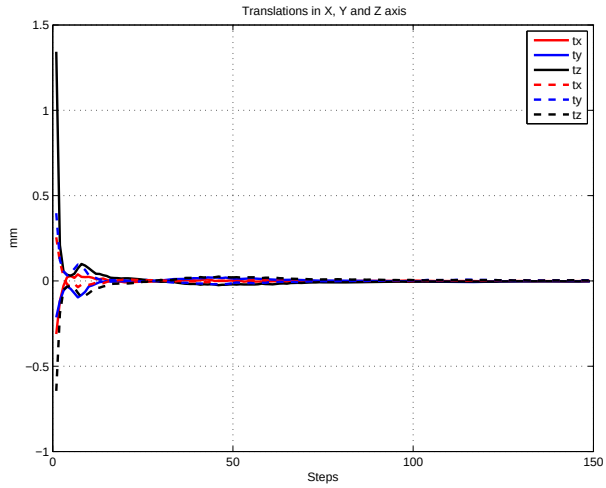


(c) *Scale convergence*

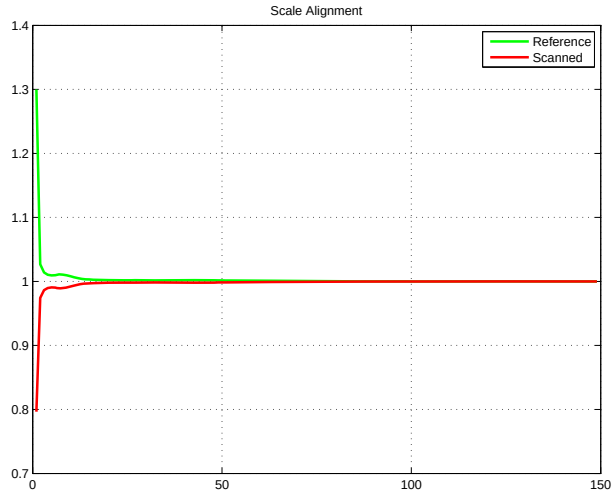
Figure 5.27: *Transformations for Rear Bearing. Fig. (a) show rotations in (roll, pitch and yaw). Fig. (b) show translations in x, y and z. Fig. (c) shows scale transformations.*



(a) *Rotations*

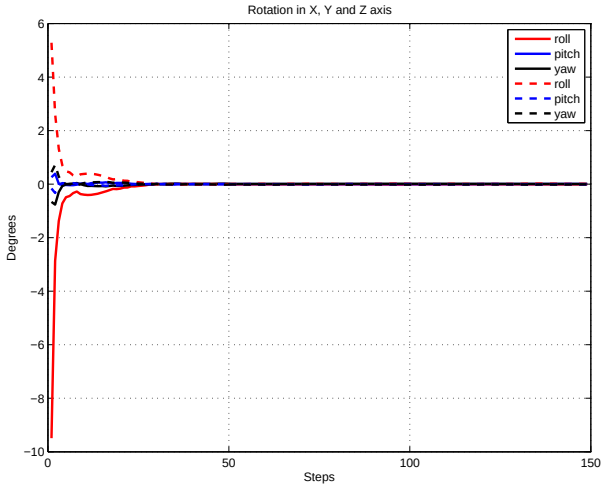


(b) *Translations*

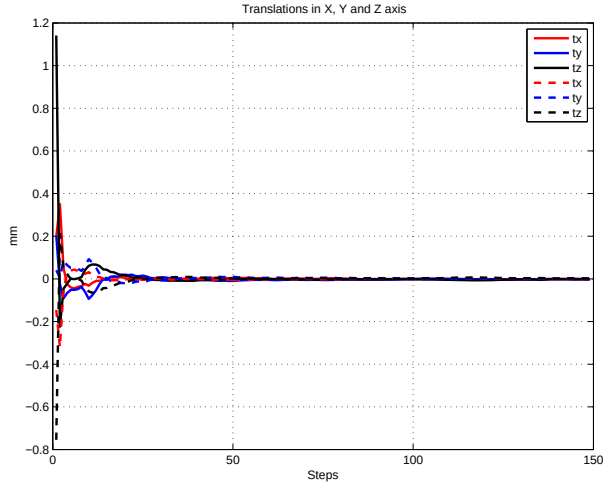


(c) *Scale convergence*

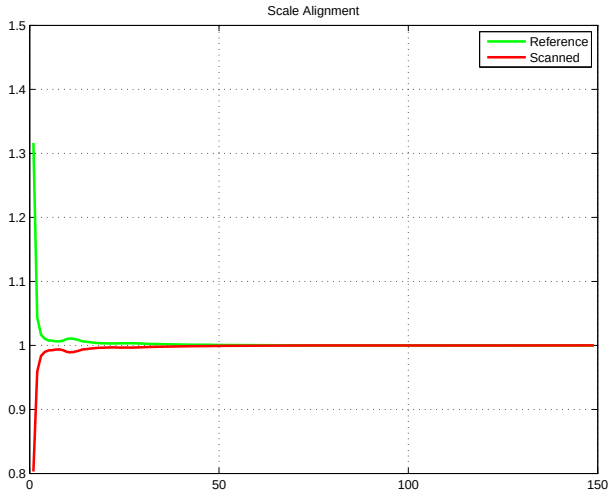
Figure 5.28: *Transformations for First Compressor. Fig. (a) show rotations in (roll, pitch and yaw). Fig. (b) show translations in x, y and z. Fig. (c) shows scale transformations.*



(a) *Rotations*

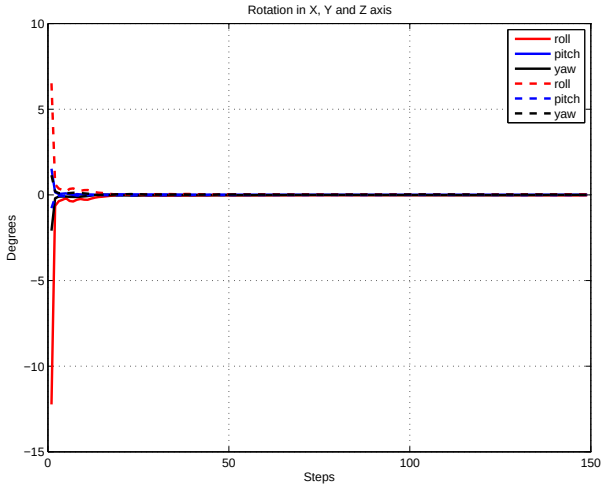


(b) *Translations*

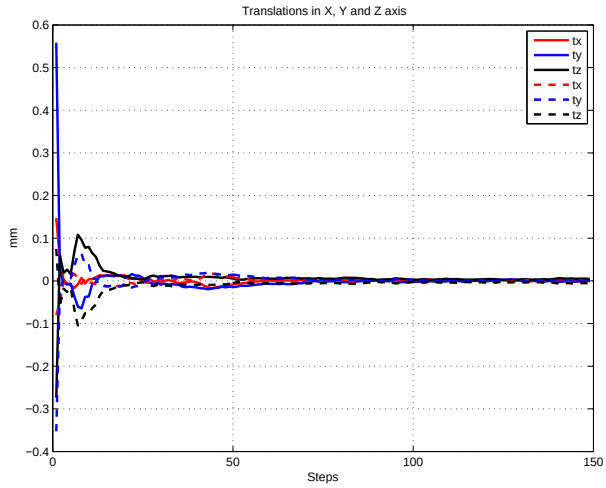


(c) *Scale convergence*

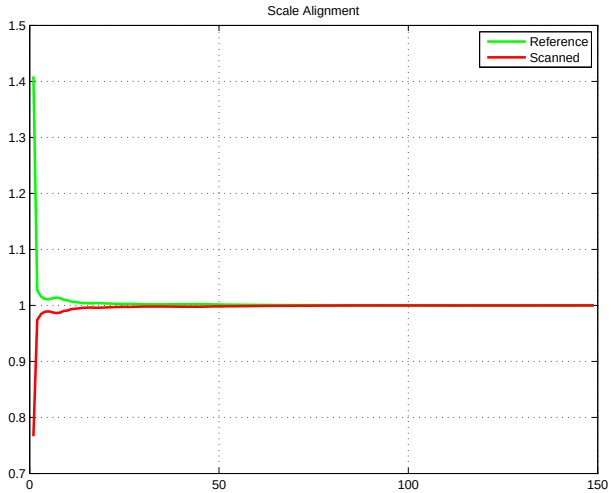
Figure 5.29: *Transformations for Second Compressor. Fig. (a) show rotations in (roll, pitch and yaw). Fig. (b) show translations in x, y and z. Fig. (c) shows scale transformations.*



(a) *Rotations*

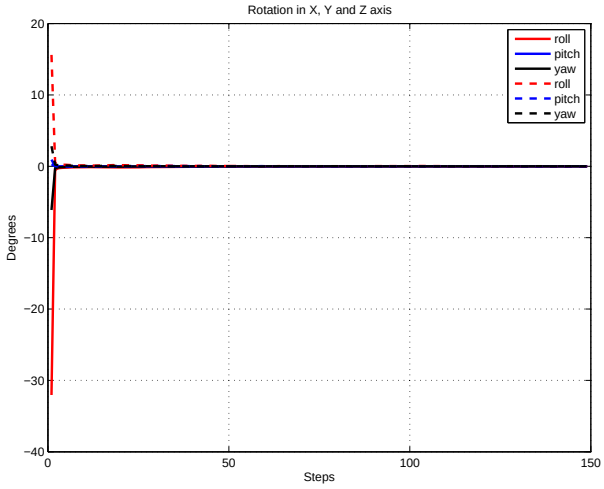


(b) *Translations*

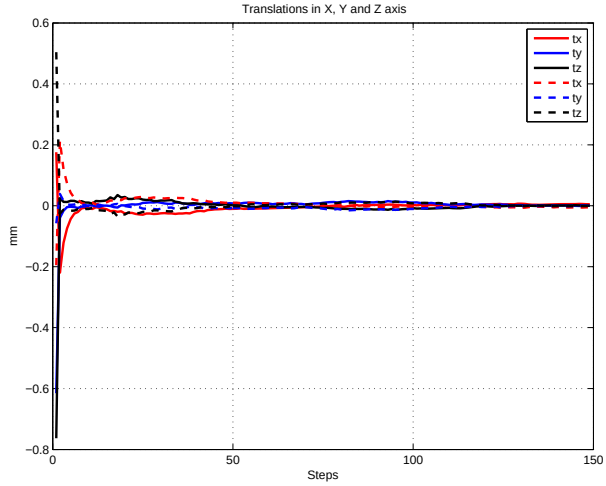


(c) *Scale convergence*

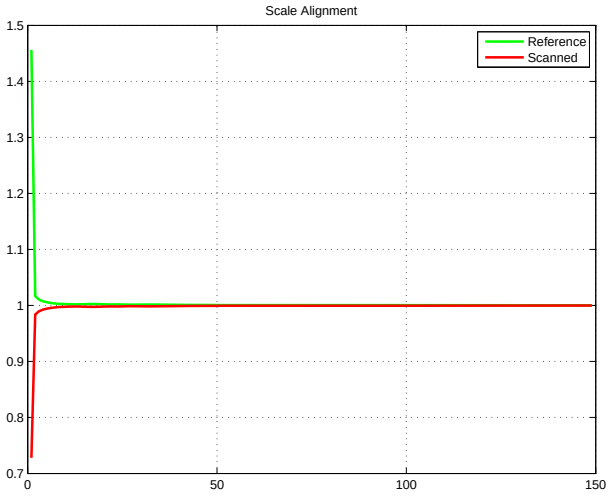
Figure 5.30: *Transformations for Third Compressor. Fig. (a) show rotations in (roll, pitch and yaw). Fig. (b) show translations in x, y and z. Fig. (c) shows scale transformations.*



(a) *Rotations*



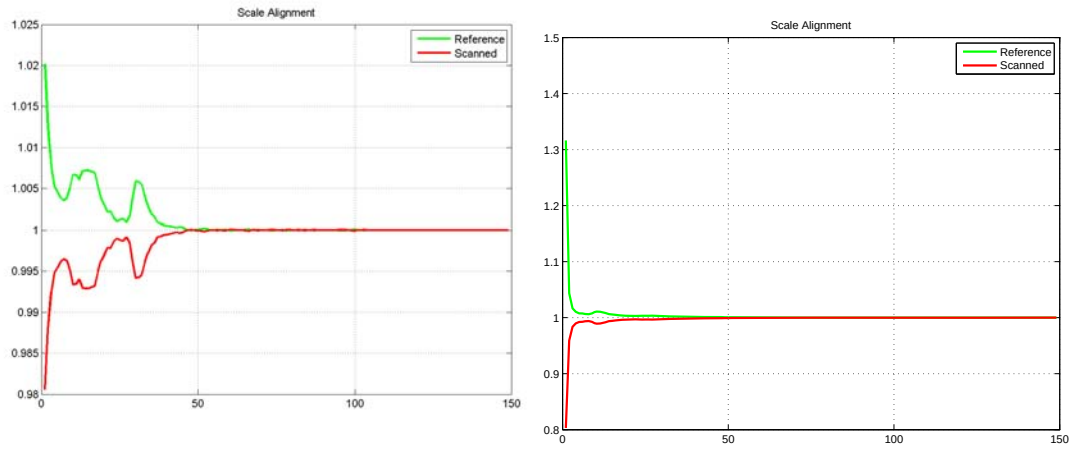
(b) *Translations*



(c) *Scale convergence*

Figure 5.31: *Transformations for Exhaust Turbine. Fig. (a) show rotations in (roll, pitch and yaw). Fig. (b) show translations in x, y and z. Fig. (c) shows scale transformations.*

3D point clouds are defined as a rigid body. Therefore, rotation, translation and scaling transformation do not deform the point clouds. This allows the algorithm to use scaling as a compensatory transformation between a noisy point cloud and the point cloud generated by the CAD model. In addition, scaling transformation evaluated at step one is used also as a termination flag. This is valid under the assumption that if scaling transformation is above an specific threshold (Fig. 5.32) then, there is a high probability that the scanned part is actually different than the CAD model used for the query.



