IMPROVING CRANE SAFETY BY AGENT-BASED DYNAMIC MOTION PLANNING USING UWB REAL-TIME LOCATION SYSTEM

Cheng Zhang

A Thesis
in the Department
of
Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

November 2010

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Cheng Zhang

Entitled: Improving Crane Safety by Agent-Based Dynamic Motion Planning using UWB Real-Time Location System

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________________ Chair
Dr. J.X. Zhang

______________________________________ External Examiner
Dr. T. Hegazy

______________________________________ External to Program.
Dr. J. Bentahar

______________________________________ Examiner
Dr. S. Alkass

______________________________________ Examiner
Dr. T. Zayed

______________________________________ Thesis Supervisor
Dr. A. Hammad

Approved by ______________________________________
Dr. K. Ha-Huy, Graduate Program Director

November 18, 2010 ______________________________________
Dr. Robin A.L. Drew, Dean
Faculty of Engineering & Computer Science

ABSTRACT

Improving Crane Safety by Agent-Based Dynamic Motion Planning Using UWB Real-Time Location System

Cheng Zhang, Ph.D.
Concordia University, 2010

The safe operation of cranes requires not only the experience of the operator, but also sufficient and appropriate support in real time. Due to the dynamic nature of construction sites, unexpected changes in the site layout may create new obstacles for the crane that can result in collisions and accidents. Limited research has been done on efficient re-planning for cranes with near real-time environment updating while considering communications between construction crews.

To improve the safety of mobile crane operations and to provide more awareness on site, the present research proposes a near real-time monitoring and motion planning approach to improve crane safety on construction sites using an ultra wideband (UWB) real-time location system (RTLS) technology. In addition, an agent system framework is proposed to guide crane operators for safe crane operations by enhancing environment awareness and by providing intelligent re-planning. Location data are collected from tags attached to cranes and are processed by the agent system to identify the poses of dynamic objects, which is used to generate a new motion plan to guide the crane movement and thus to avoid potential collision.

A motion planning algorithm, RRT-Con-Con-Mod, is proposed to efficiently generate safe and smooth paths for crane motions, mainly for the boom movement, while taking into account the engineering constraints and the path quality. A dynamic motion planning

algorithm, DRRT-Con-Con-Mod, is proposed to ensure safety during the execution phase by quickly re-planning and avoiding collisions. In addition, an anytime algorithm is proposed to search for better solutions during a given time period by improving the path smoothness and by reducing the path execution time. The proposed algorithms are compared with other motion planning and re-planning algorithms. The results show that the proposed algorithms can quickly find a safe and smooth motion plan.

Several tests of a UWB system have been applied in the laboratory and in indoor and outdoor environments to investigate the requirements of applying UWB on construction sites, that is, requirements including accuracy, visibility, scalability, and real-time. To satisfy these requirements, the configuration of the UWB system has been analyzed in detail to decide the sensors' and tags' locations and numbers based on heuristic rules. These tests show a good potential for using UWB tracking technology in construction sites by processing and organizing location data into useful information for near real-time environment updating.

Furthermore, the framework of an agent system is proposed to integrate the proposed methodologies of motion planning and near real-time tracking. Different agents are created to represent the equipment, to coordinate tasks, and to update the site information. The functions of these agents include exchanging information, deciding priorities, etc.

The current research will benefit the construction industry by providing more awareness of dynamic construction site conditions, a safer and more efficient work site, and more reliable decision support based on good communications.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Dr. Amin Hammad, for his guidance and advice throughout my research. Without his intelligent support and encouragement, I could not have made such progress in this research. Most importantly, I learned from him how to be persistent and enthusiastic in research under pressure, even at moments of great frustration. During these years of hard work, the experience I have gained from him will benefit my whole life.

Second, I would like to thank my research colleagues for their kind support in developing the simulation system and in helping with the tests of the UWB system. Homam AlBahnassi has contributed to this research by bringing new ideas to motion planning algorithms and by collaborating in the development of the system. Sonia Rodriguez has helped with the data processing of the UWB system. I deeply appreciate their contributions. Moreover, I would like to give my appreciation to Steven Stein and Shan Zhu for their support in the testing of the integrated system, and to Ali Motamedi, Abhinav Kaushik, Yu Satow, Junyu Jian, Narsreen AlHerz for their help with setting and testing the UWB system.

In addition, I would like to thank a number of other people for their contributions: Mr. Joshua Barnhardt from Ubisense for training and for the technical support of the system; Mr. Bruce Harper from Dessau Inc. and Mr. Marc Tardif from Verreault for providing the logistical support for the indoor test; Mr. Christian Servais and Mr. Rafael Palomar from GUAY Inc. for providing the space and the crane for the outdoor tests; Mr. Renaud Cote from Group Bellemare, Mr. Claude Bourassa from La Commission de la Sante et de la Securite du Travail du Quebec, Mr. Jean-Louis Lapointe from GUAY Inc., and Mr. Yvan Grenier from Association des Proprietaires de Machinerie Lourde du Quebec for providing valuable evaluation and feedback.

Last but not least, I would like to thank Dr. Sabah Alkass, Dr. Tarek Zayed, and Dr. Jamal Bentahar for spending their precious time in reading this thesis and for their valuable comments as members of my thesis committee.

This research has been supported by the Natural Sciences and Engineering Research Council (NSERC) Doctoral Fellowship, Hydro Quebec Doctoral Fellowship and research grants from NSERC and the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST).

DEDICATION

This thesis is dedicated to my family for their continuous encouragement and support from the initial to the final stages of my research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
A-GPS	Assisted Global Positioning System
AEC	Architecture, Engineering, and Construction
AI	Artificial Intelligence
AOA	Angle of Arrival
APF	Artificial Potential Field
AR	Augmented Reality
C-space	Configuration Space
CAD	Computer-Aided Design
C_{free}	Free Configuration Space
DoF	Degree of Freedom
DRRT	Dynamic Rapidly-exploring Random Trees
FIPA	Foundation for Intelligent Physical Agents
FoV	Field of View
GA	Genetic Algorithm
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphic User Interface
GVG	Generalized Voronoi Graph
ICT	Information and Communication Technologies
IR	Infrared
LED	Light Emitting Diode
MAS	Multi-Agent Systems
MSL	Motion Strategy Library
NIST	National Institute of Standards and Technology
OBI	On-Board Instrumentation
PoE	Power over Ethernet
PQP	Proximity Query Package

PRM	Probabilistic Roadmap planner
RC	Radio Controlled
RF	Radio Frequency
RFID	Radio Frequency Identification
RRG	Rapidly-exploring Random Graph
RRT	Rapidly-exploring Random Trees
RRT-Con	RRT-Connect
RRT-Con-Con	RRT-Connect-Connect
RRT-Con-Con-Mod	RRT-Connect-Connect-Modified
RSS	Received Signal Strength
RTLS	Real-time Location System
SCS	Smart Construction Site
SDK	Software Development Kit
TDOA	Time Difference of Arrival
UWB	Ultra-Wideband
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

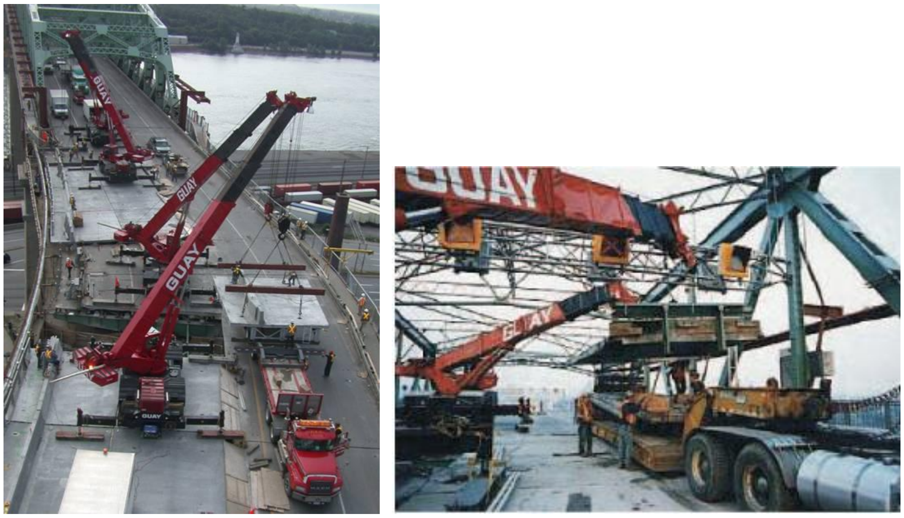
CHAPTER 1 INTRODUCTION

1.1 GENERAL BACKGROUND

Safety and productivity issues on construction sites are always among the major concerns of project managers. The complexity of on-site conditions requires careful planning and coordination of different equipment to ensure safety and efficiency. Previous research has indicated that machinery-related incidents were the fourth leading cause of traumatic occupational fatalities in the construction industry between 1980 and 1992, resulting in 1,901 deaths (2.13 deaths per 100,000 workers) (NIOSH, 2007). The same research has indicated that the construction equipment most frequently associated with fatalities are cranes (17%), excavators (15%), tractors (15%), loaders (9%), and pavers (7%). Taking cranes as an example, in 2006, there were 72 crane-related fatal occupational injuries in the U.S. (Crane-Related Occupational Fatalities, 2006). In Canada, there were 56 accidents related to cranes in the province of British Columbia in 2006 (WorkSafeBC, 2010); and during the period of 1974 to 2002, there were 23 accidents with injuries, 26 accidents with death, and 13 accidents with material damage related to cranes in Quebec province (CSST, 2010). In addition, the numbers of reported accidents and the resulting deaths have been increasing during the past 10 years (Crane Accident Statistics, 2010). Furthermore, crane accident statistics are limited because typically only deaths and injuries are reported. Property damage incidents are usually not reported; however, the seriousness of a crane accident is self-evident (Crane Safety on Construction Sites, 1998).

It is estimated that one crane upset occurs during every 10,000 hours of crane use. Approximately 3% of upsets result in death, 8% in lost time, and 20% in damage to property other than the crane. Nearly 80% of these upsets can be attributed to predictable human errors when the operator inadvertently exceeds the crane's lifting capacity (Davis and Sutton, 2003). Crane operators need sufficient and appropriate training with instruction for safe vehicle operation. Safety should be practiced and enforced each and every time the equipment is used (Heavy Equipment Safety, 2005). According to Beavers et al. (2006), mobile cranes represented over 88% of the fatal crane-related events. Furthermore, “electrocution” and “crane tip over” were associated only with mobile cranes. These data suggest that more emphasize should be put on the operation of mobile cranes.

Figures 1-1(a) and (b) show a bridge rehabilitation project (Zaki and Mailhot, 2003), carried out in 2001 and 2002, where groups of cranes and crews were involved in removing old deck sections and installing new panels. The complexity of the construction environment put many constraints on the mobilization, transportation, collaboration of equipment, work interference (multi-groups), tight schedule (traffic should be open during day time), spatial constraints (existing structure of the bridge), and so on. Consequently, the motion of the cranes was carefully planned and tested using scaled physical models and 3D simulation tools. As can be seen from this example, the spatio-temporal aspects of certain construction projects need to be well planned and monitored to ensure safety and to improve efficiency.



(a) (b)

Figure 1-1: Construction site of a bridge rehabilitation project (Zaki and Mailhot, 2003)

The operation of cranes is a complex job, requiring not only the experience of the operator, but also sufficient and appropriate support in real time. For example, the operator of a hydraulic crane may have some blind spots, and most of the time, he is engaged in concentrating on his work without full perception of the environment. Furthermore, the noise and vibration from the equipment impair the cognitive ability of the operator. To help crane operators accomplish their tasks safely and efficiently, researchers and engineers have been investigating various methods. Several software tools have been developed to select the type and location of cranes, and to plan the path of a lift. However, because of the dynamic nature of construction sites, unexpected changes in the site layout may create new obstacles for the crane that can result in collisions and accidents. Early in 1998, the use of computers to plan safe operations and to maintain the crane in a safe operating condition was proposed in a report from the

American Society of Civil Engineers (Crane Safety on Construction Site, 1998). However, at that time, the focus was only on simulating lifts for operations and for training purposes. The real-time monitoring issues were not considered. Nowadays, it is possible to think about providing a real-time planning support and about giving more awareness on site to the operators to improve the safety of crane operations. Using different types of sensors, it is possible to forecast and to avoid collisions based on continuously capturing and updating information about obstacles in the surrounding environment. These sensors can be used to obtain information about static and dynamic obstacles. Static obstacles are those obstacles about which information is known in advance so that they can be taken into account during the planning phase. Dynamic obstacles are objects that are moving on site, such as trucks, workers, and construction equipment. These dynamic obstacles need to be detected and their positions need to be updated while executing the initial plan, thereby necessitating re-planning because of potential collisions. By the utilization of a wide array of sensors, equipment operators can have better situation awareness and can make more informed decisions.

1.2 RESEARCH OBJECTIVES

The objectives of the present research are the following: (1) to propose a new approach for guiding crane operations taking into consideration engineering and spatial constraints updated from near real-time data collection and information exchange technologies; (2) to investigate motion planning and re-planning algorithms that can be applied to cranes to reduce safety risks on site; (3) to explore the usability of emerging technologies for field data capturing on construction sites and for detecting dynamic objects; (4) to develop an agent-based framework to enhance the communication and negotiation between staffs on the construction site; and (5) to develop a prototype system that can be used for near real-time support in construction by integrating the above methods and techniques.

1.3 THESIS STRUCTURE

The structure of this thesis is as follows: The literature review is described in Chapter 2, focusing on related topics in the areas of construction simulation, motion planning for robots and construction equipment, environment perception technologies, and agent technologies. Chapter 3 describes an overview of the research methodology, which includes the conceptual approach, the criteria for motion planning algorithm selection, the requirements of the sensing technology that should be satisfied in construction, and the framework of a multi-agent system. In Chapter 4, the proposed motion planning and re-planning algorithms are discussed in detail, and two case studies are used to show the feasibility of applying these algorithms. In Chapter 5, the near real-time environment data collection and analysis are described to explore how to organize these data into information that can be used for collision detection. Chapter 6 presents the prototype system design and integrated tests to validate the proposed approach in a controlled environment. The contributions and limitations of the present research and the direction of future work are discussed in Chapter 7.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION

The present literature review focuses on topics related to the present research in the areas of construction simulation, motion planning for robots and construction equipment, environment perception technologies, and agent technologies. The purpose is to investigate the trends in research and in the industry and to investigate the possibility of applying advanced technologies to allow crane operators to have a better awareness of site information and to get support from an intelligent system.

Simulation methods and tools are used to help in selecting the cranes, in simulating the task, and in visually checking the spatial constraints on site. Construction engineers need to survey the construction site and to use 3D environment models to plan critical lifts aiming to improve efficiency and reduce safety risks. Many issues need to be considered in crane selection and path planning, such as the capacity of the crane, the location of temporary material storage, the crane moving path, and the vision coverage of the crane operator. Available methods and tools are reviewed in Section 2.2.

A detailed lifting plan is needed for critical crane lift processes. In current practice, planners have to manually check the working range of the selected crane to guarantee that it can reach the initial and goal locations of the lift objects. In addition, a collision-free path is planned to execute the tasks. Five main steps are defined by Kang (2005) to simulate the process of moving a lift object: (1) picking the object with a series of delicate crane motions, (2) lifting the object vertically to a minimum safety height, (3) moving the object to the top of the destination location, (4) waiting for the decrease of the vibration caused by the crane motions, and (5) lowering and moving the object to the destination location. Step three requires that there be an effective motion planning method for the crane, which is essential to the success of the entire task. Motion planning algorithms available in robotics research are reviewed in depth to evaluate their applicability to construction equipment, as introduced in Section 2.3.

One problem in the current state-of-practice is that the information is insufficient during the planning. For example, the 3D environment model usually does not include enough details, such as details about electric poles and lines. Such details are important since electrocution is the second major cause of fatalities during the performance of hoisting activities (Beavers et al., 2006). In addition, the surrounding environment could change during construction, thereby making the original lift plan unfeasible. Hence, it is important to consider that the construction site is dynamic and unpredictable, and near real-time information updating is necessary to capture these changes. Several environment perception technologies are reviewed in Section 2.4.

Once the environment has changed, fixing on-site problems can delay the schedule or increase the cost. These problems can be significantly reduced if efficient motion re-planning is applied based on the updated environment information. Related motion planning research is reviewed in Section 2.3.

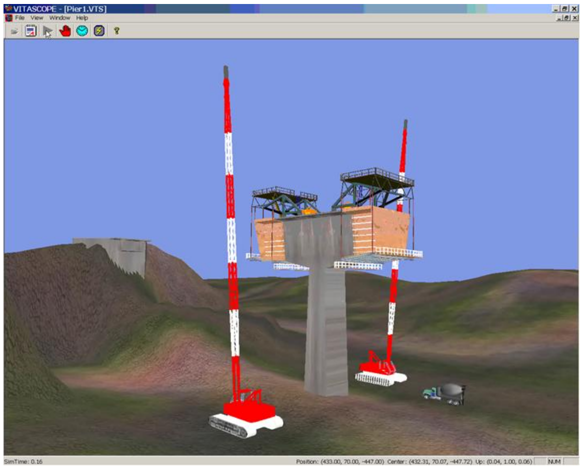
Even though detailed motion planning and efficient re-planning in near real-time are applied, the safety problems may not be completely solved because cranes should not be considered as robots without coordinating the work of different cranes working in the same area. In practice, lifting tasks are usually done through a trial-and-error process, based on the feedback provided by the operators' own vision and assessment, on hand signals of a designated ground director at the work zone, or on radio communication. Standard hand signals for crane operation are shown in Appendix A. In addition, a lift plan is a micro plan that should be integrated with other plans to ensure that the entire project is done properly. Extensive communication should be undertaken on site to coordinate the cranes' movement based on negotiation among construction team members. The priority of tasks also needs to be considered when a conflict between two tasks is detected. Beavers et al. (2006) have suggested that employers should have a system in place to assess the hazardousness of each of their construction worksites in relation to the potential for crane-related events. They have also suggested that a diligent and competent person should be assigned by the manager of the construction operations to be in charge of overall crane operations. This person should have complete authority to stop any unsafe operations. This need has inspired the research to investigate the possibility of an intelligent system to support all the on-site workers in a search for better communication and environment awareness. Therefore, in Section 2.5, agent technology is reviewed to explore the feasibility of its application in construction to enhance safety.