(a) *Second Compressor* **(b)** *Large Compressor*

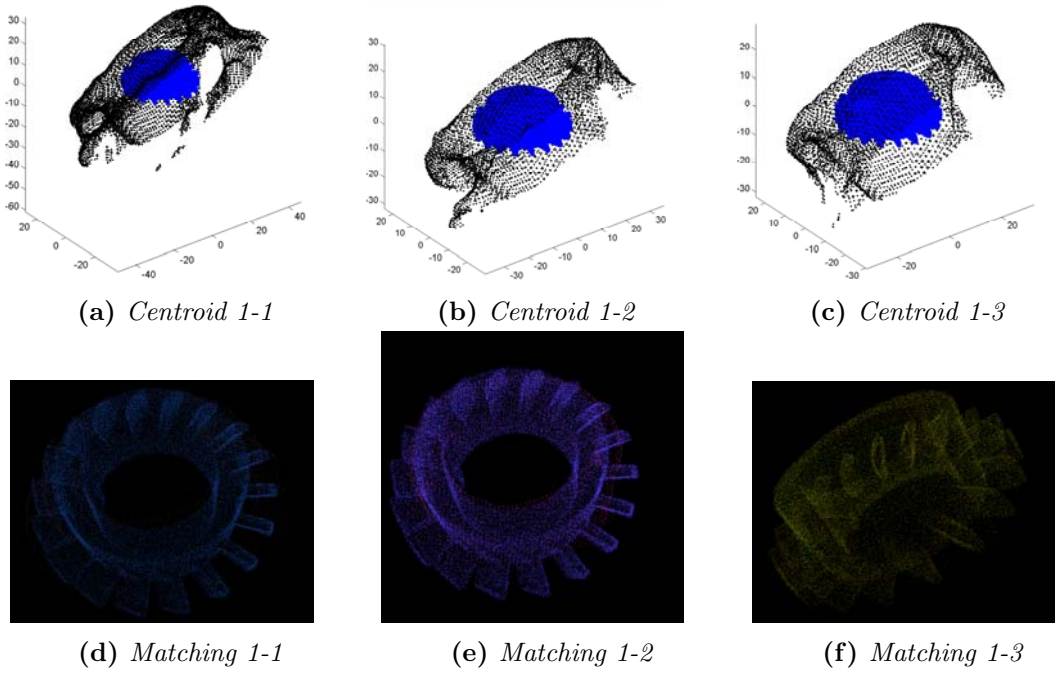
Figure 5.32: *Scale convergence analysis. (a) Scanned part with no scale variations. (b) Scanned part is larger than the CAD model*

Finally, we observed some convergence irregularities in some particular cases. This random behavior was observed in ICP algorithm as well as in $Iterative - WEOPA$ algorithm and it is due to the complexity of the 3D model that it's represented by the number of points, faces and face orientations. A random behavior in the algorithm runs the risk of ending the process in a local minima. It was included a roll-back stability analysis in order to detect these *special* cases.

5.4.2.3 Algorithm Characterization

In previous section we analyzed the performance of the $Iterative - WEOPA$ algorithm. We also experimentally demonstrate the error non-convergence to zero when a part is matched against a completely different part.

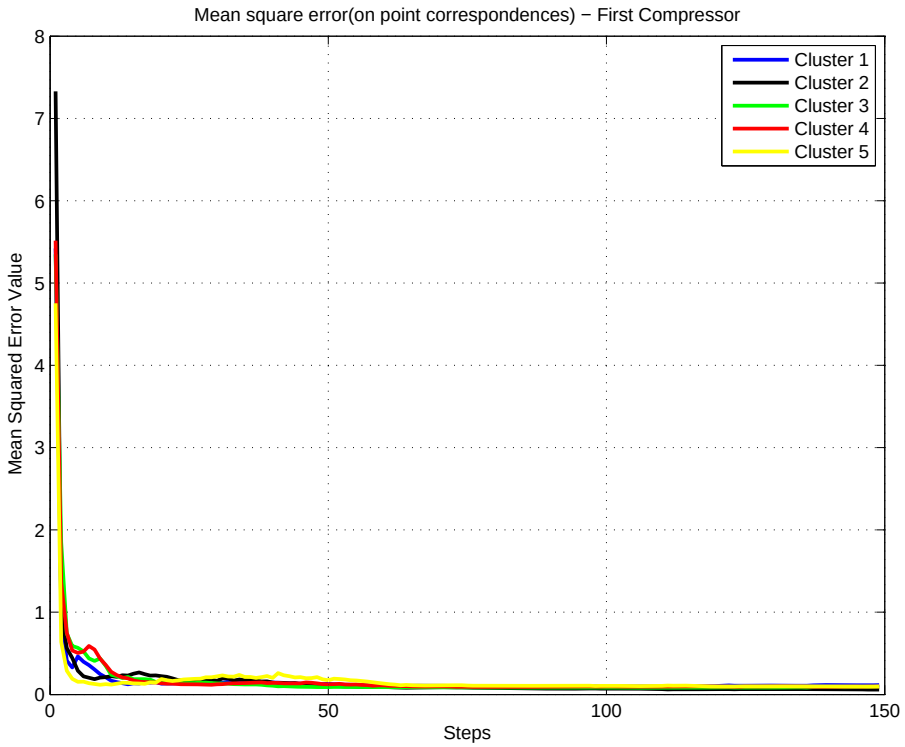
A complex problem in computer vision is detecting and identifying a part in a subset of parts that are similar. In order to test the robustness of our model, we analyzed more exhaustively five parts that are geometrically similar.



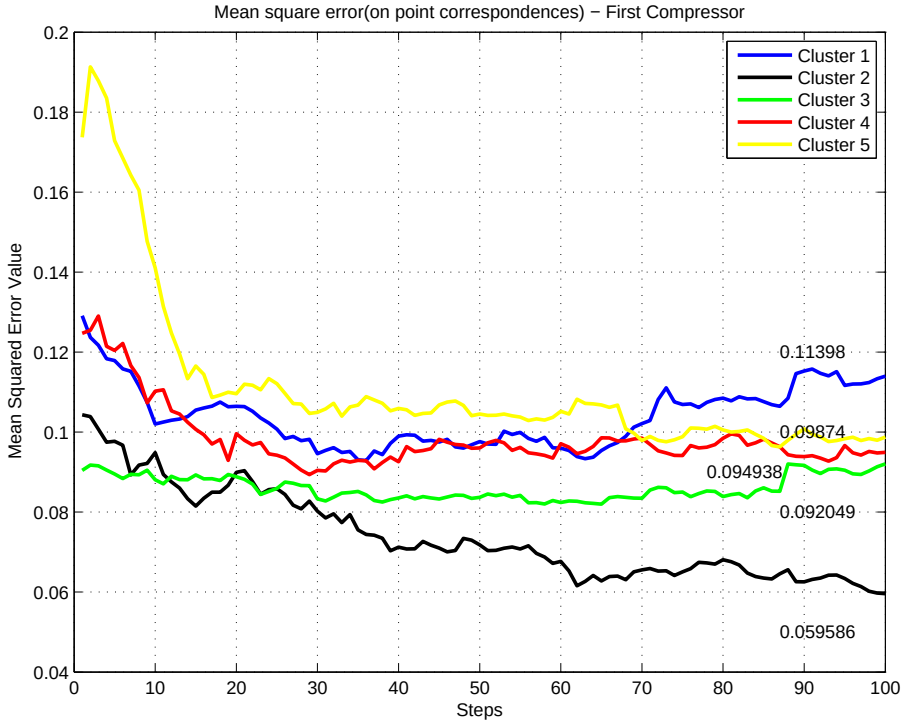
(a) *Centroid 1-1* **(b)** *Centroid 1-2* **(c)** *Centroid 1-3*

(d) *Matching 1-1* **(e)** *Matching 1-2* **(f)** *Matching 1-3*

Figure 5.33: *First Compressor generated from a 3D CAD model is compared agaist Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor).*



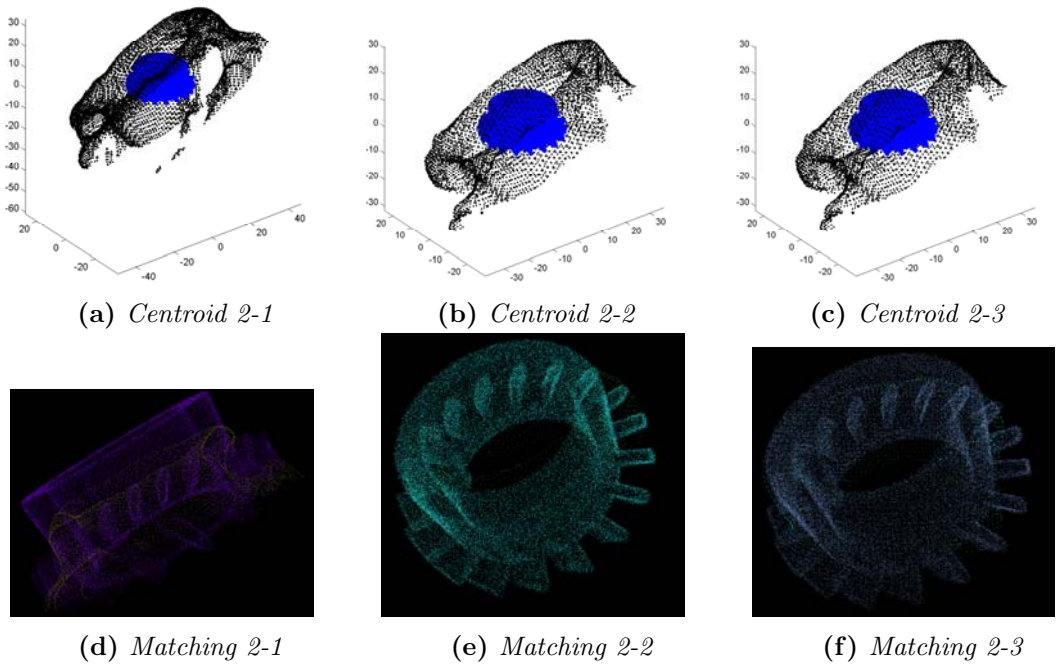
(a) *MSE*



(b) *Zoomed Image*

Figure 5.34: *First Compressor identified in a subset of similar parts. Where Cluster 1 (Rear Bearing), Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor) and Cluster 5 (Exhaust Turbine)*

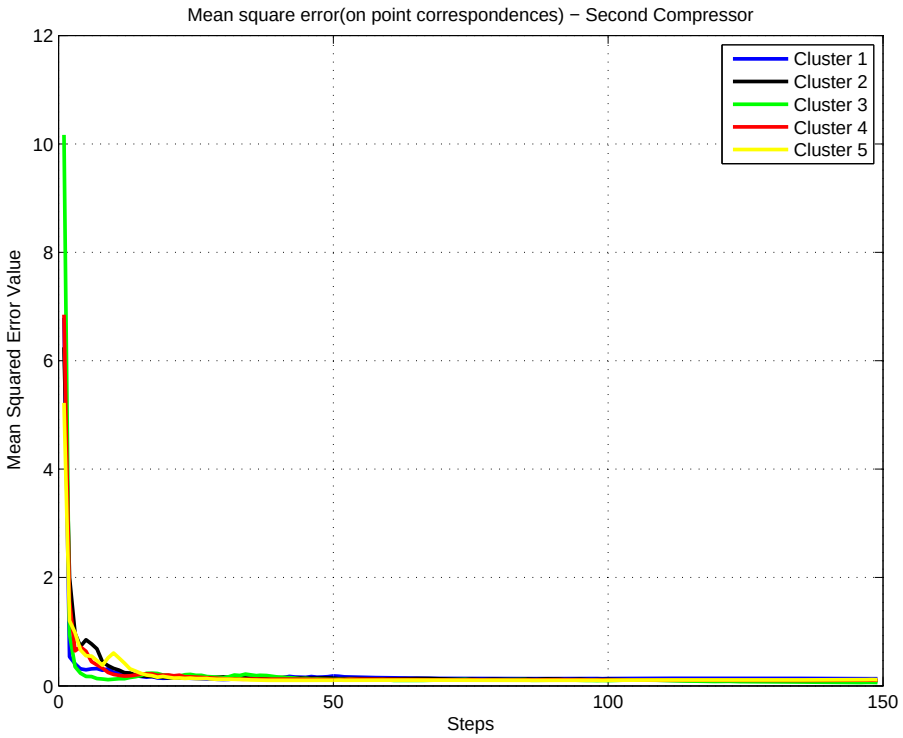
Due to the intrinsic noise and resolution of the sensor the generated point cloud has many irregularities that eventually can affect the precision of the algorithm. Fig. 5.34, Fig. 5.36 and Fig. 5.38 show the mean square error on point correspondence between five parts, where three of them have a lot of similarities between each other. Despite these irregularities, the algorithm was able to identify the correct part in all the experiments (Fig. 5.33, Fig. 5.35 and Fig. 5.37).



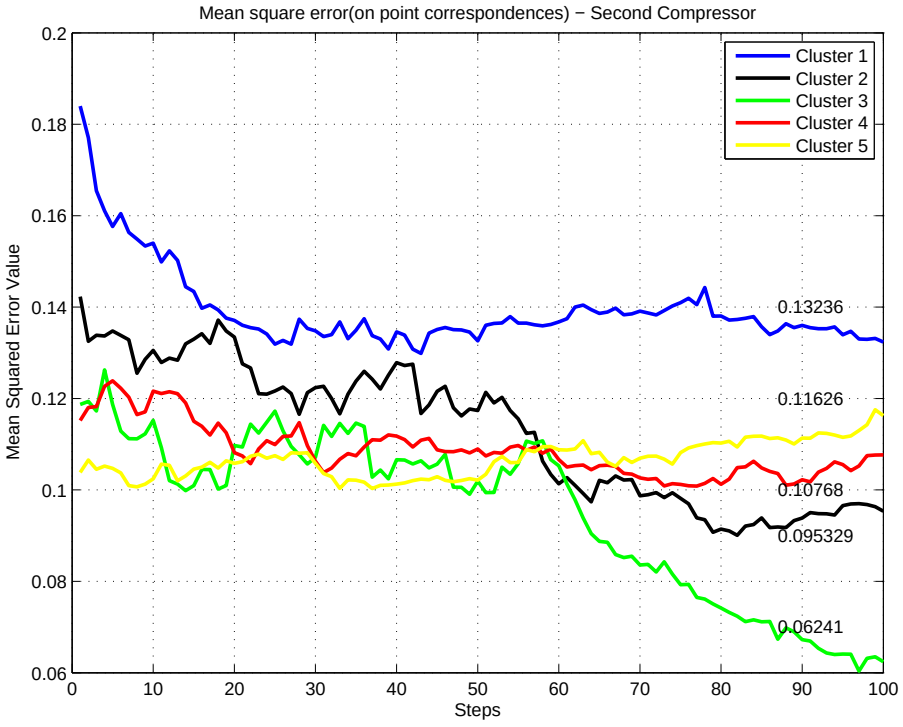
(a) *Centroid 2-1* **(b)** *Centroid 2-2* **(c)** *Centroid 2-3*

(d) *Matching 2-1* **(e)** *Matching 2-2* **(f)** *Matching 2-3*

Figure 5.35: *Second Compressor generated from a 3D CAD model is compared agaist Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor).*