2.2 SIMULATION OF CONSTRUCTION PROCESSES

To achieve a better understanding of construction processes, simulation tools have been developed: (1) to simulate and visualize these processes (Kamat and Martinez, 2001), (2) to analyze and to avoid collisions between equipment (Zhang et al., 2007), (3) to test and to visualize equipment location and to plan the path of equipment manually (Cranimax, 2010; LiftPlanner, 2010), and (4) to train operators of heavy equipment using virtual reality (Simlog, 2010). The advantage of visualizing the work processes is that the user can simulate and check the functional constraints and interferences that can happen in reality between the 3D physical elements and virtual workspaces.

Figure 2-1 shows an animation snapshot of a construction site. The visualization is based on the results of the simulation, which is not equipped with any collision-detection mechanism and which does not have any feedback about the unplanned environment changes. Therefore, if a spatial problem is detected in the visualization phase, the simulation has to be repeated after changing the input data.

Figure 2-2 shows the simulation of an automatic pouring system of a concrete boom pump. The trajectories of the boom are analysed using inverse kinematics to select a feasible path of the boom; however, no collision detection is applied between the boom sections and the obstacles in the environment.



SimTime: 0.16

Position: (433.00, 70.00, -447.00) Center: (432.31, 70.07, -447.72) Up: (0.04, 1.00, 0.06)

Figure 2-1: VITASCOPE animation snapshot of a construction site (Kamat and Martinez, 2001)

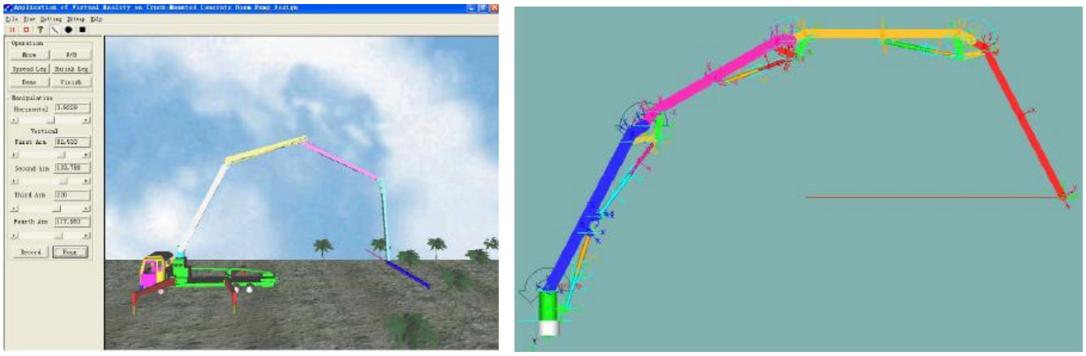


Figure 2-2: Automatic pouring system simulation (Zhou and Zhang, 2007)

One method to represent the physical spaces occupied by objects on site is to create virtual workspaces. Using such virtual workspaces enhances safety by allowing such equipment workspaces to be defined as safety zones for carrying out specific tasks. In a study of the bridge rehabilitation project mentioned in Chapter 1, simplified shapes are used to represent the workspaces of equipment and to analyse possible collisions between equipment, and between equipment and obstacles (Hammad et al., 2007). Figure 2-3 shows a schematic representation of two cranes working together on the bridge (Zaki and Mailhot, 2003). Figures 2-4(a) and (b) show the side view and the top view of the workspaces represented on the bridge, respectively (Hammad et al., 2007). However, this analysis deals only with the static environment without considering the dynamic features on site, thus reducing the practical value of the simulation in the support of real-time decision-making.

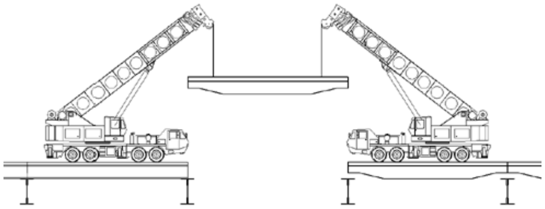
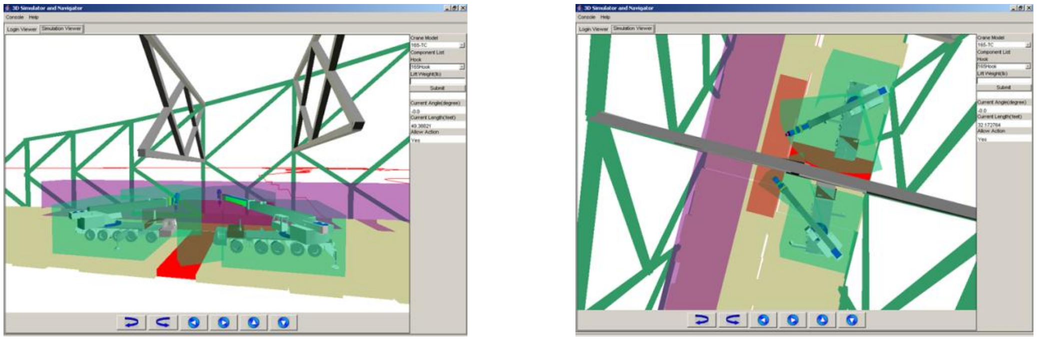


Figure 2-3: Schematic representation of two cranes (Zaki and Mailhot, 2003)



Crane Model
165-TC
Component List
Hook
165Hook
Lift Weight(lb)
Current Angle(degree)
-0.0
Current Length(feet)
49.36621
Allow Action
Yes

Crane Model
165-TC
Component List
Hook
165Hook
Lift Weight(lb)
Current Angle(degree)
-0.0
Current Length(feet)
32.172764
Allow Action
Yes

(a) Side view **(b) Top view**

Figure 2-4: Workspace representation on the bridge (Hammad et al., 2007)

Cranimax (2010) is crane selection software, which calculates the outrigger forces for mobile cranes, the distribution of ground pressures for crawling cranes, and the minimum and maximum radius ranges. Figure 2-5 shows an example of positioning a crane on site. LiftPlanner (2010) is a 3D crane and rigging planning software system, which produces drawings to plan and document critical lifts. However, these software systems focus on the engineering constraints of the crane and provide the detailed selection and configuration of the crane; however, they require the users to plan the path manually for moving the object while taking into account obstacles in the 3D environment.

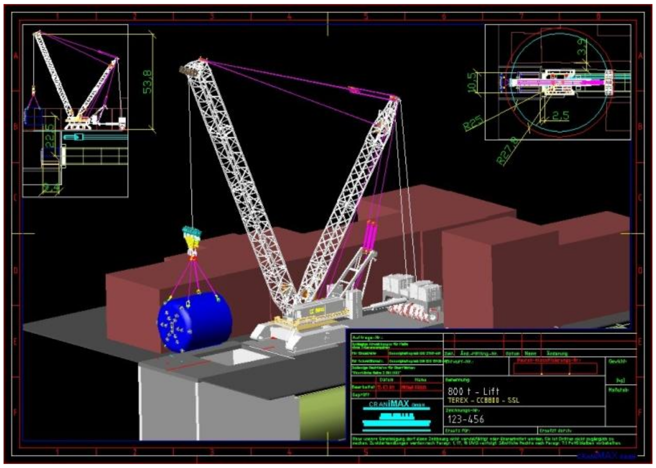


Figure 2-5: Crane positioned in a 3D environment (Cranimax, 2010)

Training simulation for equipment operation has been used as a cost-effective tool for the operators (Ritchie, 2004). Simlog (2010) provides training for different equipment with various scenarios, such as pouring concrete using a bucket lifted by a tower crane (Figure 2-6). The shortcoming of this training software is that it does not use the real construction environment and it is not designed to provide motion planning.

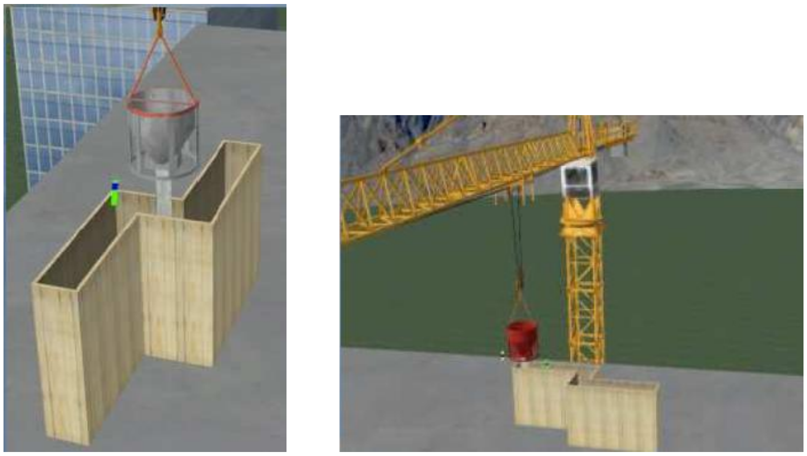


Figure 2-6: Simlog training scenario (Simlog, 2010)

Simulating the construction environment and processes has the advantage of ensuring the reliability of a construction plan by visually checking for potential collisions or other problems. However, the simulation tools reviewed above provide only off-line support for the crane positioning and motion planning. Capturing and using near real-time data would provide new opportunities for quality control and safety assurance. Near real-time support for collision avoidance and re-planning is investigated in Chapters 3, 4, and 5.