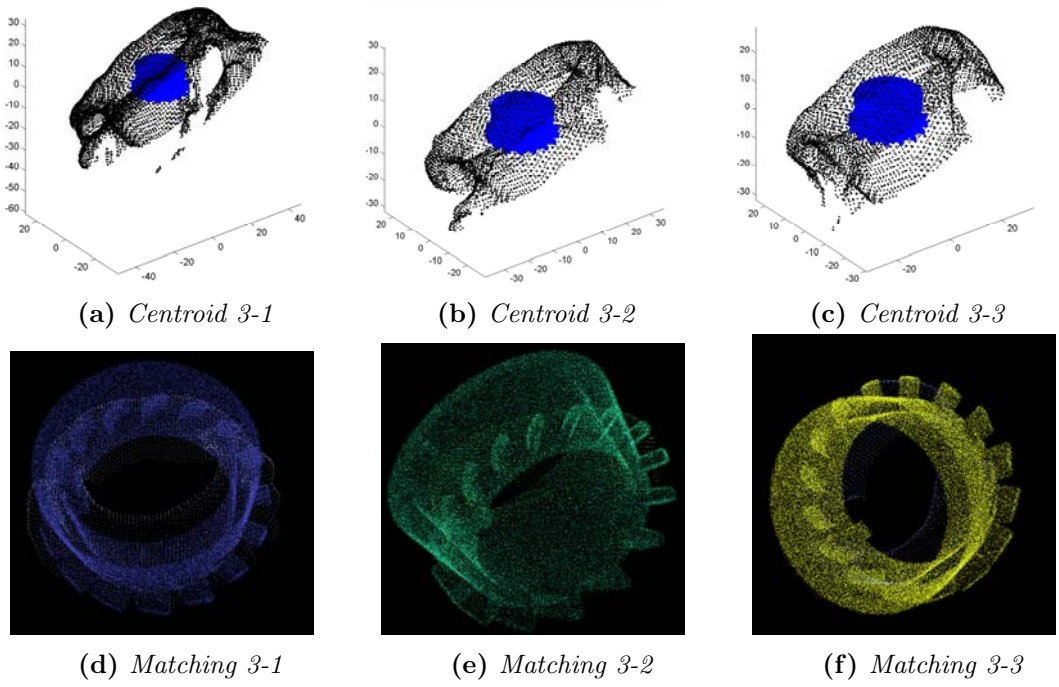
(a) *MSE*



(b) *Zoomed Image*

Figure 5.36: *Second Compressor identified in a subset of similar parts. Where Cluster 1 (Rear Bearing), Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor) and Cluster 5 (Exhaust Turbine)*

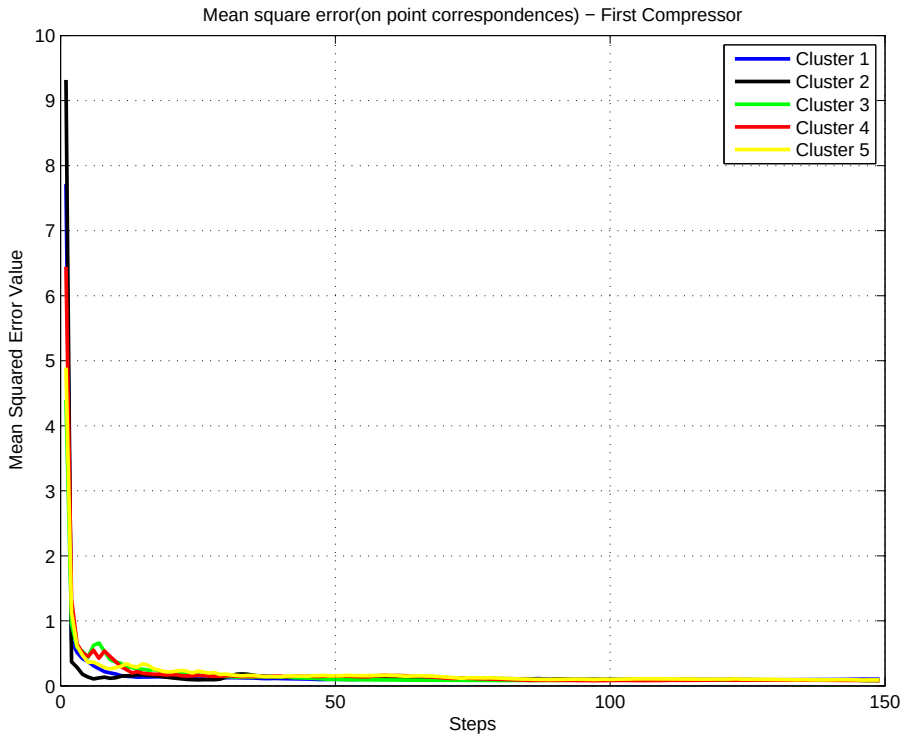
Any mean square error on point correspondence bellow 0.09 can be considered as a true positive. Fig. 5.34, Fig. 5.36 and Fig. 5.38 show that the MSE of the three considered most similar parts are bellow the threshold. In order to reduce the uncertainty, our algorithm uses a local comparison between parts that belong to an specific assembly. This step helps to sort the error and find the minimum of the MSE on point correspondence and, therefore, identify the part.



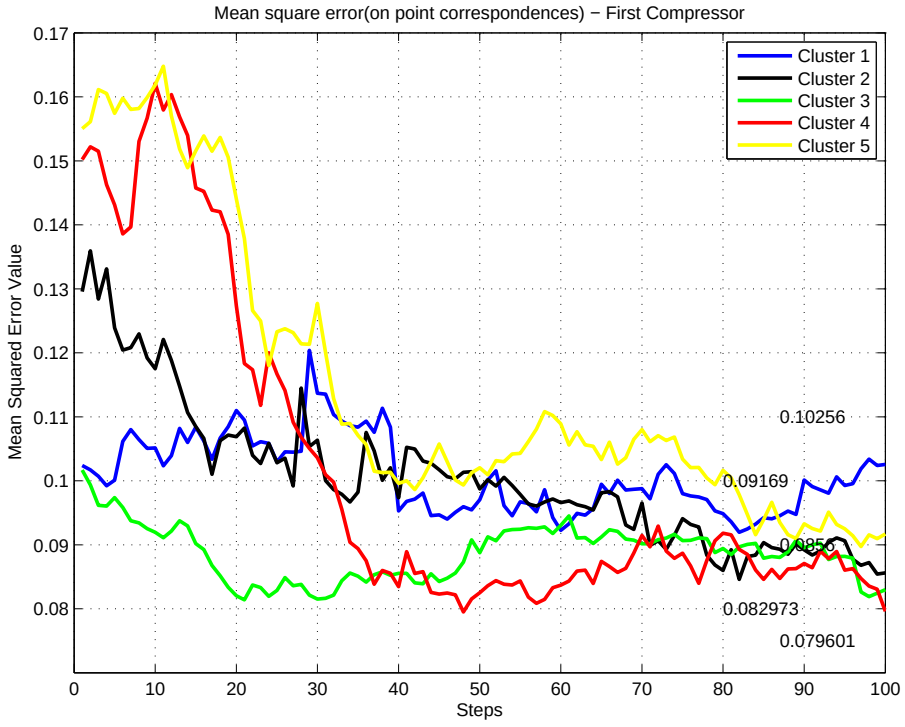
(a) *Centroid 3-1* **(b)** *Centroid 3-2* **(c)** *Centroid 3-3*

(d) *Matching 3-1* **(e)** *Matching 3-2* **(f)** *Matching 3-3*

Figure 5.37: *Third Compressor generated from a 3D CAD model is compared agaist Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor).*

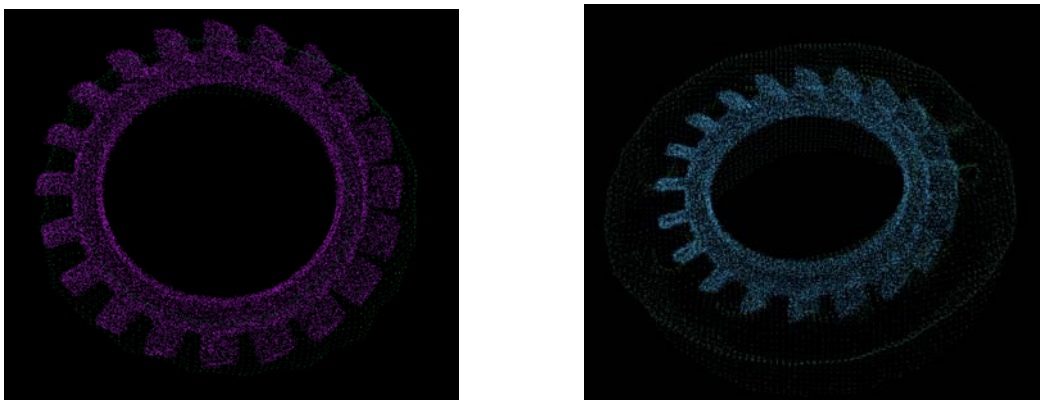


(a) *MSE*



(b) *Zoomed Image*

Figure 5.38: *Third Compressor identified in a subset of similar parts. Where Cluster 1 (Rear Bearing), Cluster 2 (First Compressor) Cluster 3 (Second Compressor) and Cluster 4 (Third Compressor) and Cluster 5 (Exhaust Turbine)*



(a) *Matching 5-5* **(b)** *Matching 5-4*

Figure 5.39: *Exhaust Turbine identified in a subset of similar parts. Exhaust Turbine generated from a 3D CAD model is compared agaist Cluster 5 (Exhaust Turbine) and Cluster 1 (Rear Bearing)*

Similar behavior was observed between Exhaust Turbine and Rear Bearing parts (Fig. 5.39, Fig. 5.40, Fig. 5.41 and Fig. 5.42). Features or geometries presented in the Rear Bearing part were reproduced in the point cloud after scanning but most of the features from the Exhaust Turbine where lost due to the size of the geometries and the resolution of the sensor.

Finally, the empirical assumption was that by extracting a more dense point cloud or by increasing the number of steps in the algorithm the part recognition algorithm was going to improve its performance. Experimental results showed that there was no visible improvement in the point correspondence error. However, while almost non-improvement was observe in the matching process, the processing time increased exponentially. (Fig. 5.43)

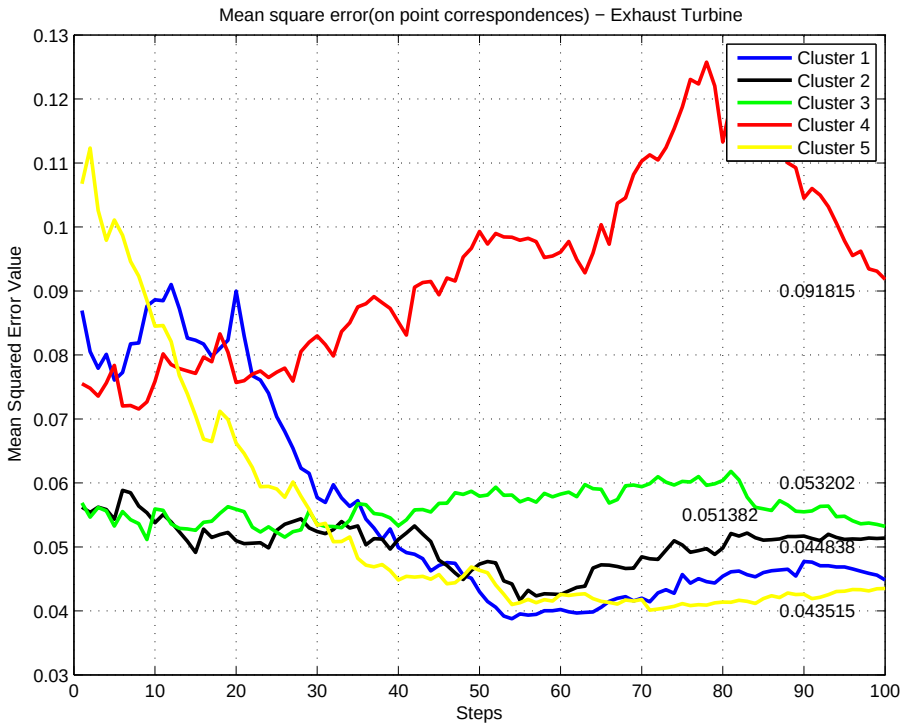


Figure 5.40: *Exhaust Turbine comparison against assembly parts with similar geometry*

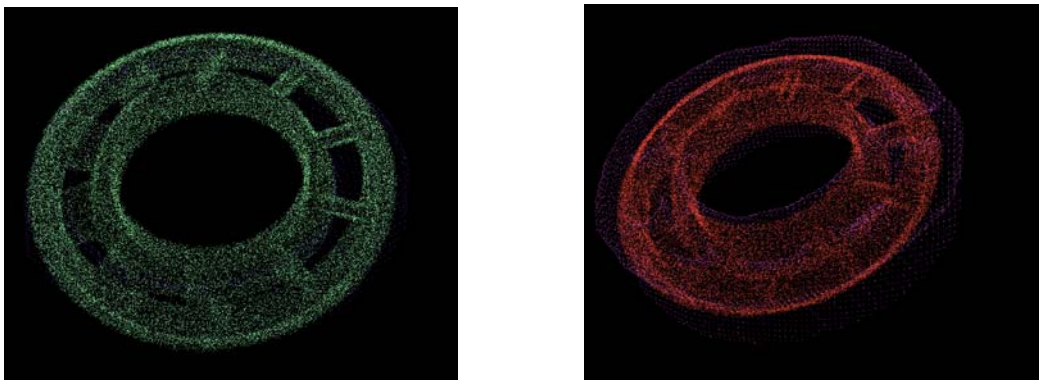
5.4.3 Robot Tracking

We assume that the robot will be able to execute motion commands given to it so that the assembly cell will know the state of the robot.

5.5 Contingency Handling

5.5.1 Collision Avoidance Between Robot and Human

Ensuring safety in the hybrid cell via appropriate control of the robot motion is related to traditional robot collision avoidance. However, interaction scenarios in shared work cells differ from classical settings significantly. For instance, we cannot ensure safety always, if the robot reacts to a sensed imminent collision by moving



(a) *Matching 4-4* **(b)** *Matching 4-5*

Figure 5.41: *Rear Bearing identified in a subset of similar parts. Rear Bearing generated from a 3D CAD model is compared agaist Cluster 1 (Rear Bearing) and Cluster 5 (Exhaust Turbine)*

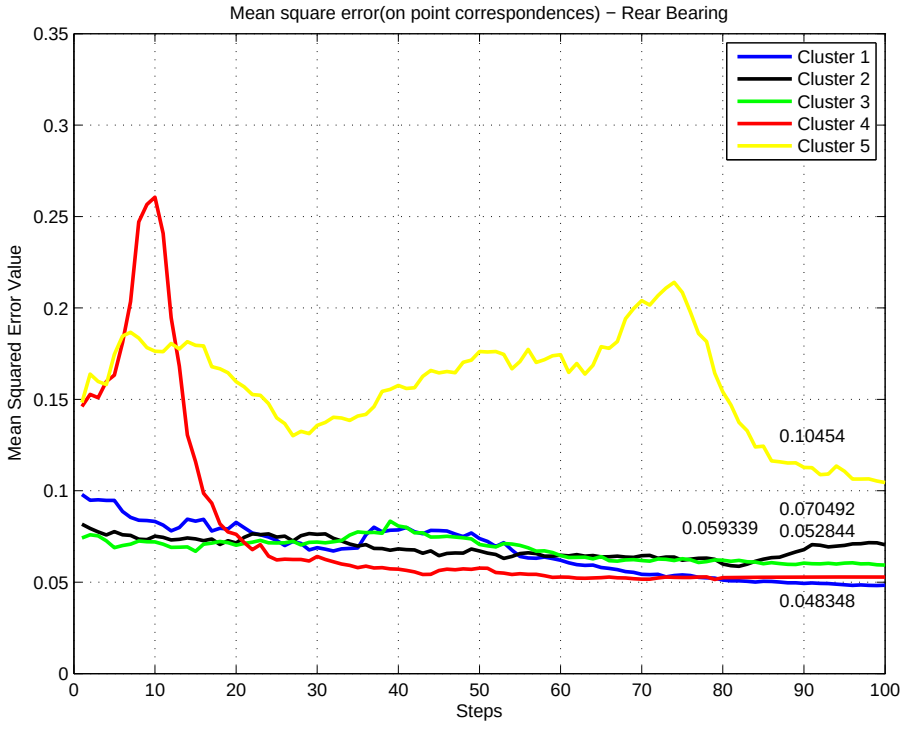


Figure 5.42: *Rear Bearing comparison against assembly parts with similar geometry*

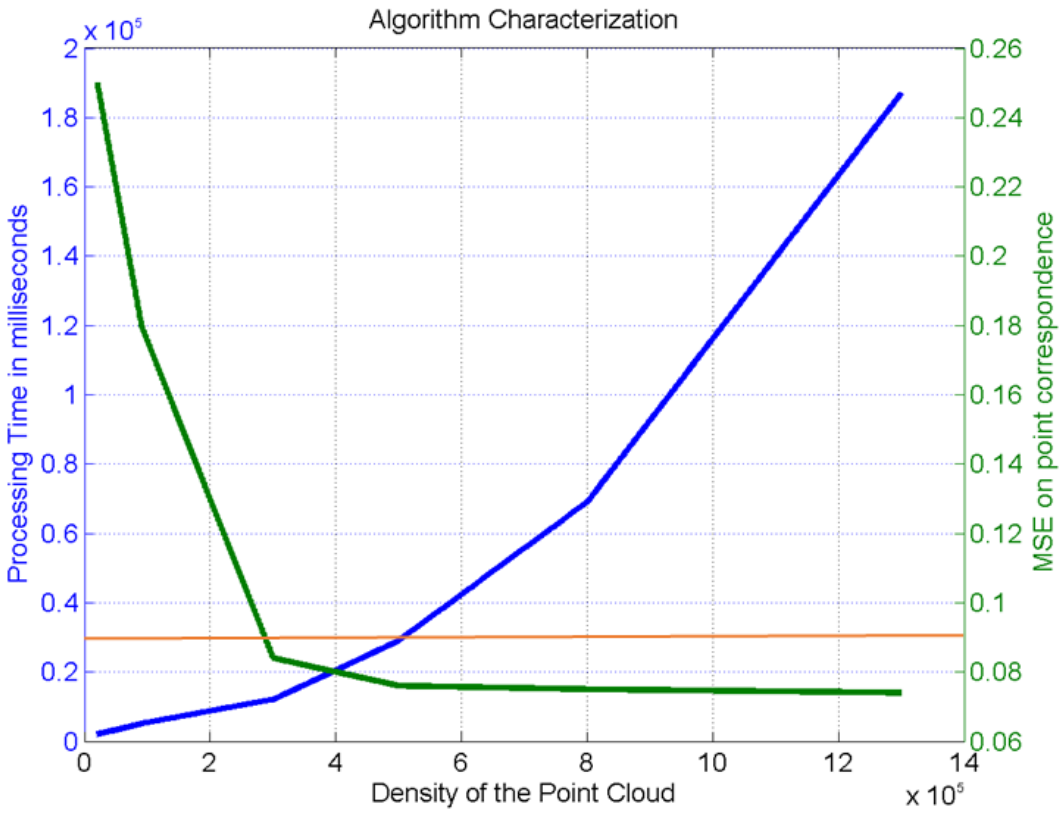


Figure 5.43: *Performance characterization. Region close to the intersection between Processing Time and MSE, and bellow the threshold represents the "sweet spot"*

along alternative paths. This is primarily due to the randomness of human motion, which is difficult to estimate in advance, and the dynamics of the robot implementing such a collision avoidance strategy. Also, these methods increase the computational burden as collision-free paths must be computed in real-time. Velocity-scaling [171] can be used to overcome these issues by operating the robot in a tri-modal state: the robot is in a *clear* (normal operation) state when the human is far away from it. When the distance between them is below a user specified threshold, the robot changes into a *slow* (same path, but reduced speed) state. When the distance is below a second threshold (whose value is lesser than that of the first threshold), the robot changes to a *pause* (stop) state.

Our approach to ensuring safety in the hybrid cell is based on the pre-collision strategy developed in [25]: robot's pauses to move whenever an imminent collision between the human and the robot is detected. This is a simpler bi-modal strategy, in which the robot directly changes from *clear* to *pause* when the estimated distance is below a threshold. This *stop-go* safety approach conforms to the recommendations of the ISO standard 10218 [172, 173].

In order to monitor the human-robot separation, the human model generated by the tracking system (described in the previous section) is augmented by fitting all pairs of neighboring joints with spheres that move as a function of the human's movements in real-time. A *roll-out* strategy is used, in which the robot's trajectory into the near future is pre-computed to create a temporal set of robot's postures for the next few seconds. Now, we verify if any of the postures in this set collides with one of the spheres of the augmented human model. The method is implemented in

Table 5.1: *Exploration structure for Jet Engine assembly*

Level	Assembly parts
10	Cover.
9	Exhaust Turbine.
8	Rear Bearing.
7	Shell.
6	Rear Shaft.
5	Second Compressor.
4	Front Shaft - First Compressor.
3	Shroud.
2	Main Fan.
1	Front Shroud Safety.

a virtual simulation engine developed based on Tundra software.

5.5.2 Replanning and Warning Generation

In this module, we focus on ensuring that the assembly process is progressing as per plan. If a deviation from the plan is detected, the system will automatically generate plans to handle the contingency. We present a proposal for the design of a contingency handling architecture for hybrid assembly cell that has the ability to inexpensively re-plan its sequence in real-time. This design permits a human operator to introduce adjustments or improvements into the assembly sequence in real-time with little delays to the assembly cell output. In order to illustrate our approach, we consider a five part assembly is shown in (Fig. 5.5(a)) along with its preliminary assembly plan and the configuration of the assembly cell (Fig. 1.4(b)).

From the disassembly layers (Table 5.1) generated from the CAD model of the jet engine assembly, we can extract the following assembly sequence: (1) Front

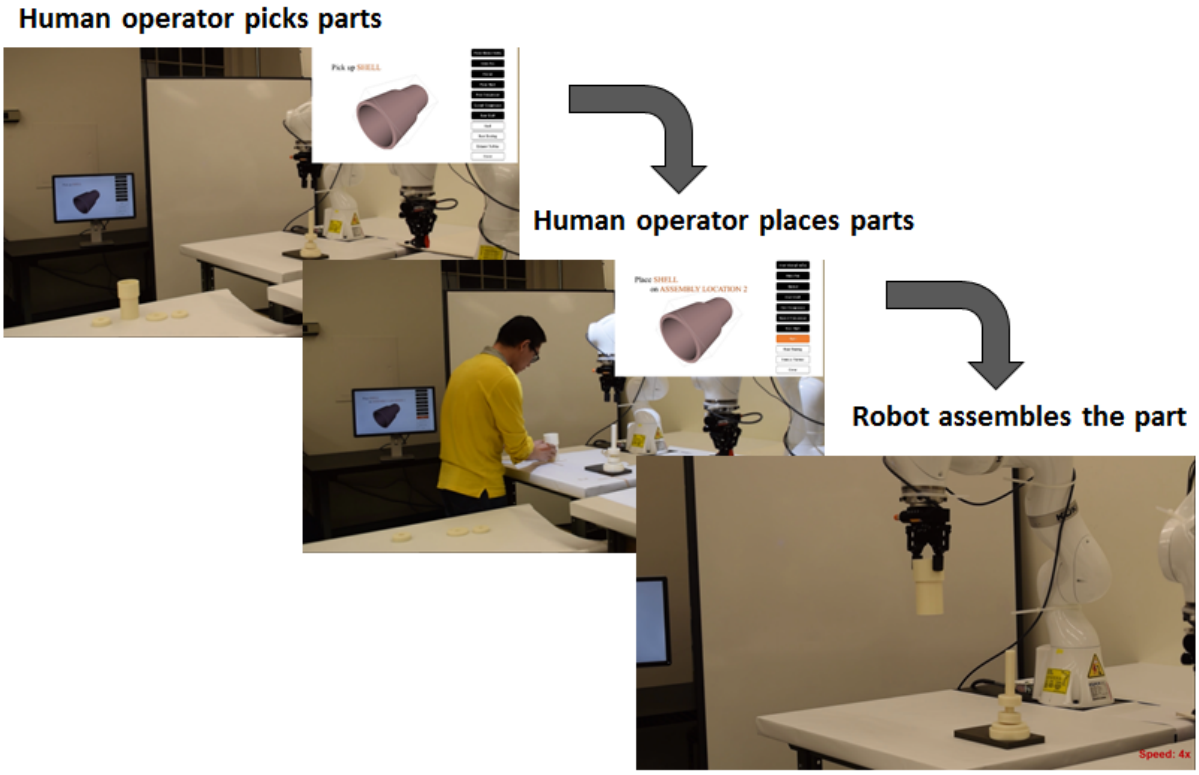


Figure 5.44: *Assembly cell configuration (Human operator picks and places parts and robot assembles the parts.*

Shroud Safety, (2) Main Fan, (3) Shroud, (4) Front Shaft, (5) First Compressor, (6)Second Compressor, (7) Rear Shaft, (8) Shell, (9) Rear Bearing, (10) Exhaust Turbine and (11) Cover. This assembly sequence also defines the plans for the human and the motion planning for the robot.

The following list represents some key elements for the assembly cell state monitoring and contingency planning system.

HRC requires that robots recognize the activity of human operator to maintain synchronization during assembly sequence execution: A reactive property is required to recognize the assembly part during operation, synchronize the assembly task, and later modify the assembly sequence based on the evaluation

of the changes (i.e. human picks up the assembly part, the system recognizes the part and shares the information with the robot to synchronize the assembly operations).

Detecting error in position and orientation of assembly components: An effective collaboration between humans and robots to perform assembly operations requires that if the human does not not present the part at the right position (e.g., outside of the workspace of the robot) or orientation then these errors need to be detected to prevent further errors in the process.

Real-time re-planning: Human operator can introduce modification in the original assembly plan. The assembly plan has to adapt to the changes. This property guaranties the efficient collaboration between human operators and robots. Some of the changes made by the humans will be helpful while others might be simply an error. We will perform real-time re-planning to accommodate deviations from the original plan of they appear to be feasible.

Initially we can describe a scene where the human operator follows the system generated assembly plan with no-errors or requested adjustments. Figure 5.45 shows the complete process of the assembly operation.

An initial assembly plan is generated before the operations begin in the hybrid assembly cell. The plan generates the sequence for the human pick and place operations and the motion plan for the robot assembly operations. A full integration among the assembly plan, human tracking system, and the robot significantly reduces the probability of error introduced by the robot in the cell. We will ignore

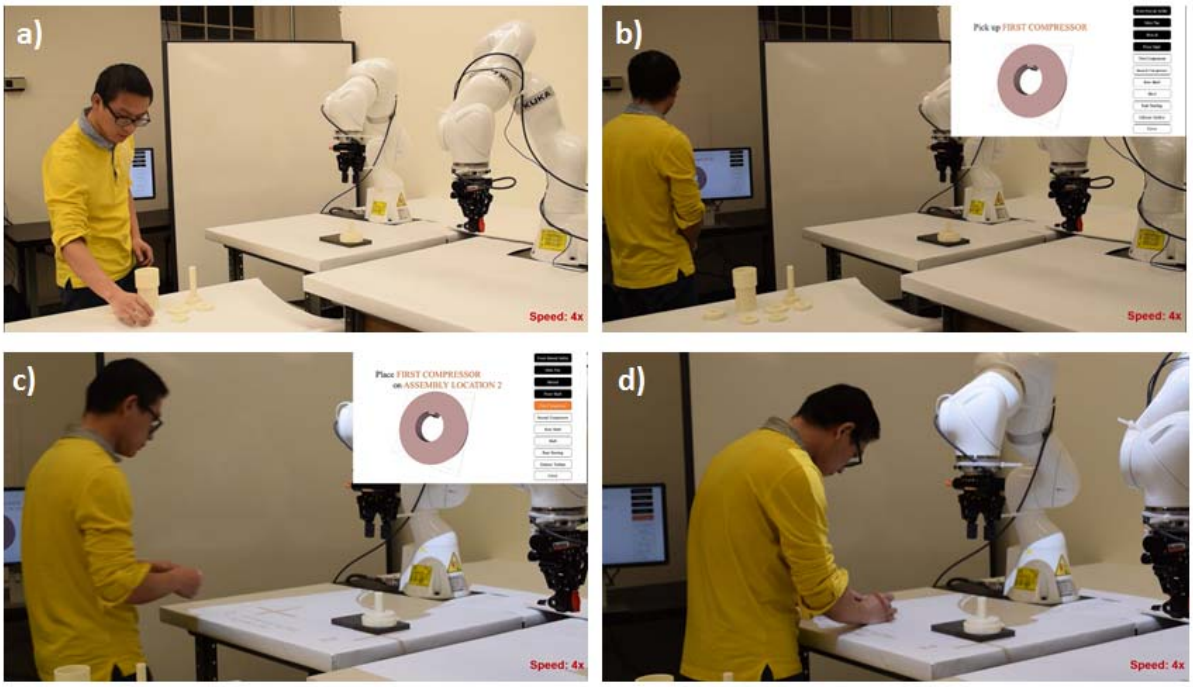


Figure 5.45: *Assembly operations: (a) Human picks up the part. (b) In order to allow synchronization, the system recognizes the part. (c) Human moves the part to the intermediate location. (d) Human places the part in the intermediate location.*

those errors in this work. This configuration leaves the human operator as the only agent with the capacity to introduce errors in the assembly cell. We define deviations in the assembly cell as a modification to the predefined plan. Based on our preliminary work, these modifications can be classified into three main categories:

i Deviations that leads to process errors

ii Deviations that leads to improvements in the assembly speed or output quality

iii Deviation that leads to adjustment in the assembly sequence

Next, we describe the above described deviations in more details.