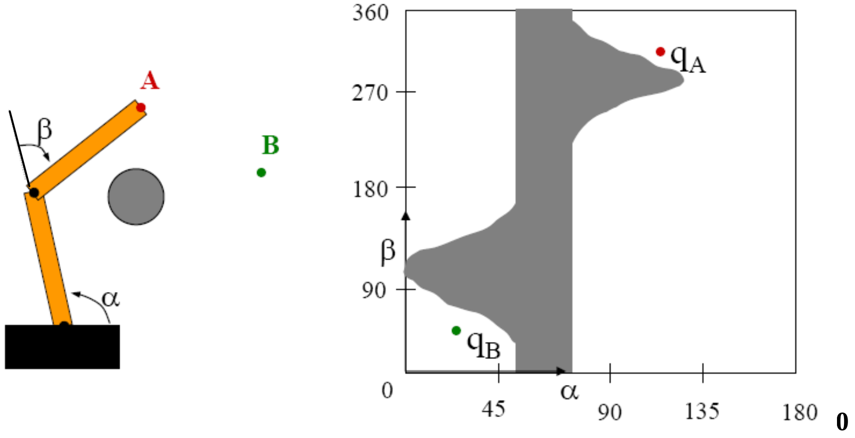
2.3 MOTION PLANNING

The research into motion planning has a long history in robotics. One of the most important tasks for robot motion planning is navigation, which aims to find a collision-free path for the robot system from one configuration (or state) to another. In the present thesis, Subsection 2.3.1 reviews research into robotic motion planning; Subsection 2.3.2 reviews motion planning algorithms; and applications of motion planning for construction equipment are reviewed in Subsection 2.3.3.

2.3.1 ROBOTIC MOTION PLANNING

To create motion plans for a robot, a search space is needed in either the actual workspace or a space representing the configurations of the robot, which is called the configuration space (*C-space*). Most of the current approaches for motion planning are based on the concept of *C-space* introduced by Lozano-Pérez and Wesley (1979). *C-space* is the set of all possible configurations of a robot. A configuration is simply a point in this abstract *C-space*. The configuration of a robot system is the complete specification of the position of every point in that system. Once the motion planning problem has been formulated in the *C-space*, it becomes equivalent to finding the connected sequence of collision-free configurations running from the initial configuration to the goal configuration. The number of the degrees of freedom (DoFs) of a robot system is the dimension of the *C-space*, or the minimum number of parameters needed to specify the configuration.

Figure 2-7(a) shows an obstacle in the workspace of a robot with 2 DoFs, α and β. Figure 2-7(b) shows the representation of an obstacle in the *C-space*, which is a two dimensional representation of angles α and β. q_A and q_B correspond to the configurations of the endpoint positions *A* and *B*, respectively. The area corresponding to the unfeasible configurations caused by the obstacle is shown in grey in the *C-space*. The remaining part in the *C-space* is the set of configurations at which the robot does not intersect with any obstacle. This part is called the free configuration space (C_{free}). Once the *C-space* is generated, motion planning requires only a search between the pick (origin) and the place (destination) locations in the *C-space*.



(a) An obstacle in the workspace of the robot

(b) The *C-space* showing the obstacle and the two configurations at A and B

Figure 2-7: C-space of a two-link arm robot (Choset et al., 2005)

One of the advantages of using the *C-space* is that the unfeasible configurations arising out of invalid joint angles are treated as obstacles modeled in the *C-space* (Bandi and Thalmann, 1997). Another advantage is the avoidance of the need to solve the inverse kinematics problem to find the different solutions corresponding to the DoFs of the manipulator for a particular location of the end-effector.

2.3.2 MOTION PLANNING ALGORITHMS

Many algorithms are available for generating collision-free paths in the *C-space*. Based on the data structure representation of the *C-space*, motion planning algorithms can be categorized under two major approaches (Choset et al., 2005):

(1) Motion Planning in Discrete Space: In this case, the *C-space* is defined as a state-space model with a countable finite set of states. The planning algorithms build

roadmaps in the free (or semi-free) state-space and search for the feasible path. Each of these algorithms relies on an explicit representation of the geometry of the free space. Because of this, as the dimension of the *C-space* grows, these algorithms become impractical. Grid A* and Visibility Graph are representative algorithms of discrete space planning as discussed later in this section.

(2) Motion Planning in Continuous Space: In this case, the algorithm is not limited to a pre-defined finite search space representation of the *C-space*. Instead, a variety of strategies are utilized for generating samples (collision-free configurations) and for connecting the samples with paths to obtain solutions to path-planning problems in a continuous *C-space*. Sampling-based algorithms are capable of dealing with robots with many DoFs and with many different constraints. Such algorithms do not attempt to explicitly construct the boundaries of the *C-space* obstacles or to represent cells of the free space. Instead, they rely on a procedure that can decide whether a given configuration of the robot is in collision with the obstacle or not. Efficient collision detection procedures ease the implementation of sampling-based algorithms and increase the range of their applicability. PRM (Probabilistic Roadmap Planner) and RRT (Rapidly-exploring Random Trees) are representative algorithms of continuous space planning.

The following paragraphs briefly summarize the major representative algorithms.

Grid A*

Grid A* is a classical search method that finds the optimal path with respect to metrics such as energy, time, traversability, safety, etc., as well as combinations of them. The input of the search is a graph. The algorithm searches the graph efficiently with respect to a chosen heuristic, which hypothesizes an expected, but not necessarily actual, cost to the goal node. It is often applied to grids where each of the cells has its heuristic distance to the goal. As shown in Figure 2-8, a vertical or horizontal step between free space cells has a relatively low cost whereas the cost for travelling from a free space cell to an obstacle cell is made arbitrarily high. The detailed description of this algorithm can be found in Choset et al. (2005).

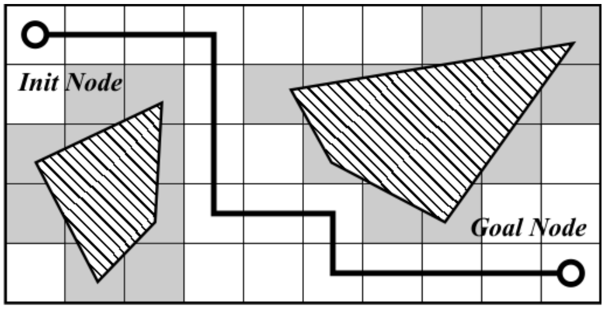


Figure 2-8: Grid example of A* algorithm (Adapted from Bruce, 2004)

Visibility Graph

The standard visibility graph is defined in a two-dimensional polygonal *C-space*. The nodes v_i of the visibility graph include the initial location, the goal location, and all the vertices of the *C-space* obstacles. The graph edges e_{ij} are straight-line segments that connect two line-of-sight nodes v_i and v_j. Note that the nodes and edges are embedded in the free space, and the edges of the polygonal obstacles also serve as edges in the visibility graph. The visibility graph is connected in a free space where all the components are connected.

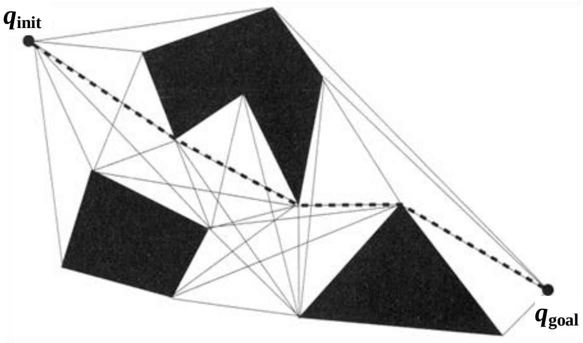


Figure 2-9: Example of a visibility graph (Choset et al., 2005)

Figure 2-9 shows an example of a visibility graph where the thin solid lines delineate the edges of the visibility graph for the three obstacles, which are represented as filled polygons. The thick dotted line represents the shortest path between the initial and goal nodes. Using the standard Euclidean distance, the visibility graph can be searched for the shortest path (Choset et al., 2005).

Probabilistic RoadMap planner (PRM)

The basic PRM first constructs a roadmap in a probabilistic way in the C_{free}. The roadmap is represented by an undirected graph. Figure 2-10 shows the steps of the basic PRM algorithm: (a) Find random sample of free configurations (vertices); (b) Attempt to connect pairs of nearby vertices. If a valid plan is found, add an edge to the graph; (c) Find local connections to the graph from the initial and goal nodes; and (d) Search the roadmap graph. The nodes of the roadmap are configurations in C_{free}, and the edges of the roadmap correspond to the free paths. The objective of the first phase (learning phase) is to capture the connectivity of C_{free} so that path-planning queries can be answered efficiently. In the query phase, the roadmap is used to solve individual path-planning problems (Choset et al., 2005).

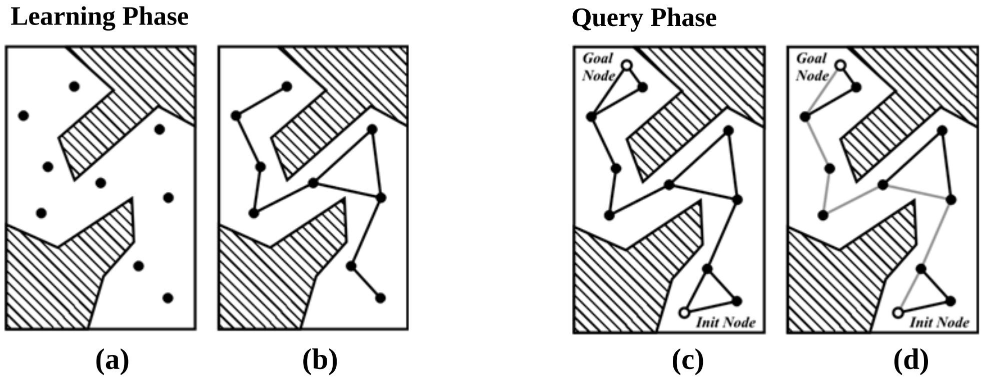


Figure 2-10: Example of PRM algorithm steps (Adapted from Bruce, 2004)

Rapidly-Exploring Random Trees (RRTs)

RRT algorithms incrementally construct a search tree rooted either at q_{init} or q_{goal} by incrementally branching to collision-free nodes. At each iteration, a random configuration, q_{rand}, is sampled uniformly in C_{free}. The nearest configuration, q_{near}, to q_{rand} in the tree is found, and an attempt is made to make progress, and finally connect q_{init} and q_{goal}. This method was originally developed by LaValle (1998). Figure 2-11 shows the steps of the basic RRT algorithm: (1) Start with the initial configuration as the root of a tree; (2) Pick a random state in the *C-space*; (3) Find the closest node in the tree; (4) Extend that node toward the state if possible; and (5) Goto step (2). Figure 2-12 shows an example of an RRT with 2000 vertices (LaValle and Kuffner, 1999).

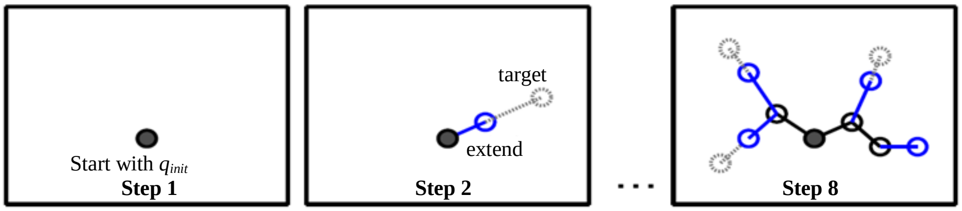


Figure 2-11: Steps of the basic RRT algorithm (Bruce and Veloso, 2006)

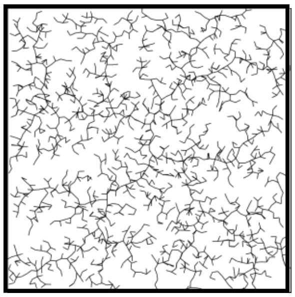


Figure 2-12: Example of an RRT with 2000 vertices (LaValle and Kuffner, 1999)

The comparison of the four algorithms above is shown in Table 2-1, which is based on the following criteria:

(1) Completeness: Complete planning approaches are guaranteed to find a solution when it exists, or correctly report failure if one does not exist (LaValle, 2006). For sampling-based algorithms (e.g. RRT), completeness depends on the probability of finding a solution. As more time is spent, the probability of producing a solution approaches 1. Improvements to the standard RRT can be carried out to address this issue (Cheng and LaValle, 2002). For Grid A*, finding the solution depends on the resolution of the grid representing the *C-space*; low resolution grids may result in the failure to find the solution even if it exists. PRM combines both cases of being probabilistic and resolution complete. This is due to its nature of finding the path in two phases.

(2) Optimality: Algorithm optimality is its ability to return an optimal path with respect to some metrics. Single-query sampling-based algorithms (e.g. RRT) are not able to guarantee the generation of an optimal path based on pre-defined criteria; an optimization update is required to address this point. Fortunately, for many of these

algorithms, the solutions produced are not too far from optimal in practice (LaValle, 2006).

(3) Efficient environment updates: Because of dynamic obstacles, changes in the environment are very common cases. Therefore, efficiency in re-planning the path after updating the environment is important. Among the algorithms reviewed in this research, RRT is the best even though it is considered semi-efficient (Bruce, 2004). RRT is a single query algorithm, which attempts to solve a query as fast as possible but does not focus on the exploration of the entire free space. A* efficiency in world updates can be improved with D* (Choset et al., 2005), by propagating cost changes, while maintaining the optimality of A* and making minimal changes to the universal plan. Literature has shown that RRTs are much faster than grid-based searching algorithms (Brandt, 2006).

(4) Efficient query updates: In addition to environment updates, query update efficiency is important for cases like re-planning to new goals while fixing environment constraints. The PRM algorithm is efficient in this type of query, since it can reuse the roadmap that it constructed in the preprocessing phase.

(5) Good DoF scalability: The DoFs directly affect the complexity of *C-space*; thus many algorithms are not able to solve problems involving configurations with high DoFs efficiently. Grid A* and Visibility Graph are not suitable for solving configurations with high DoFs, thereby limiting realistic kinematic modeling for construction equipment.

(6) Non-holonomic: The capability of solving non-holonomic configurations is a key feature in path-planning algorithms, where the algorithm is not only limited by

considerations of global constraints that are generated from explicit obstacles in the environment (Kuffner and LaValle, 2000), but algorithms should also be able to address local/differential constraints that may be found in some types of construction equipment. Among all reviewed algorithms, RRT stands out because of its high ability in solving non-holonomic configurations.

Other criteria could be related to one or more of the criteria above. For example, complexity could be related to the criteria of DoF scalability and the efficient world updates.

Table 2-1: Summary of the comparison between algorithms (Bruce, 2004)

Algorithm	Completeness	Optimality	Efficient Environment Updates	Efficient Query Updates	Good DoF Scalability	Non-Holonomic
Grid A∗	res	grid	no	no	no	no
Visibility Graph	yes	yes	no	no	no	no
PRM	prob, res	graph	no	yes	yes	semi
RRT	prob	no	semi	semi	yes	yes

Res: Resolution Complete, Prob: Probabilistic Complete

2.3.3 MOTION PLANNING FOR CONSTRUCTION EQUIPMENT

Although motion planning algorithms have been studied in computer science and robotics for more than thirty years, little research has focused on motion planning for construction equipment. Construction equipment can be treated as robots, and the same motion planning algorithms can be applied; however, appropriate domain heuristics should be added to find a good/optimal plan within a reasonable time (Reddy and Varghese, 2002), and no industry-wide standard exists for heavy lift path-planning practices (Varghese et al., 1997).

Tserng et al. (2000) have proposed a methodology and several algorithms for interactive motion planning that are developed for multi-equipment landfill operations in an Automated Landfill System (ALS). This methodology simulates the operational processes of landfill vehicles and equipment in planning a landfill project. Based on this simulation, efficient and collision-free motion patterns are found to control autonomous landfill equipment during the construction phase. However, this system depends on pre-defined patterns to do motion planning for the equipment, which prevents the system from solving actual cases where there could be equipment on site that does not follow any of the specified moving patterns.

Kim and Paulson (2003) have introduced a path-planning method for a mobile construction robot to find a continuous collision-free path from the initial position of the construction robot to its goal position. This work presents an improved Bug-based algorithm, called SensBug, which can produce an effective and short path in an unknown environment with both stationary and moving obstacles because of the following characteristics: (1) An improved method to decide a local direction, which allows the mobile construction robot to generate an effective path in the environment with both stationary and mobile obstacles; (2) A reverse mode, which can provide a mobile construction robot with a way to overcome the problem of obstacles in a complex configuration; and (3) A simple leaving condition, which allows the mobile construction robot to leave the obstacle boundary as soon as possible. These improvements make it possible to overcome the weak points of the previous algorithms. However, these improvements did not overcome the safety issue in all variations of bug algorithms where generated paths touch the obstacles. This issue is caused by how the bug algorithm navigates through the environment where it depends on wall-hugging the obstacles until it reaches the specified goal.

Sivakumar et al. (2003) have tried different algorithms, such as A* and Genetic Algorithms (GAs) to optimize the collision-free path for cooperative lifting with two cranes. In the research of Ali et al. (2005), a GA is used and compared with the A* algorithm, and the former is considered a better solution for two cranes working together. Figure 2-13 shows the paths traced by the hooks' ends of two cooperative manipulators using a GA. However, the authors have assumed that the site contains only static obstacles, and the proposed solutions provide only off-line planning, rather than real-time control of the movement.

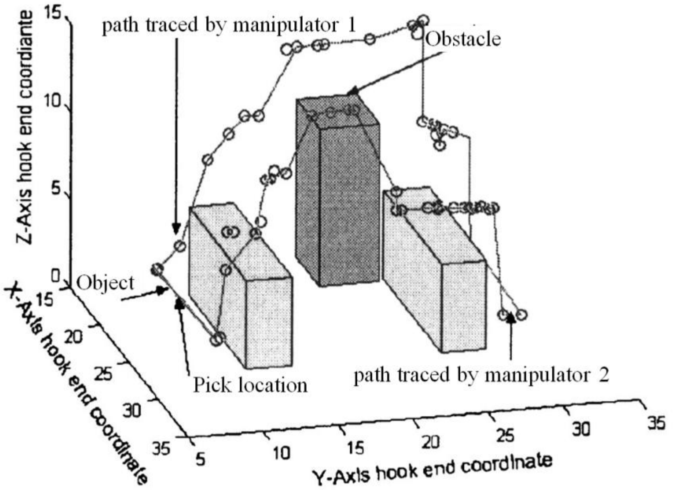


Figure 2-13: Paths of cooperative manipulators using GA (Ali et al., 2005)

One specific issue should be considered when planning the motion for construction equipment in multi-equipment environments. The multi-equipment problem deals with motion planning for many pieces of equipment, where each piece of equipment can be a moving obstacle for other equipment. When obstacles are in motion, a continuous function of time specifying the equipment's configuration at each instant must be generated. A collision-free path from an initial configuration to a goal configuration implies that at every step there is no collision between the equipment and an obstacle or between different pieces of equipment. A solution to the multi-equipment problem must be able to coordinate these paths so that no two pieces of equipment enter into collision. This requirement makes the problem significantly harder than in the case of a single piece of equipment.