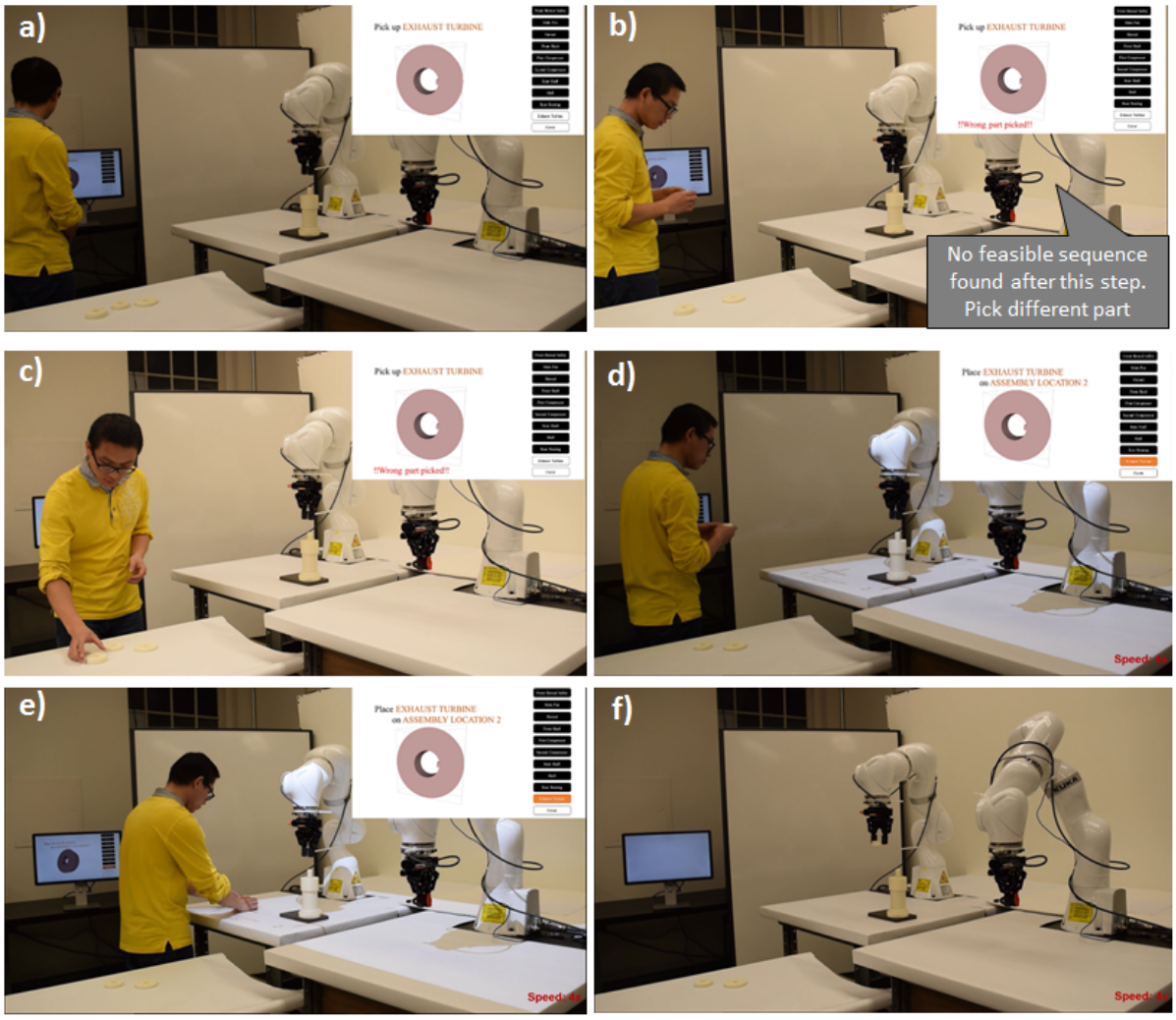


Figure 5.46: *(a) Human picks a part (Compressor); appropriate text annotations are generated as a feedback to the human. (b) Part selected is different from the assembly sequence; after a real-time evaluation, the system doesn't accept the modification in the assembly plan. (c) Human return the part to location 1. (d) Human picks a part (Exhaust Turbine), after real-time evaluation the part is accepted. (e) Human places the part into the robot's workspace. (f) The robot motion planning is executed for the Exhaust Turbine. If the assembly plan is modified (replanning), the robot uses the altered motion plan to pick the part and place it in its target position in the assembly.*

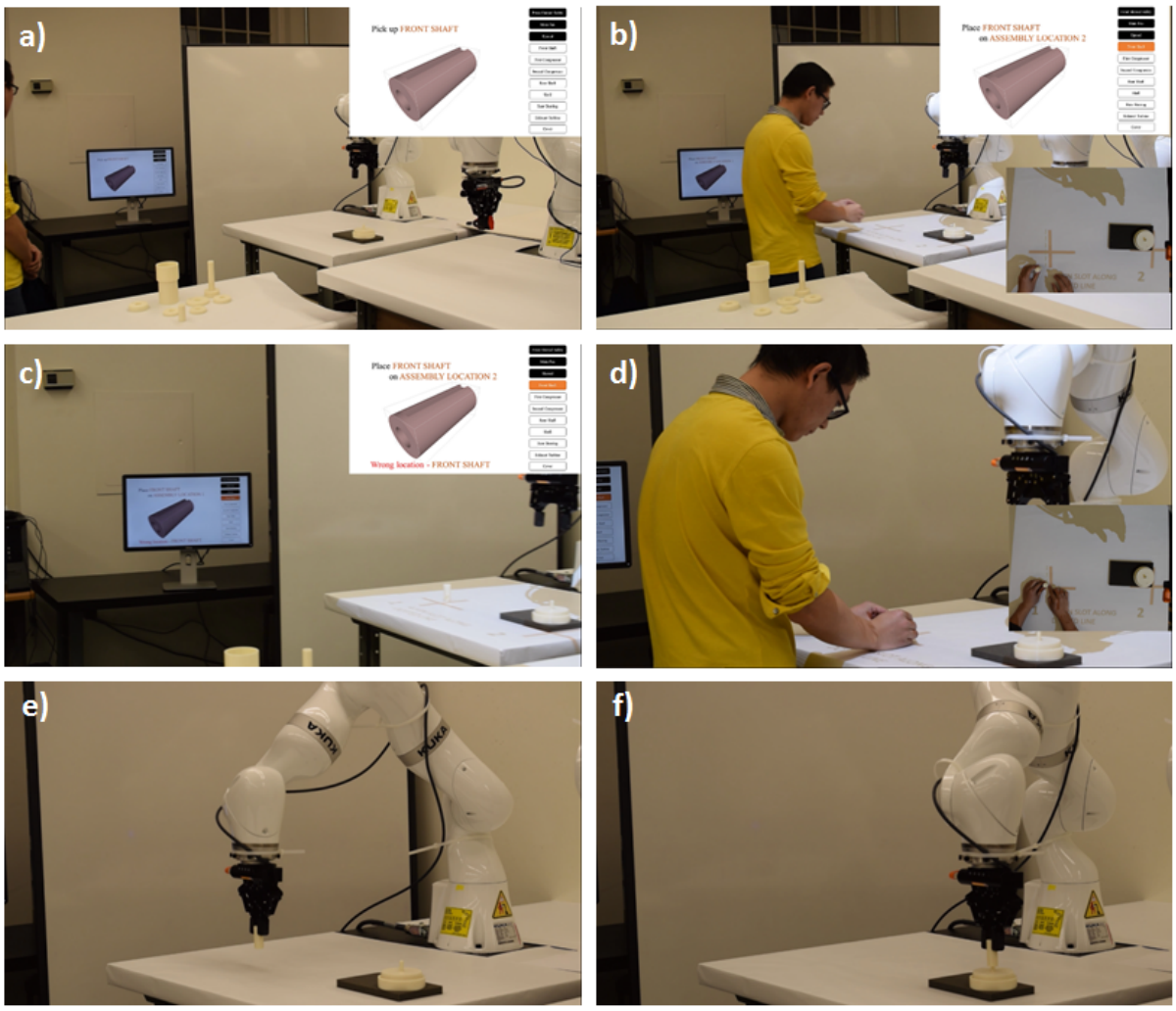


Figure 5.47: *(a) Human picks the Front Shaft part. (b-c) Human places the part in a wrong location. The system detects an inconsistency in the part location and shows a warning message. (d) Human places the part in the correct location. (e-f) Robot picks the Front Shaft part and places it in its target position in the assembly.*

5.5.2.1 Deviations that lead to process errors

Deviations that lead to process errors are modifications introduced by the human operator that cannot generate a feasible assembly plan. These errors can generate an error in the assembly cell in a way that will require costly recovery.

In order to prevent this type of errors, the system has to detect the presence of this modification by the registration of the assembly parts. Once the system has the information about the selected assembly part, it evaluates the error in real-time by propagating the modification in the assembly plan and giving a multi-modal feedback (e.g., text, visual and audible annotations).

We have hand coded several examples to illustrate the deviation described above. Following the assembly plan in our example and after placing the Rear-Bearing, the next part to be assembled is 'Exhaust Turbine'.

Rather than following the assembly sequence, the human operator can decide to use a different sequence. For example, the human picks the 'Compressor' part instead of 'Exhaust Turbine' as shown in Fig. 5.46(a). In order to find a feasible plan, the new assembly sequence with 'Compressor' as a second step is evaluated in real-time. Using the exploration matrix the system determines that there is no possibility to find a feasible assembly sequence following this step. Therefore, the system raises an alarm and generates appropriate feedback using text annotations. This forces the human operator to rely on the predefined assembly sequence.

5.5.2.2 Deviations that leads to improvement

Every single modification to the master assembly plan is detected and evaluated in real-time. The initial assembly plan is one of the many feasible plans that can be found. A modification in the assembly plan that generates another valid feasible plan classifies as an improvement. These modifications are accepted and give the ability and authority to the human operators to use their experience in order to produce better plans. This process helps the system to evolve and adapt quickly using the contributions made by the human agent. Following the assembly sequence, the next part to be assembled is 'Front Shaft'.

The human operator decides based on his/her previous experience that placing the 'First Compressor' next will improve the performance of the assembly process. The part 'First Compressor' is selected and the step is evaluated in real-time. The system discovers that the changes made in the predefined assembly sequence can also generate a feasible assembly sequence. Therefore the step is accepted and human is prompted to continue with the assembly operation. The updated assembly sequence becomes:

Feasible assembly sequence : (1) *Front Shroud Safety*, (2) *Main Fan*, (3) *Shroud*, (4) *First Compressor*, (5) *Front Shaft*, (6) *Second Compressor*, (7) *Rear Shaft*, (8) *Shell*, (9) *Rear Bearing*, (10) *Exhaust Turbine and* (11) *Cover.*

The most important feature of the framework is that the hybrid assembly cell not only accepts the modification in the assembly sequence, but also adapts its configuration in order to complete the assembly process.

Using the new assembly sequence the system recomputed the robot motion planning in real-time to perform the assembly tasks. Hence, the robot knows the new pose of the expected part.

5.5.2.3 Deviations that leads to adjustment

Adjustments in the assembly process may occur when the assembly cell can easily recover from the error introduced by the human by requesting additional interaction in order to fix it. Assuming that the human operator is following the predefined assembly sequence, the next assembly part to be assembled is 'Front Shaft' (Fig. 5.47(a)). The system recognizes the assembly part and validates the step. Therefore the part can be moved and placed in the intermediate location.

Another common mistake in assembly part placement is the wrong pose (rotational and translational transformation that diverges from the required pose) as shown in Fig. 5.47(b−c). Two strategies can be found to solve this issue: a) robot recognizes the new pose and recomputes its motion plan in order to complete the assembly of the part or b) human is informed by the system about the mistake and is prompted to correct it. This work follows the latter strategy.

The system verifies the poses of the assembly parts in the intermediate location in real-time and forces the human operator to place the part in the right location in order to resume the assembly process. Once the assembly part is located in the right position and orientation (Fig. 5.47(d)), the assembly process resumes (Fig. 5.47(e)−5.47(f)).

5.6 Summary

This chapter introduces a novel approach for an automated monitoring of the entire assembly cell that supports safe and efficient human-robot collaboration during assembly operations. The real-time human tracking, human-robot collision detection, and part state monitoring allow the detection of deviations from plans instantaneously and trigger the re-planning to handle these contingencies in order to minimize disruptions on the assembly cell operations. While the assembly operations are performed by human and robot, the system send constantly feedback to the human operator about the performed tasks. This constant feedback in the form of 3D animations, text and audio help to reduce the training time and eliminating the possibility of assembly errors. Moreover, the discrete part monitoring allow the Assembly cell detect and identify the assembly part in at least one of the control points.

Chapter 6

Conclusions

This chapter presents the intellectual contributions and anticipated benefits from the work reported in this dissertation.

6.1 Intellectual Contributions

The research work described in previous chapters broadly aims toward building development of algorithms to support realization of hybrid assembly cells. The main expected contributions from the reported dissertation are the following:

6.1.1 Automated planning for hybrid assembly cell operation

This dissertation introduces a new motion planning and part interaction clusters-based assembly sequence planning approach to improve the generation of feasible assembly sequences for non-deformable parts. This new planning approach uses geometric and kinematic constraints in generate assembly plans. The generation of multiple clusters and the improved motion planning technique helps the algorithm manage assemblies consisting of a large and diverse number of parts as well as address the complexity of the assembly due to complex motions needed to perform the assembly operations. The framework generates feasible assembly sequences directly from 3D assembly models. The algorithm guarantees the generation of at least one

feasible assembly plan if a valid assembly sequence exists.

6.1.2 Monitoring the state of the human operator in the assembly cell to ensure safe operation

This dissertation demonstrates an automated approach for human-robot real-time monitoring in assembly an assembly cell. The sensor integration and the real-time extended world enable the development of an HRC system that can be utilized to prevent collisions between humans and robots when they work together in an assembly cell. The system gives the robot estimated human state. This can be be used to prevent collision between human and robot. The contributions of the HRC framework are the following:

- An N-Kinect human tracking framework that generates a 3D model of human's movements in real-time.
- Full assembly cell workspace monitoring and tracking coverage by fusing the information from multiple sensors,
- A roll-out pre-collision strategy that allows the system to compute future positions of the robot to evaluate possible human collision.
- A low cost human motion tracking system with sufficient accuracy (approximately 2 cm) to enable hybrid assembly cells.
- A design that does not require the assembly cell human operator to wear any special suit or additional device.

6.1.3 Monitoring the state of the assembly cell and contingency planning

This dissertation introduces a novel design approach of a framework for hybrid cells that support safe and efficient human-robot collaboration during assembly operations. The dissertation presents concrete physical experiments to show how safety can be ensured in the cell, how error-free operation can be achieved in the cell, and how the robot can re-plan in response to unpredictable human behaviors and modify its motion according to new plans. Our framework can be extended to hybrid cells that support collaboration among multiple humans and robots. A simple chassis assembly was used in the experiments. Our approach can be easily extended to more complex assemblies.

Additionally, this dissertation presents a design framework to automatically generate multimodal instructions for complex assembly operations performed by humans. The generated instructions are easy-to-follow, which thereby reduces learning time and eliminates the possibility of assembly errors. The system's ability to automatically translate assembly plans into instructions enables a significant reduction in the time taken to generate instructions and update them in response to design changes. In the current design of animations, parts move by themselves from initial to final postures. As the animation generation is grounded in motion planning, this issue can be addressed by incorporating human models, with increasingly complex degrees of freedom, into the framework. This results in more realistic animations that utilize a human model to show how to lift and move the parts/tools during

assembly.

6.2 Anticipated Benefits

This dissertation addresses the key issues of automated planning, human state monitoring for safety, and contingency planning to enable hybrid assembly cell operations. As discussed in chapters 1 and 2 small and medium size manufacturing (SMM) companies need low cost and flexible automation technologies to allow them to be cost competitive and responsive to frequent product changes and low volume production.

Low-cost hybrid assembly cells that allow production schedule compression and assembly cell production flexibility in small and medium size manufacturing operations will contribute to the global competitiveness of manufacturing companies in high-wage countries. Short-volume production and batch production are expected to benefit from the work reported in this dissertation.

6.3 Future Work

This dissertation focused on one-human one-robot cells. Many real-world complex assemblies will require collaboration among multiple people and robots. This will require the planning process to generate plans that will include tasks that can be performed concurrently. This will require an analysis of task independence and dependencies. The presence of multiple agents in the cells will also pose new challenges for system state monitoring and ensuring safety.

Tasks investigated in this dissertation did not require physical collaboration between humans and robots. Some tasks may require physical collaboration between humans and robots. In such cases, the plan will need to account for force and motion interactions between humans and robots. This will require modeling the capabilities of hybrid agents using them in planning.

In this dissertation, we developed an assembly planning algorithm that does not account for the human safety during the generation of the assembly sequence. The planning algorithm can be extended to explicitly consider safety considerations.

Appendices

Appendix A

Test-bed Overview

In order to validate the formulations and methodologies reported in this dissertation, a test-bed system comprising a robotic arm, multiple sensors, tracking system and a 3D virtual reality system were developed for this work. This appendix presents the system's architecture, the robot setup, the sensors' setup and the overall communications system architecture.

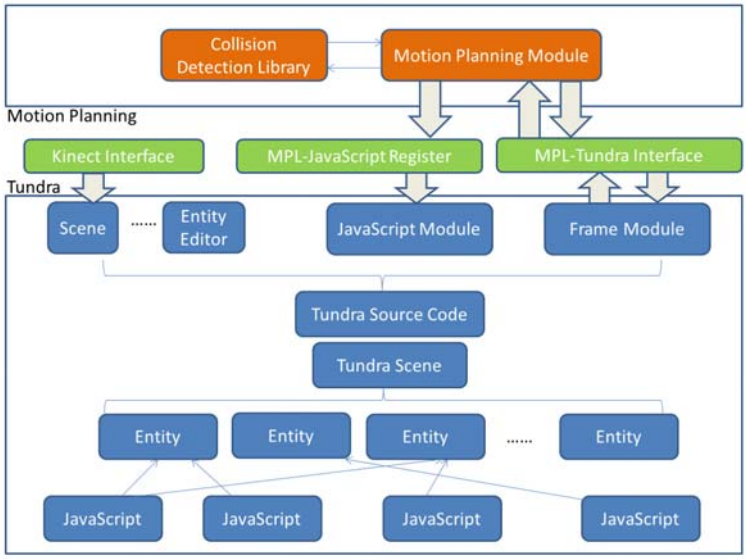


Figure A.1: *Virtual environment architecture*

An open source game engine is used as a 3D platform to translate the real environment into a very structured virtual environment. Additional systems were developed to obtain full functionality (fig. A.1).

A.1 Robotic System

The system was tested with the Lab-Volt Model 5150 robotic manipulator and with a KUKA collaborative robot. The Lab-Volt Model 5150 is a typical laboratory scale robot with a payload of 1kg that performs assembly operations. The KUKA collaborative robotic manipulator is a robot designed to work in close proximity to humans in a collaborative environments.

The Lab-Volt Robot is designed with a tool-tip and five step-motor actuators that allow it five degrees of freedom. The Robot can use its joints simultaneously to perform a pre-programmed move sequence.

A.2 Robot motion Planning

In order to compute the sequences of motion of the robotic arm, the systems use a STL based scene representation of the original setup to find a collision-free-path and set of motion configurations from the initial configuration to the final configuration A.2, A.4.

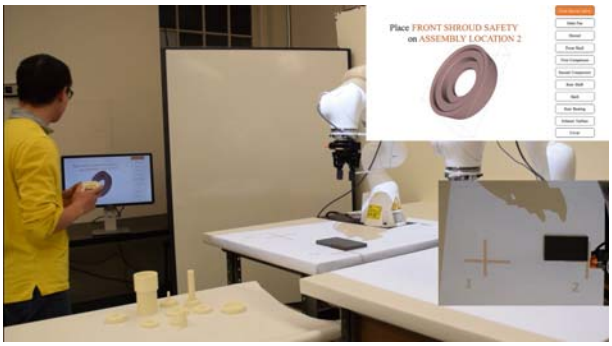


Figure A.2: *Experimental Setup*

The framework represents the assembly parts via STL format to generated the assembly sequence (fig. A.2)

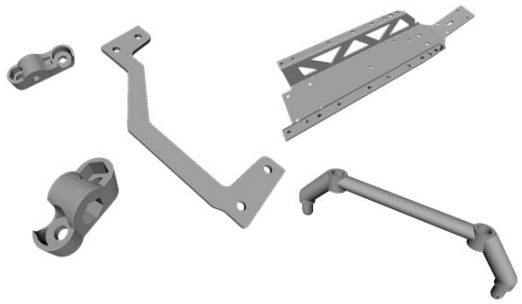


Figure A.3: *STL representation of the assembly parts*

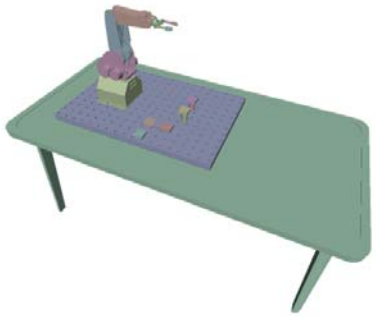


Figure A.4: *STL based experimental Setup*

A.3 Safety in Human Robot Collaboration in Assembly Operations

A complete 3D representation of the hybrid assembly cell environment permits the system to predict any possible collision between humans and robots (fig. A.6). The position of the robot during the assembly sequence is known by the system given that the robot motion plan is pre-computed before the real-time HRC start. The motion plan controls the action of the robot while the tracking system monitors and computes the position of the human in real-time. Visual and auditory feedback

about the relative position of the robot and the human operator is provided to the operator by the system during the assembly sequence. This feedback allows the operator ongoing situational awareness of the assembly cell environment (fig. A.5). Additionally, the system is designed to automatically stop the robot when an imminent collision is predicted. After such stops the system will not permit robot motion to resume until it detects that the human operator is located in a safe place.

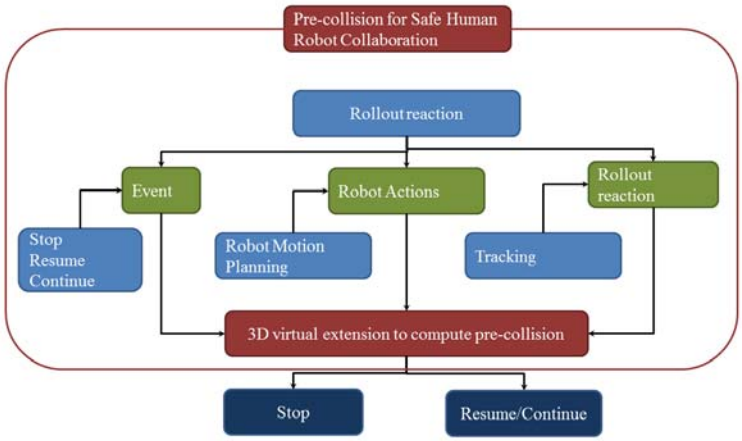


Figure A.5: *Safety architecture for Human Robot Collaboration in Assembly Applications*

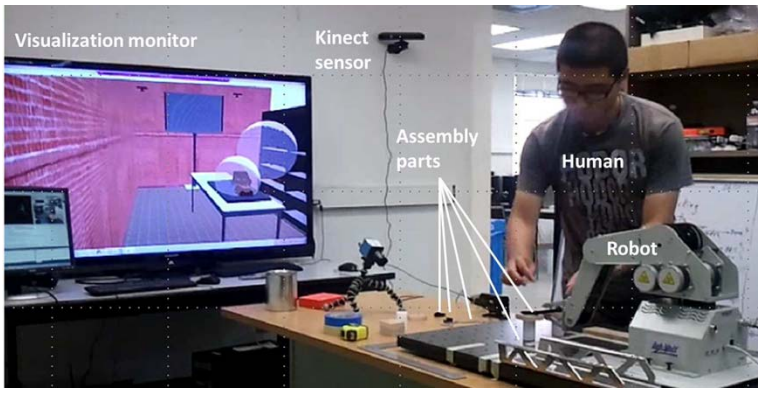


Figure A.6: *3D virtual representation of the assembly cell*

A.4 Communication architecture

The system's data processing takes place in four separate computers each connected as a client with the Kinects sensors. These computers send UDP packages with a frequency of 30Hz to a server that processes all the data. The communication architecture is shown in fig. A.7. All the on-line fusion processing is executed on the server. It was necessary to separate the sensing acquisition and processing onto multiple PCs in order to guaranty real-time execution and overcome the Kinect SDK's limitation with processing more than one skeleton per computer.

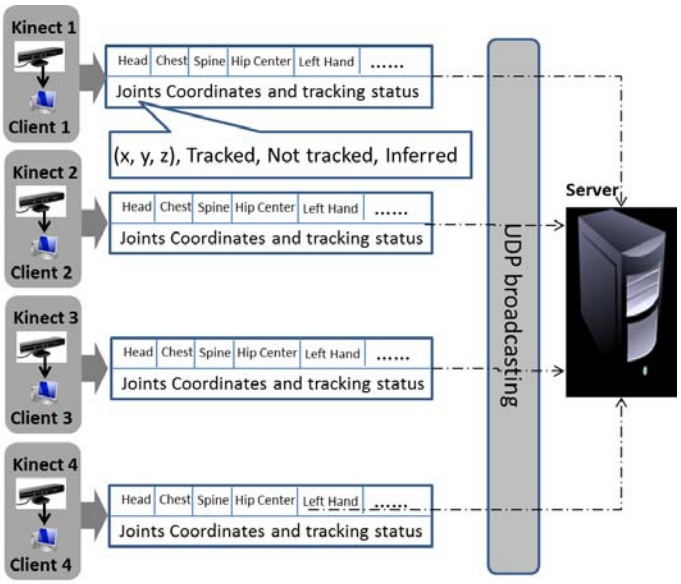


Figure A.7: *Communication architecture used to integrate the human model data from multiple kinect sensors.*

Bibliography

[1] Nortel, "Assembly of telecommunication equipments," in *Available: http://www.nortel-us.com/category/news-releases/*, 2010.

[2] P. Test, "Welding process," in *Available: http://www.primetest.com/robotic-systems.html*, 2009.

[3] Esson, "New robotic standards to improve safety and productivity," in *ANSI/RIA R15.06-2012 standard references ISO 10218-1*, 2013.

[4] K. Mitobe, T. Kaiga, T. Yukawa, and Y. N, "Development of a motion capture system for a hand using a magnetic three dimensional position sensor," in *ACM SIGGRAPH 2006*, 2006.

[5] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," in *Technical Report TR 98–11, Computer Science Dept., Iowa State University*, 1998.

[6] J. Kuffner and S. LaValle, "Rrt-connect: An ecient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, 2000, pp. 995–1001.

[7] S. LaValle and J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2012, pp. 293–308.

[8] S. Balakirsky, Z. Kootbally, C. Schlenoff, T. Kramer, and S. K. Gupta, "An industrial robotic knowledge representation for kit building applications," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 1365–1370.

[9] A. G. Banerjee, A. Barnes, K. N. Kaipa, J. Liu, S. Shriyam, N. Shah, and S. K. Gupta, "An ontology to enable optimized task partitioning in human-robot collaboration for warehouse kitting operations," in *Proc. SPIE, Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX, 94940H*, 2015.

[10] K. N. Kaipa, S. S. Thevendria-Karthic, S. Shriyam, A. M. Kabir, J. D. Langsfeld, and S. K. Gupta, "Resolving automated perception system failures in bin-picking tasks using assistance from remote human operators." in *Proc. of IEEE International Conference on Automation Science and Engineering*, August 2015.

[11] K. N. Kaipa, S. Shriyam, N. B. Kumbla, and S. K. Gupta, "Automated plan generation for robotic singulation from mixed bins," in *IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing*, 2015.

[12] K. N. Kaipa, N. B. Kumbla, and S. K. Gupta, "Characterizing performance of sensorless fine positioning moves in the presence of grasping position uncertainty," in *IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing*, 2015.

[13] K. N. Kaipa, A. S. Kankanhalli-Nagendra, and S. K. Gupta, "Toward estimating task execution confidence for robotic bin-picking applications," in *AAAI Fall Symposium: Self-Confidence in Autonomous Systems*, 2015.

[14] K. N. Kaipa, S. Shriyam, N. B. Kumbla, and G. S. K., "Resolving occlusions through simple extraction motions in robotic bin-picking," in *ASMEs 11th Manufacturing Science and Engineering Conference*, June 2016.

[15] V. Shivashankar, K. N. Kaipa, D. S. Nau, and S. K. Gupta, "Towards integrating hierarchical goal networks and motion planners to support planning for human robot collaboration in assembly cells," in *2014 AAAI Fall Symposium Series*, 2014.

[16] J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, and S. K. Gupta, "Online learning of part deformation models in robotic cleaning of compliant objects," in *ASMEs 11th Manufacturing Science and Engineering Conference*, June 2016.