One way to deal with multiple equipment operating in the same workspace is centralized planning, which treats the multiple equipment as single multi-bodied equipment (Latombe, 1991). The composite *C-space* is the set product of the *C-spaces* of the individual equipment. Every configuration in the composite *C-space* determines a unique position and orientation for each piece of equipment. The difficulty of centralized planning arises from the high dimensionality of the composite *C-space*.

Another approach to motion planning with multiple equipment is decoupled planning, which consists of planning the motion of each equipment independently and considering the interactions among the paths in a second phase of planning. Decoupled planning works in two stages: initially, collision-free paths are computed for each piece of equipment individually, not taking into account the other equipment but simply considering the obstacles of the workspace. In the second stage, coordination is achieved by computing the relative velocities of the equipment along their individual paths so as to avoid collision among them (Choset et al., 2005).

Kang and Miranda (2006) have proposed an incremental decoupled method to plan motions for multiple cranes so that collisions among any of the cranes are avoided as are possible collisions between the cranes and the transported objects. In the case of two cranes, first, plans are generated for each crane individually by ignoring the other crane during a small period δ, and then coordinating both of the cranes by tuning their relative velocities to avoid collisions. If successful, the system plans the next time period δ until the entire project is finished. Otherwise, a new δ is considered and steps are repeated for the entire project.

Three different algorithms, QuickLink, QuickGuess, and RandomGuess, were integrated to find a path efficiently (Kang and Miranda, 2006). However, these three algorithms are conceptually similar to the RRT algorithm but are simplified for the specific case of a tower crane. Path-refining algorithms were developed to optimize a given path, as shown in Figure 2-14. Although this research considered dynamic changes on site to make the path more realistic, it was assumed that the environment information was known by exactly following the work schedule.

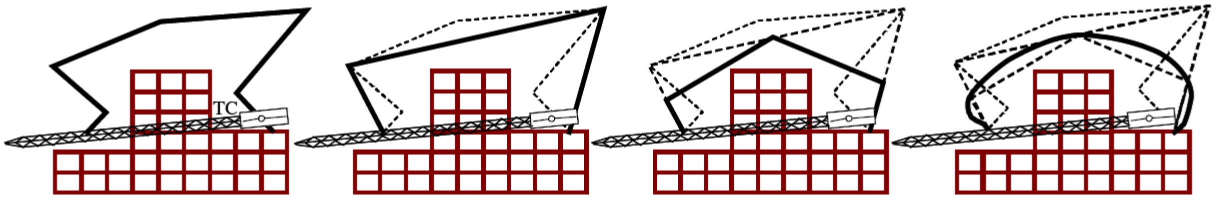


Figure 2-14: An example of the path refining process (top view) (Kang and Miranda, 2006)

The real situation on site is that unknown objects should also be monitored and taken into account to ensure the collision-free movement of equipment. Information about unknown objects can be collected by sensors. In robotics research, sensor-based motion planning incorporates sensor information, reflecting the current state of the environment, into a robot's planning process, as opposed to *classical planning*, where full knowledge of the world's geometry is assumed to be known prior to the planning event (Choset and Burdick, 2000). In sensor-based motion planning, prior knowledge of the world is not available, is inaccurate, or changes rapidly, where the robot is supposed to sense the data in real-time and make quick responses.

There are many algorithms for solving motion planning problems; however, many of them are not amenable to sensor-based interpretation. It is not possible to simply add a step to acquire sensory information and then to construct a plan from the acquired model using a classical technique when the world model is unknown, since the robot needs a path planning strategy in the first place to acquire the world model. To address this problem, Khatib (1986) has proposed an artificial potential field (APF) to guide the movement of robots in real time. Choset and Burdick (2000) have developed an incremental approach to constructing the Generalized Voronoi Graph (GVG) from sensor data.

In the case of motion planning for equipment on construction sites, the model-based approach is used during the planning stage. In this approach, a 3D model of the site is available and full information about the geometry of the equipment and the obstacles is given beforehand, so path planning becomes a one-time off-line operation. During the execution stage, the dynamic environment needs sensor-based planning on the assumption that some obstacles are unknown. This lack of information is compensated for by local on-line (real-time) information coming from sensory feedback (Spong et al., 1992). The difference between motion planning for equipment on construction sites and the robotic exploration in an unknown environment is that every task carried out on a construction site has a schedule; therefore, the unknown information can be assumed to be minor or less essential to the whole plan, and a motion re-planning approach can efficiently modify the off-line plan based on real-time sensed data.

2.4 ENVIRONMENT PERCEPTION TECHNOLOGIES

A construction site has two types of obstacles: static and dynamic. Static obstacles are those obstacles that do not move, and about which information is known in advance. They can, therefore, be considered during the planning phase. Examples of these obstacles include buildings, electrical poles, etc. Dynamic obstacles are objects that move on site, such as trucks, workers, and construction equipment. These dynamic obstacles should be detected and updated while the initial plan is being executed. Such obstacles may necessitate re-planning because of potential collisions. By the utilization of a wide array of sensors, equipment operators can have better situation awareness and can make more informed decisions. Researchers have been trying different technologies to create accurate 3D models of construction sites and to track and control equipment automatically.

Methods used for modeling static objects and for tracking moving objects are reviewed in Subsections 2.4.1 and 2.4.2, respectively. Subsection 2.4.3 briefly reviews research related to the automation of construction equipment.

2.4.1 MODELING STATIC OBJECTS

Several methods can be used to create the 3D models of static objects. Photogrammetry is used for calculating geometric properties of objects based on photographic images (Photogrammetry, 2007). 3D modeling using Geographic Information Systems (GIS) is also used to create an urban model based on extruding polygons representing the footprints of buildings on maps according to the heights of the buildings (ArcGIS 3D Analyst, 2010).

As shown in Figure 2-15, the downtown campus of Concordia University is highlighted in a partial 3D model of Montreal City. These data are becoming more available in some cities. However, these models include buildings mainly and miss other small objects, such as traffic signs, fire hydrants, and electric poles and lines. Consequently, researchers are trying different technologies to create detailed 3D models of construction sites. 3D laser scanners are used to collect point clouds, which can be transformed by software tools into volumetric objects, representing a precise 3D model, including all the buildings and other objects. Repeated work should be carried out to update the model over time. Gordon and Akinci (2005) have collected data using a 3D laser scanner to support inspection and quality control on construction sites. Figure 2-16 shows a 3D scanner and a sample of point cloud collected for the Jacques Cartier Bridge shown in Figure 1-1. These point clouds can be used to create the 3D model of the bridge to avoid collision between cranes and the bridge structures during rehabilitation projects of the bridge.

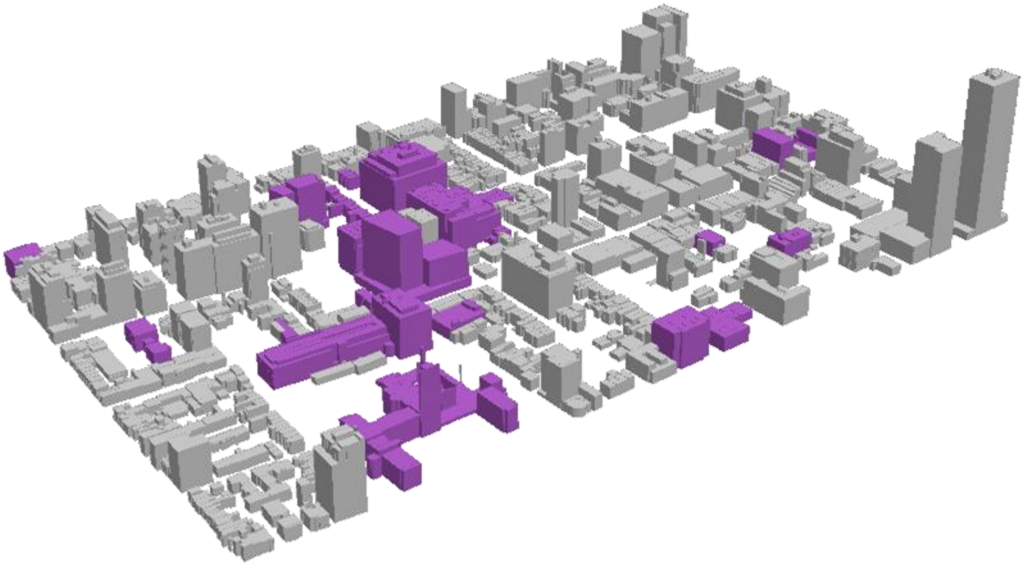
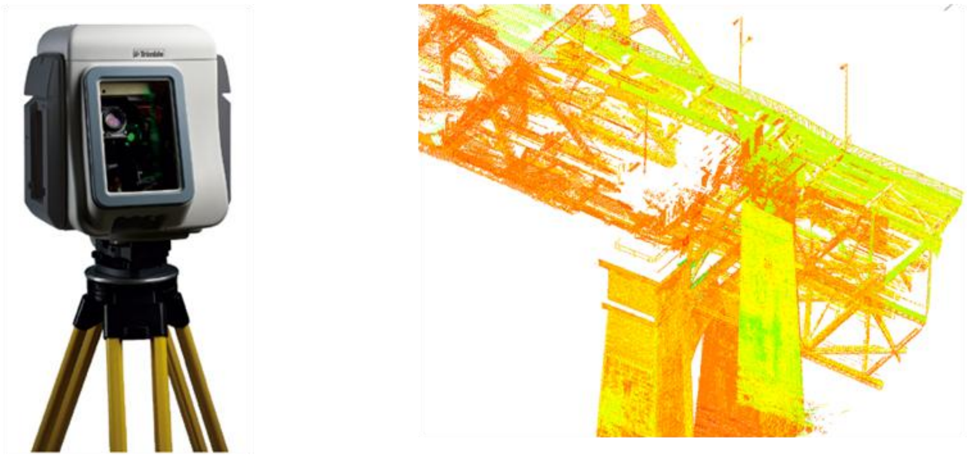