[17] A. M. Kabir, J. D. Langsfeld, C. Zhuang, K. N. Kaipa, and S. K. Gupta, "Automated learning of operation parameters for robotic cleaning by mechanical scrubbing," in *ASMEs 11th Manufacturing Science and Engineering Conference*, June 2016.

[18] S. Lee, B. Choi, and R. Suare, "Frontiers of assembly and manufacturing: Selected papers from isam-09," in *Springer*, 2009.

[19] L. Curry and R. Feldman, "Manufacturing systems modeling and analysis," in *Springer-Verlag*, 2009.

[20] Baxter. (2010) Rethink robotics. [Online]. Available: http://www.rethinkrobotics.com/products/baxter/

[21] Kuka. (2010) Kuka lbr iv. [Online]. Available: http://www.kukalabs.com/en/medical_robotics/lightweight_robotics/

[22] ABB. (2013) Abb friendly robot for industrial dual arm frida. [Online]. Available: http://www.new.abb.com/products/robotics/yumi

[23] C. Morato, K. Kaipa, and S. K. Gupta, "Assembly sequence planning by using multiple random trees based motion planning," in *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2012, pp. 1461–1471.

[24] C. Morato, K. N. Kaipa, and S. K. Gupta, "Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters," *Comput. Aided Des.*, vol. 45, no. 11, pp. 1349–1364, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.cad.2013.06.005

[25] C. Morato, K. Kaipa, B. Zhao, and S. K. Gupta, "Safe human robot interaction by using exteroceptive sensing based human modeling," in *ASME Computers and Information in Engineering Conference, Portland, OR, August*, 2013.

[26] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward safe human robot collaboration by using multiple kinects based real-time human tracking,"

Journal of Computing and Information Science in Engineering, vol. 14, no. 1, p. 011006, 2014.

[27] K. Kaipa, C. Morato, B. Zhao, and S. K. Gupta, "Instruction generation for assembly operations performed by humans," in *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2012, pp. 1121–1130.

[28] K. N. Kaipa, C. Morato, J. Liu, and S. K. Gupta, "Human-robot collaboration for bin-picking tasks to support low-volume assemblies," in *Human-Robot Collaboration for Industrial Manufacturing Workshop, held at Robotics: Science and Systems Conference (RSS 2014)*, 2014.

[29] C. Morato, K. Kaipa, J. Liu, and S. Gupta, "A framework for hybrid cells that support safe and efficient human-robot collaboration in assembly operations," in *ASME Computers and Information Engineering Conference*, 2014.

[30] T. De Fazio and D. Whitney, "Simplified generation of all mechanical assembly sequences," in *IEEE Transactions on Robotics and Automation*, no. 3, 1987, pp. 640–658.

[31] L. Homem De Mello and A. Sanderson, "A correct and complete algorithm for the generation of mechanical assembly sequences," in *IEEE Transactions on Robotics and Automation*, no. 7, 1991, pp. 228–240.

[32] R. Hoffman, "A common sense approach to assembly sequence planning," in *Computer Aided Mechanical Assembly Planning.* Boston, MA, USA: Kluwer Academic, 1991, pp. 289–314.

[33] S. Krishnan and A. Sanderson, "Path planning algorithms for assembly sequence planning," in *International Symposium on Intelligent Robotics*, 1991, pp. 428–439.

[34] S. Lee and Y. Shin, "Assembly planning based on geometric reasoning," in *Proceedings of Computers and Graphics*, vol. 14, no. 2, 1990, pp. 237–250.

[35] R. H. Wilson, "On geometric assembly planning," Ph.D. dissertation, stanford university, 1992.

[36] W. Zhang, T. Freiheit, and H. Yang, "Dynamic scheduling in fexible assembly system based on timed petri nets model," in *Robot. Comput.-Integr. Manuf.*, vol. 21, no. 6, 2005, pp. 550–558.

[37] Q. Su, "Applying case based reasoning in assembly sequence planning," *International Journal of Production Research*, vol. 45, no. 1, pp. 29–47, 2007.

[38] ——, "A hierarchical approach on assembly sequence planning and optimal sequences analyzing," *Robot Comput-Integration Manufacturing*, vol. 1, no. 25, pp. 224–234, 2009.

[39] W. Chen, Y. Hsu, L. Hsieh, and P. Tai, "A systematic optimization approach for assembly sequence planning using taguchi method, doe, and bpnn," *Experts Systems with Applications*, vol. 1, no. 37, pp. 716–726, 2010.

[40] L. Kavraki, J. Latombe, and R. Wilson, "On the complexity of assembly partitioning," in *Information Processing Letters*, no. 48, 1993, pp. 229–235.

[41] L. Kavraki and M. Kolountzakis, "Partitioning a planar assembly into two connected parts is np-complete," in *Information Processing Letters*, vol. 3, no. 55, 1995, pp. 159–165.

[42] R. Wilson, L. Kavraki, J. Latombe, and T. Lozano-Perez, "Two-handed assembly sequencing," *International Journal of Robotics Research*, vol. 4, no. 14, pp. 335–350, 1996.

[43] M. Goldwasser, J. Latombe, and R. Motwani, "Complexity measures for assembly sequences," in *IEEE International Conference on Robotics and Automation*, 1996, pp. 1851–1857.

[44] D. Baldwin, T. Abell, T. De Fazio, and D. Whitney, "An integrated computer aid for generating and evaluating assembly sequences for mechanical products," in *IEEE Transactions on Robotics and Automation*, no. 7, 2012, pp. 78–94.

[45] A. Lambert, "Optimal disassembly of complex products," *International Journal of Production Research*, vol. 9, no. 35, pp. 2509–2523, 1997.

[46] ——, "Linear programming in disassembly/clustering sequence generation," in *Computers and Industrial Engineering*, no. 36, 1999, pp. 723–738.

[47] ——, "Determining optimum disassembly sequences in electronic equipment," *Computers and Industrial Engineering*, no. 43, pp. 553–575, 2002.

[48] ——, "Exact methods in optimum disassembly sequence search for problems subject to sequence dependent costs," *Omega - Reverse production systems*, vol. 34, no. 6, pp. 538–549, 2006.

[49] A. Bourjault, "Contribution a une approche methodologique de lassemblage automatise: Elaboration automatique des sequences operatoires," in *PhD thesis, Universite de Franche-Compte*, 1984.

[50] R. H. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly," *Artificial Intelligence*, vol. 71, no. 2, pp. 371 – 396, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370294900485

[51] J. Dong and G. Arndt, "A review of current research on disassembly sequence generation and computer aided design for disassembly," in *Mechanical Engineering Part B Eng Manuf 217*, 2003.

[52] X. Niu, H. Ding, and X. Y, "A hierarchical approach to generating precedence graph for assembly planning," in *International Journal of Machine Tools and Manufacturing*, 2003.

[53] I. Aguinaga, D. Borro, and L. Matey, "Parallel rrt-based path planning for selective disassembly planning," *International Journal of Advanced Manufacture Technology*, no. 36, pp. 1221–1233, 2008.

[54] S. K. Gupta, W. C. Regli, D. Das, and D. S. Nau, "Automated manufacturability analysis: A survey," *Research in Engineering Design*, vol. 9, no. 3, pp. 168–190, 1997. [Online]. Available: http://dx.doi.org/10.1007/BF01596601

[55] X. Zha, S. Lim, and S. Fok, "Integrated intelligent design and assembly planning: A survey," *The International Journal of Advanced Manufacturing Technology*, vol. 14, no. 9, pp. 664–685, 1998. [Online]. Available: http://dx.doi.org/10.1007/BF01192287

[56] B. OShea, S. Grewal, and H. Kaebernick, "State of the art literature survey on disassembly planning," in *Concurrent Engineering: Research and Applications*, 1998, pp. 345–380.

[57] A. Lambert, "Disassembly sequencing: A survey," *International Journal Production Research*, vol. 16, no. 41, pp. 3721–3759, 2003.

[58] P. Jimenez, "Survey on assembly sequencing: A combinatorial and geometrical perspective," *Journal of Intelligent Manufacturing*, no. 23, 2011.

[59] T. C. Woo and D. Dutta, "Automatic disassembly and total ordering in three dimensions," *Journal of Manufacturing Science and Engineering*, vol. 113, no. 2, pp. 207–213, 1991.

[60] H. Srinivasan and R. Gadh, "A geometric algorithm for single selective disassembly using the wave propagation abstraction," *omputer Aided Design*, vol. 8, no. 30, pp. 603–613, 1998.

[61] S. Chen, J. Oliver, S. Chou, and C. L, "Parallel disassembly by onion peeling," in *Journal of Mechanical Design*, 1997.

[62] A. Selva, R. Castro, and V. Frias, "Design of disassembly sequences using search strategies. application of ida in state diagrams," *International Journal of Production Research*, no. 49, pp. 3395–3403, 2010.

[63] R. Chen, K. Lu, and S. Yu, "A hybrid genetic algorithm approach on multi-objective of assembly planning problem," in *Engineering Applications of Artificial Intelligence*, vol. 5, no. 15, 2002, pp. 447–457.

[64] B. Romney, C. Godard, M. Goldwasser, and G. Ramkumar, "An efficient system for geometric assembly sequence generation and evaluation," in *ASME International Computers in Engineering Conference*, 1995, pp. 699–712.

[65] S. Chen, "Assembly planning–a genetic approach," in *ASME Proceedings of Design Engineering Technical Conferences,DAC-5798*, 1999.

[66] P. Iacob R, Mitrouchev and L. J, "Assembly simulation incorporating component mobility modeling based on functional surfaces," *International Journal Interactive Design Manufacturing*, pp. 119–132, 2012.

[67] X. Yuan, "An interactive approach of assembly planning," in *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 2002.

[68] S. Chen and Y. Liu, "A multi-level genetic assembly planner," in *ASME Proceedings of Design Engineering Technical Conferences, Baltimore, MD, DAC-14246*. Marcel Dekker Press, 2000.

[69] P. De Lit, P. Latinne, B. Rekiek, and A. Delchambre, "Assembly planning with an ordering genetic algorithm," *International Journal of Production Research*, vol. 39, no. 16, pp. 3623–3640, 2001.

[70] Q. Guan, J. Liu, and Y. Zhong, "A concurrent hierarchical evolution approach to assembly process planning," *International Journal of Production Research*, vol. 40, no. 14, pp. 3357–3374, 2002.

[71] W. Hui, X. Dong, and D. Guanghong, "A genetic algorithm for product disassembly sequence planning," *Neurocomputing*, vol. 71, no. 1315, pp. 2720 – 2726, 2008, artificial Neural Networks (ICANN 2006) / Engineering of Intelligent Systems (ICEIS 2006). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523120800235X

[72] H. Tseng, W. Wang, and H. Shih, "Using memetic algorithms with guided local search to solve assembly sequence planning," in *Expert Systems with Applications*, 2007.

[73] C. Sinanoglu and H. R. Börklü, "An assembly sequence-planning system for mechanical parts using neural network," *Assembly automation*, vol. 25, no. 1, pp. 38–52, 2005.

[74] W.-C. Chen, P.-H. Tai, W.-J. Deng, and L.-F. Hsieh, "A three-stage integrated approach for assembly sequence planning using neural networks," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1777 – 1786, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417407000541

[75] J. P. Wang and V. Allanda, "Serviceability evaluation using weighted fuzzy attributes," in *Engineering Design Automation*, 1998.

[76] J. Wang and V. Allanda, "Hierarchical fuzzy neural network-based serviceability evaluation," in *International Journal of Agile Management Systems*, 2000.

[77] J. Duflou, G. Seliger, S. Kara, Y. Umeda, A. Ometto, and B. Willems, "Efficiency and feasibility of product disassembly: a case based study," in *CIRP Ann Manuf Technol*, 2008, pp. 583–600.

[78] B. Scholz-Reiter and H. Scharke, "Implementation and testing of a reactive disassembly planner," in *4th CIRP International Seminar on Life Cycle Engineering*, 1997.

[79] A. Bagchi and A. Mahanti, "Admissible heuristic search in and/or graphs," in *Theoretical Computer Science*, 1983.

[80] A. Motavalli and A. Islam, "Multi-criteria assembly sequencing," in *Computers and Industrial Engineering*, 1997.

[81] S. Gupta, C. Paredis, R. Sinha, C. Wang, and P. Brown, "An intelligent environment for simulating mechanical assembly operation," in *ASME Design for Manufacturing Conference*, Atlanta, GA, USA, 1998.

[82] S. K. Gupta, C. J. Paredis, R. Sinha, and P. F. Brown, "Intelligent assembly modeling and simulation," *Assembly Automation*, vol. 21, no. 3, pp. 215–235, 2001.

[83] U. Thomas and F. Wahl, "Assembly planning and task planningtwo prerequisites for automated robot programming," in *Robotic Systems for Handling and Assembly*, no. 67, 2010, pp. 333–354.

[84] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in highdimensional configuration spaces," in *IEEE Transactions on Robotics and Automation*, vol. 4, no. 12, 1996, pp. 566–580.

[85] S. Sundaram, I. Remmler, and N. Amato, "Disassembly sequencing using a motion planning approach," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, 2001, pp. 1475–1480 vol.2.

[86] D. Le, J. Cortes, and T. Simeon, "A path planning approach to (dis)assembly sequencing," in *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, Aug 2009, pp. 286–291.

[87] J. Xiao and X. Ji, "Automatic generation of high-level contact state space," *International Journal of Robotics Research*, vol. 7, no. 20, pp. 584–606, 2001.

[88] L. Jaillet, J. Cortes, and T. Simeon, "Transition-based rrt for path planning in continuous cost spaces," in *IEEE/RSJ International Conference of Intelligent Robotic Systems*, 2008, pp. 22–26.

[89] J. Cortes, L. Jaillet, and T. Simeon, "Disassembly path planning for complex articulated objects," *Robotics, IEEE Transactions on*, vol. 24, no. 2, pp. 475–481, April 2008.

[90] G. Dini and M. Santochi, "Automated sequencing and subassembly detection in assembly planning," {*CIRP*} *Annals - Manufacturing Technology*, vol. 41, no. 1, pp. 1 – 4, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007850607611408

[91] Y. Zhang, J. Ni, Z. Lin, and X. Lai, "Automatic sequence planning by subassembly detection for automobile body assembly," in *ASME 2002 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2002, pp. 485–490.

[92] W. Wang, G. Chen, Z. Lin, and X. Lai, "Automated hierarchical assembly system construction in automobile body assembly planning," *Journal of Mechanical Design*, vol. 127, no. 2, pp. 347–351, 2005.

[93] Y. Xing, G. Chen, X. Lai, S. Jin, and J. Zhou, "Assembly sequence planning of automobile body components based on liaison graph," *Assembly Automation*, vol. 27, no. 2, pp. 157–164, 2007.