Figure 2-15: GIS-based partial 3D model of downtown area in Montreal



(a) Laser scanner (Trimble GX, 2010)

(b) Point cloud sample (Mailhot and Busuio, 2006)

Figure 2-16: Laser scanner and point cloud collected for a part of a bridge

Researchers at NIST have been studying the performance and applicability of 3D range cameras. The cameras measure the distance to an object by measuring the time needed for light to travel from the instrument to the object and back. They can capture the 3D scene in real time at video frame rates (MESA Imaging, 2010). Lytle et al. (2005) have evaluated the performance of a 3D range camera for construction applications. Some important parameters are indicated to optimize the accuracy and to minimize errors (Price et al., 2007). Teizer et al. (2006) have used a 3D range camera to model static and dynamic construction resources. Figure 2-17 shows an example of 3D range cameras.

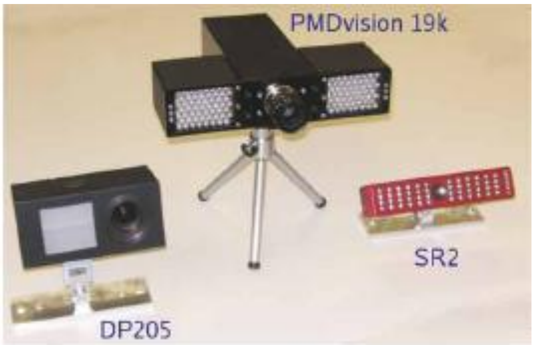


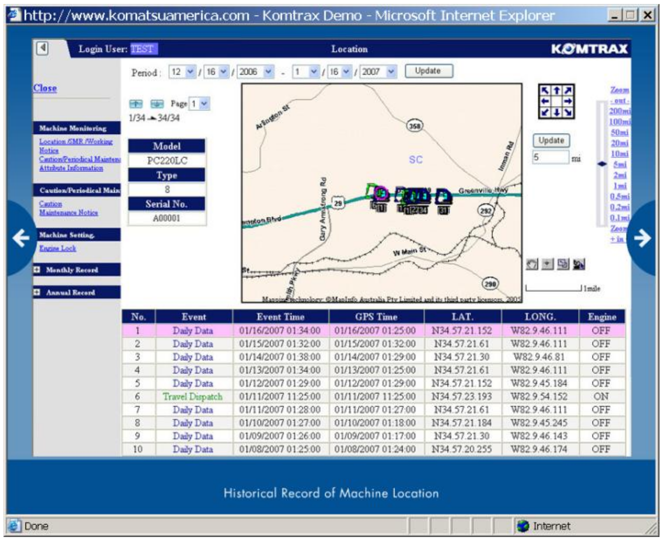
Figure 2-17: Range cameras (Price et al., 2007)

2.4.2 TRACKING MOVING OBJECTS

As explained earlier, construction sites are dynamic, requiring the continuous updating of the location data of all moving objects, including equipment and workers, to mitigate safety risks. Subsection 2.4.2.1 reviews research related to tracking of construction equipment. Subsections 2.4.2.2 and 2.4.2.3 introduce ultra wideband technology and its applications in construction.

2.4.2.1 Tracking of Construction Equipment

The most popular tracking technologies used on construction sites is the Global Positioning System (GPS), which is widely used in construction, mining, surveying, and infrastructure projects. For example, in earthmoving projects, GPS and total station technology are used to accurately position the blade of the excavator in real time, significantly reducing material overages and dramatically improving contractors' productivity and profitability (Trimble GCS900, 2010). Navon et al. (2004) have developed a tracking and control system using GPS and on-board instrumentation (OBI) to monitor, in real-time, the activity of major construction equipment, such as tower cranes, concrete pumps, etc. Alshibani and Moselhi (2007) have used GPS for tracking earthmoving equipment to forecast performance. Riaz et al. (2006) have tracked vehicles and workers using GPS and sensors to reduce accident rates. GPS is also used to locate equipment world wide as shown in Figure 2-18 (Komtrax, 2010).



Login User: TEST

Period : 12 / 16 / 2006 - 1 / 16 / 2007

Page 1

5 mi

Model
PC220LC
Type
8
Serial No.
A00001

No.	Event	Event Time	GPS Time	LAT.	LONG.	Engine
1	Daily Data	01/16/2007 01:34:00	01/16/2007 01:25:00	N34.57.21.152	W82.9.46.111	OFF
2	Daily Data	01/15/2007 01:32:00	01/15/2007 01:32:00	N34.57.21.61	W82.9.46.111	OFF
3	Daily Data	01/14/2007 01:38:00	01/14/2007 01:29:00	N34.57.21.30	W82.9.46.81	OFF
4	Daily Data	01/13/2007 01:34:00	01/13/2007 01:25:00	N34.57.21.61	W82.9.46.111	OFF
5	Daily Data	01/12/2007 01:29:00	01/12/2007 01:29:00	N34.57.21.152	W82.9.45.184	OFF
6	Travel Dispatch	01/11/2007 11:25:00	01/11/2007 11:25:00	N34.57.23.193	W82.9.54.152	ON
7	Daily Data	01/11/2007 01:28:00	01/11/2007 01:27:00	N34.57.21.61	W82.9.46.111	OFF
8	Daily Data	01/10/2007 01:27:00	01/10/2007 01:18:00	N34.57.21.184	W82.9.45.245	OFF
9	Daily Data	01/09/2007 01:26:00	01/09/2007 01:17:00	N34.57.21.30	W82.9.46.143	OFF
10	Daily Data	01/08/2007 01:25:00	01/08/2007 01:24:00	N34.57.20.255	W82.9.46.174	OFF

Figure 2-18: Historical record of machine locations (Komtrax, 2010)

However, GPS requires direct line of sight from the satellites to the receiver, and accurate GPS receivers are expensive to install on every moving object on site. Therefore, other tracking technologies have been applied in several research projects, such as infrared, optical, ultrasound, and Radio Frequency Identification (RFID) technologies. Chae and Yoshida (2008) have discussed collecting data on site using RFID active tags to prevent collision accidents (Figure 2-19). BodyGuard - Vehicle Proximity Alert and Collision Avoidance System (Orbit Communications, 2008) is an RFID-based system that offers continuous detection and notification of proximity between moving objects and other moving or fixed objects by setting up protection zones around a vehicle, equipment, and buildings to offer continuous protection for valuable resources. However, RFID can give only approximate locations.

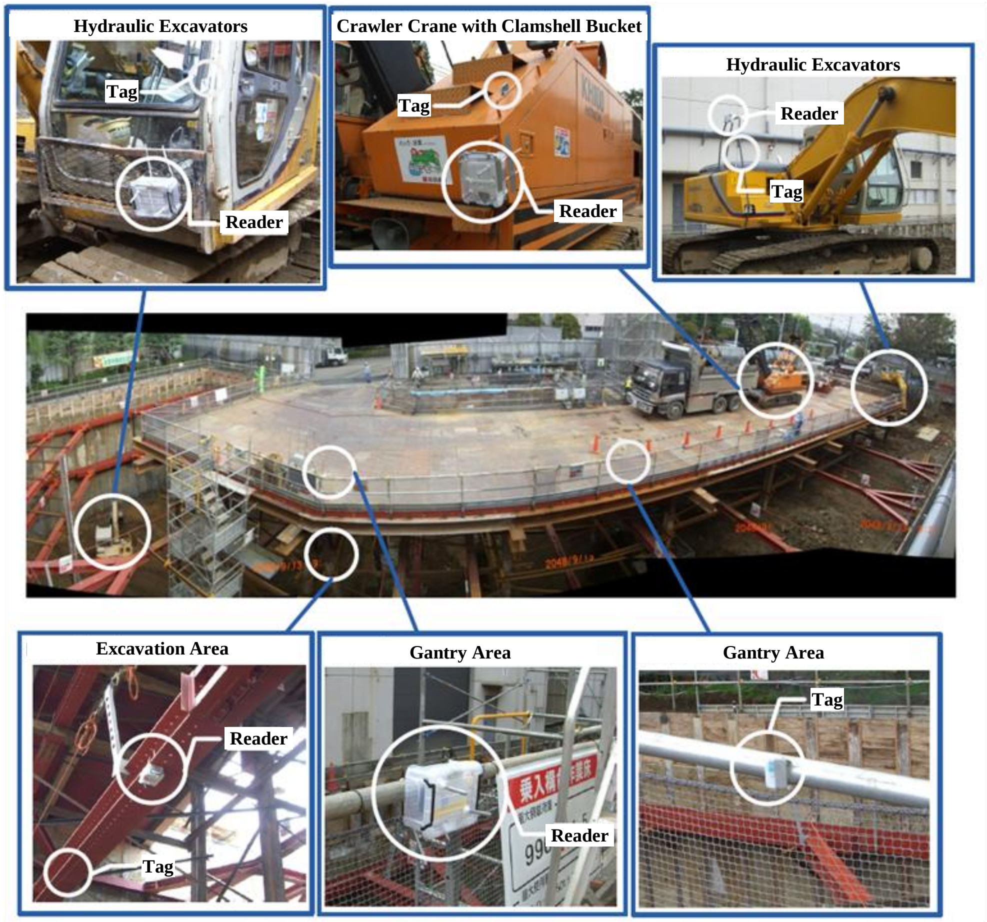


Figure 2-19: Construction site equipped with RFID devices (Chae and Yoshida, 2008)