[94] L. Balan and G. Bone, "Real-time 3d collision avoidance method for safe human and robot coexistence," in *IEEE/RSJ International Conference onIntelligent Robots and Systems*, Oct 2006, pp. 276–282.

[95] F. Flacco and A. De Luca, "Multiple depth/presence sensors: Integration and optimal placement for human/robot coexistence," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3916–3923.

[96] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *IEEE International Conference on Robotics and Automation*, May 2012, pp. 338–345.

[97] S. Kuhn, T. Gecks, and D. Henrich, "Velocity control for safe robot guidance based on fused vision and force/torque data," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Sept 2006, pp. 485–492.

[98] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10514-006-9009-4

[99] N. Najmaei and M. Kermani, "Prediction-based reactive control strategy for human-robot interactions," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3434–3439.

[100] R. Schiavi, A. Bicchi, and F. Flacco, "Integration of active and passive compliance control for safe human-robot coexistence," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 259–264.

[101] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 299–306.

[102] A. Bicchi and G. Tonietti, "Fast and "soft-arm" tactics [robot arm design]," *IEEE Robotics Automation Magazine*, vol. 11, no. 2, pp. 22–33, June 2004.

[103] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3356–3363.

[104] D. Shin, I. Sardellitti, Y.-L. Park, O. Khatib, and M. Cutkosky, "Design and control of a bio-inspired human-friendly robot," *The International Journal of Robotics Research*, 2009. [Online]. Available: http://ijr.sagepub.com/content/early/2009/11/23/0278364909353956.abstract

[105] D. Henrich and T. Gecks, "Multi-camera collision detection between known and unknown objects," in *ACM/IEEE International Conference on Distributed Smart Cameras*, Sept 2008, pp. 1–10.

[106] N. Najmaei, M. Kermani, and M. Al-Lawati, "A new sensory system for modeling and tracking humans within industrial work cells," *Instrumentation and*

Measurement, IEEE Transactions on, vol. 60, no. 4, pp. 1227 –1236, april 2011.

[107] S. D. Field M., Pan Z. and N. F., "Human motion capture sensors and analysis in robotics," in *Industrial Robots: An International Journal*, 2011.

[108] Polhemus. (2011) Liberty electromagnetic tracking technology. [Online]. Available: http://polhemus.com/motiontracking/alltrackers/liberty

[109] A. T. Corporation. (2011) Motionstar magnetic motion capture system. [Online]. Available: http://www.ascension-tech.com/realtime/RTmotionStarTethered.php

[110] Metamotion. (2011) Gypsy 7 motion capture system. [Online]. Available: http://www.metamotion.com/gypsy/gypsymotioncapturesystem.htm

[111] Sarcos, "Sarcos sensuit," in *Available: http://www.sarcos.com/telepic$_s$ensuit.html*, 2012.

[112] X. Inc, "mtx inertial sensor," in *Xsens*, 2011.

[113] X. Yun, E. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 2526–2533.

[114] A. M. Sabatini, "Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation," *Computational*

Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques, pp. 70–100, 2006.

[115] H. Zhou and H. Hu, "Human motion tracking for rehabilitation–a survey," *Biomedical Signal Processing and Control*, vol. 3, no. 1, pp. 1 – 18, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809407000778

[116] Vicon. (2010) Optical motion capture system. [Online]. Available: http://www.vicon.com

[117] PhaseSpace, "Phasespace impulse system," in *Available: http://www.phasespace.com/productsMain.html*, 2013.

[118] Microsoft. (1999) Xbox kinect motion capture system. [Online]. Available: http://research.microsoft.com/enus/projects/vrkinect/

[119] IpiSoft, "Ipisoft system," in *Available: http://ipisoft.com/*, 2012.

[120] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013. [Online]. Available: http://doi.acm.org/10.1145/2398356.2398381

[121] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, Oct 2013.

[122] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *Trans. PAMI*, 2012. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=174951

[123] P. P. Valentini, "Natural interface in augmented reality interactive simulations," *Virtual and Physical Prototyping*, vol. 7, no. 2, pp. 137–151, 2012.

[124] D. S. Alexiadis, P. Kelly, P. Daras, N. E. O'Connor, T. Boubekeur, and M. B. Moussa, "Evaluating a dancer's performance using kinect-based skeleton tracking," in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM '11. New York, NY, USA: ACM, 2011, pp. 659–662. [Online]. Available: http://doi.acm.org/10.1145/2072298.2072412

[125] J. Heiser, D. Phan, M. Agrawala, B. Tversky, and P. Hanrahan, "Identification and validation of cognitive design principles for automated generation of assembly instructions," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, ser. AVI '04. New York, NY, USA: ACM, 2004, pp. 311–319. [Online]. Available: http://doi.acm.org/10.1145/989863.989917

[126] M. Dalal, S. Feiner, K. McKeown, S. Pan, M. Zhou, T. Höllerer, J. Shaw, Y. Feng, and J. Fromer, "Negotiation for automated generation of temporal multimedia presentations," in *Proceedings of the Fourth ACM International Conference on Multimedia*, ser. MULTIMEDIA '96.

New York, NY, USA: ACM, 1996, pp. 55–64. [Online]. Available: http://doi.acm.org/10.1145/244130.244147

[127] G. Zimmerman, J. Barnes, and L. Leventhal, "A comparison of the usability and effectiveness of web-based delivery of instructions for inherently-3d construction tasks on handheld and desktop computers," in *Proceedings of the Eighth International Conference on 3D Web Technology*, ser. Web3D '03. New York, NY, USA: ACM, 2003, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/636593.636601

[128] S. Kim, I. Woo, R. Maciejewski, D. S. Ebert, T. D. Ropp, and K. Thomas, "Evaluating the effectiveness of visualization techniques for schematic diagrams in maintenance tasks," in *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, ser. APGV '10. New York, NY, USA: ACM, 2010, pp. 33–40. [Online]. Available: http://doi.acm.org/10.1145/1836248.1836254

[129] D. Dionne, S. de la Puente, C. León, R. Hervás, and P. Gervás, "A model for human readable instruction generation using level-based discourse planning and dynamic inference of attributes disambiguation," in *Proceedings of the 12th European Workshop on Natural Language Generation*, ser. ENLG '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 66–73. [Online]. Available: http://dl.acm.org/citation.cfm?id=1610195.1610205

[130] J. E. Brough, M. Schwartz, S. K. Gupta, D. K. Anand, R. Kavetsky, and R. Pettersen, "Towards the development of a virtual environment-based training system for mechanical assembly operations," *Virtual Reality*, vol. 11, no. 4, pp. 189–206, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10055-007-0076-4

[131] F. Duan, J. Tan, J. G. Tong, R. Kato, and T. Arai, "Application of the assembly skill transfer system in an actual cellular manufacturing system," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 1, pp. 31–41, Jan 2012.

[132] S. Henderson and S. Feiner, "Exploring the benefits of augmented reality documentation for maintenance and repair," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1355–1368, Oct 2011.

[133] D. Kalkofen, M. Tatzgern, and D. Schmalstieg, "Explosion diagrams in augmented reality," in *IEEE Virtual Reality Conference*, March 2009, pp. 71–78.

[134] A. Pichler and C. Wogerer, "Towards robot systems for small batch manufacturing," in *IEEE International Symposium on Assembly and Manufacturing*, May 2011, pp. 1–6.

[135] S. Gupta, D. Anand, J. Brough, R. Kavetsky, M. Schwartz, and A. Thakur, "A survey of the virtual environments-based assembly training applications," in *In Virtual Manufacturing Workshop*, Turin, Italy, 2008.

[136] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 348–361, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0296-z

[137] ——, "A novel representation and feature matching algorithm for automatic pairwise registration of range images," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 19–40, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-3221-0

[138] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, Sept 2009, pp. 689–696.

[139] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, "Salient local visual features for shape-based 3d model retrieval," in *IEEE International Conference on Shape Modeling and Applications*, June 2008, pp. 93–102.

[140] X. Li, A. Godil, and A. Wagan, "Spatially enhanced bags of words for 3d shape retrieval," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. Chun, T.-M. Rhyne, and L. Monroe, Eds. Springer Berlin Heidelberg, 2008, vol. 5358, pp. 349–358. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-89639-5_34

[141] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli, "Depth mapping using projected patterns," 2012, uS Patent 8,150,142. [Online]. Available: http://www.google.com/patents/US8150142

[142] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252 – 1262, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865507000621

[143] J. Novatnack and K. Nishino, "Scale-dependent 3d geometric features," in *IEEE International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[144] Y. Liu, H. Zha, and H. Qin, "Shape topics: A compact representation and new algorithms for 3d partial shape retrieval," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2025–2032.

[145] A. Frome, D. Huber, R. Kolluri, T. Blow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3023, pp. 224–237. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24672-5_18

[146] J. Tangelder and R. Veltkamp, "A survey of content based 3d shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11042-007-0181-0

[147] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, May 1999.

[148] C. Chua and R. Jarvis, "Point signatures: A new representation for 3d object recognition," *International Journal of Computer Vision*, vol. 25, no. 1, pp. 63–85, 1997. [Online]. Available: http://dx.doi.org/10.1023/A%3A1007981719186

[149] F. Stein and G. Medioni, "Structural indexing: Efficient 3-d object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 125–145, Feb. 1992. [Online]. Available: http://dx.doi.org/10.1109/34.121785

[150] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3d object recognition from range images using local feature histograms," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. II–394–II–399 vol.2.

[151] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *ECCV*, vol. 3, 2004, pp. 224–237.

[152] A. Mian, B. M, , and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, 2009.

[153] A. Mian, M. Bennamoun, and R. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 19–40, 2009.

[154] J. Tangelder and R. Veltkamp, "A survey of content based 3d shape retrieval methods," in *Shape Modeling International*, 2004, pp. 145–156.

[155] A. Freedman, B abd Shpunt, M. Machline, and Y. Arieli, "Depth mapping using projected patterns," in *Patent Application, 10. WO 2008/120217 A2*, 2008.

[156] L. Laperriere and H. ElMaraghy, "Planning of products assembly and disassembly," in *CIRP Annals - Manufacturing Technology*, vol. 1, no. 41, 1992, pp. 51–9.

[157] N. Rajan, K. Sivasubramanian, and J. Fernandez, "Accessibility and ergonomic analysis of assembly product and jig designs," in *International Journal of Industrial Ergonomics*, vol. 23, no. 5–6, 1999, pp. 473–487.

[158] M. Strandberg, "Augmenting rrt-planners with local trees," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 3258–3262.

[159] S. Haddadin, A. Albu-Schffer, and G. Hirzinger, "Safety analysis for a human-friendly manipulator," *International Journal of Social Robotics*, vol. 2, pp. 235–252, 2010, 10.1007/s12369-010-0053-z. [Online]. Available: http://dx.doi.org/10.1007/s12369-010-0053-z

[160] S. Haddadin, A. Albu-Schffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, November/December 2009. [Online]. Available: http://ijr.sagepub.com/content/28/11-12/1507.abstract

[161] J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human-robot interaction," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 479–504, 2003. [Online]. Available: http://ijr.sagepub.com/content/22/7-8/479.abstract

[162] C. Vogel, M. Poggendorf, C. Walter, and N. Elkmann, "Towards safe physical human-robot collaboration: A projection-based safety system," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 3355 –3360.

[163] D. Shin, I. Sardellitti, Y.-L. Park, O. Khatib, and M. R. Cutkosky, "Design and control of a bio-inspired human-friendly robot," *I. J. Robotic Res.*, pp. 571–584, 2010.

[164] J. A. Marvel, "Performance metrics of speed and separation monitoring in shared workspaces," *Automation Science and Engineering, IEEE Transactions on*, vol. PP, no. 99, pp. 1–10, (accepted).

[165] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3,

pp. 253–270, 2008.

[166] P. Laborie and M. Ghallab, "Planning with sharable resource constraints," in *International Joint Conference on Artificial Intelligence*, vol. 14. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995, pp. 1643–1651.

[167] C. Smith and H. I. Christensen, "Wiimote robot control using human motion models," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on.* IEEE, 2009, pp. 5509–5515.

[168] P. Maybeck, *Stochastic Models, Estimation, and Control.* Academic Press, Inc, New York, 1979, vol. 1.

[169] J. A. Corrales, F. A. Candelas, and F. Torres, "Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, ser. HRI '08. New York, NY, USA: ACM, 2008, pp. 193–200. [Online]. Available: http://doi.acm.org/10.1145/1349822.1349848

[170] C. Andrieu and A. Doucet, "Particle filtering for partially observed gaussian state space models," *Journal of Royal Statistical Society*, vol. 64, pp. 827–836, 2002.

[171] S. Davies, "Watching out for the workers [safety workstations]," *Manuf., IET.*, vol. 86, no. 4, p. 3234, 2007.

[172] A. C. and D. A., "Robots for industrial environments: Safety requirements. part 1: Robot," *ISO 10218-1:2011*, 2011.

[173] C. Andrieu and A. Doucet, "Robots and robotic devices: Safety requirements. part 2: Industrial robot systems and integration," *ISO/FDIS 10218-2:2011*, 2011.

[174] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," in *Collaborative Manipulation Workshop at the ACM/IEEE International Conference on Human-Robot Interaction*, 2013.

[175] S. Petitjean, "A survey of methods for recovering quadrics in triangle meshes," in *ACM Computing Surveys*, 2002.

[176] R. Newcombe and A. Davison, "Live dense reconstruction with a single moving camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[177] R. Newcombe, S. Lovegrove, and A. Davison, "Dtam: Dense tracking and mapping in real-time," in *International Conference on Computer Vision (ICCV)*, 2011.

[178] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, K. Pushmeet, J. Shoton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[179] S. Izadi, D. Kim, O. Hilliges, R. Newcombe, D. Molyneaux, R. Newcombe, P. Kohli, J. Shoton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving

depth camera," in *ACM Symposium on User Interface Software and Technology*, 2011.

[180] R. Toldo, A. Beinat, and F. Crosilla, "Global registration of multiple point clouds embedding the generalized procrustes analysis into an icp framework," in *3DPVT International Conference on 3D Data Processing, Visualization and Transmission*, 2010.

[181] C. Goodall, "Procrustes methods in the statistical analysis of shape," in *Journal of the Royal Statistical Society. Series B (Methodological)*, 1991.

[182] S. Krishnan, P. Lee, J. Moore, and S. Venkatasubramanian, "Global registration of multiple 3d point sets via optimization-on-a-manifold," in *Third Eurographics Symposium on Geometry processing*, 2005.

[183] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Third International Conference on 3D Digital Imaging and Modeling*, 2001.

[184] W. P. A. and V. T., "Algorithms for 3-dimensional weighted orthogonal procrustes problems," in *Technical Report UMINF-06.06, Department of Computing Science, Umea University*, June 2006.