The Real-time Automated Project Information and Decision Systems (RAPIDS) lab at Georgia Institute of Technology is testing 3D laser scanners, 3D range cameras, total stations, GPS, RFID, and other types of sensors and technologies for automated collection and processing of data for applications in construction projects (Figure 2-20) (Teizer et al., 2005; Teizer et al., 2006; Teizer and Castro-Lacouture, 2007). However, most of this research is still at the initial testing stage.

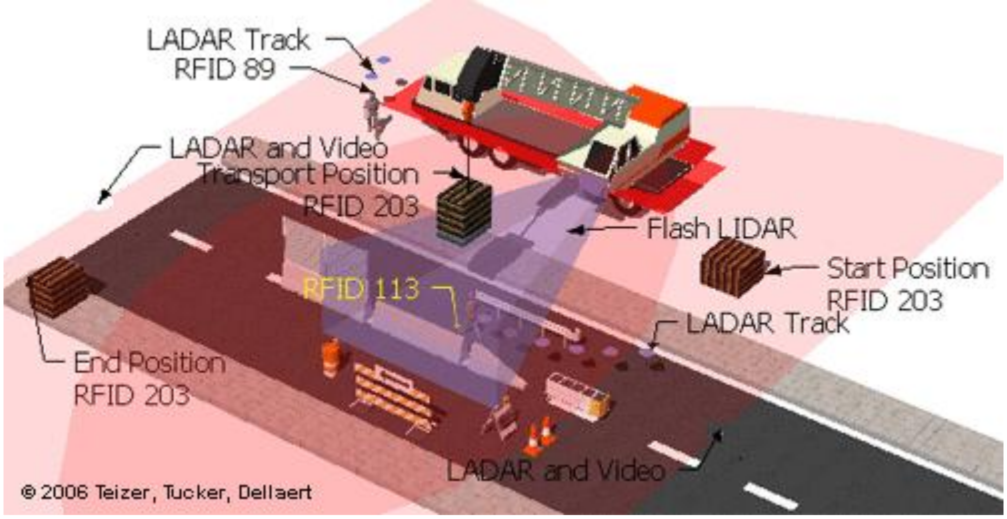


Figure 2-20: Tracking devices used in construction research (Teizer, 2006)

Recently, real-time location systems (RTLSs) have been applied in various areas, such as in logistics and manufacturing. RTLSs can track and identify the location of objects in real time using tags attached to objects and sensors fixed at known locations. The sensors detect signals emitted by the tags and calculate the locations of these tags. Figure 2-21 compares different location technologies, such as the following: passive RFID, electromagnetic, laser, ultrasound, infrared (IR) proximity, conventional Radio Frequency (RF) timing, UWB, Wireless Local Area Network (WLAN), Received Signal

Strength (RSS), and assisted GPS (A-GPS). This comparison is carried out based on the accuracy and the coverage offered by each technology to identify the ideal technology (Ward, 2007). According to Muthukrishnan and Hazas (2009), ultra wideband (UWB) technology delivers a robust localization with an accuracy of up to 15 cm in good conditions.

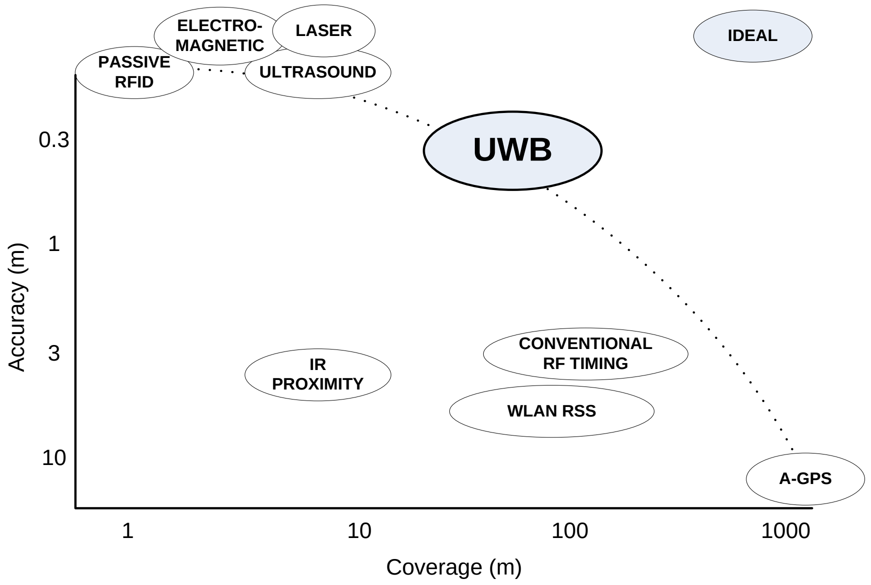
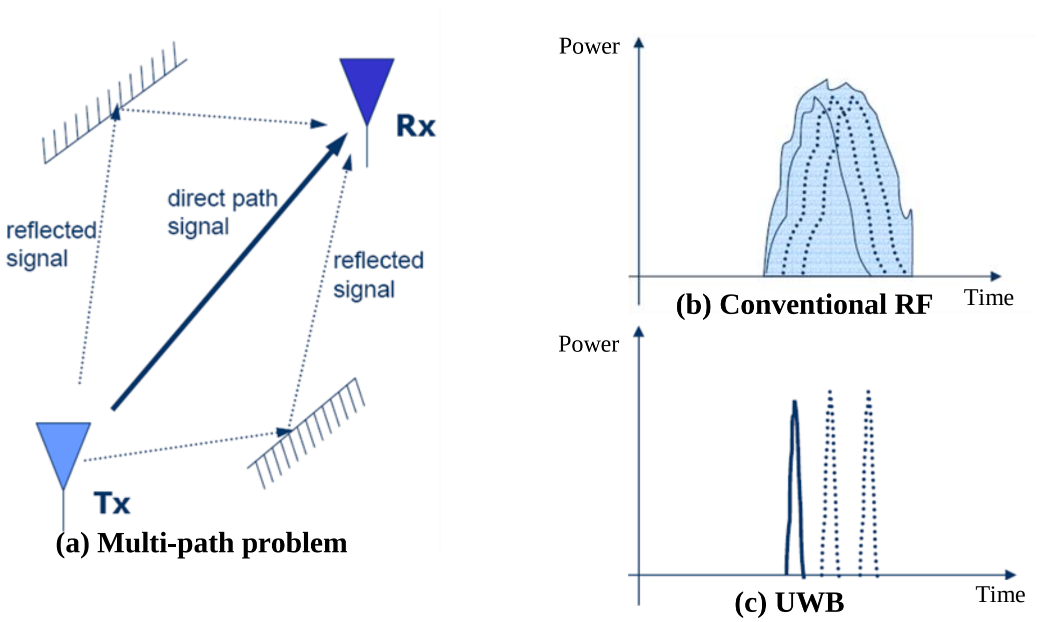


Figure 2-21: Comparison of location technologies (adapted from Ward, 2007)

Other than those general purpose sensors, specific sensors are designed and used in the industry for monitoring the physical condition of the equipment. In recent research, wireless sensors are installed on the boom of a crane to make sure the boom can withstand the varying stresses and strains as it turns, lifts, lowers, and reaches (Machinedesign, 2004). A locking mechanism based on OBI is applied to some cranes to limit the movement of the boom when it is approaching the target (Hirschmann, 2010). However, not all the cranes are equipped with OBI and, even when it is available OBI only provides the kinematic geometry of the cranes without the location relative to other objects. Therefore, this information is not enough for collision avoidance. In addition, sensors for detecting ground support settlements are applied during the lifting to ensure safety. Topcon Grading Control System (Topcon, 2008) is a two-dimensional system that consists of four tilt sensors. Whether the boom, stick and bucket of an excavator are fully extended or close to the cab, the distance of the bucket teeth from the final grade is clearly displayed, allowing total control of the job. Dialog Visu (Potain, 2008) is a tool for the man/machine dialogue with a complete display of the crane's operational characteristics in graphic and numerical forms. It provides direct reading of the load curve on the screen showing either the permitted load for a given range, or the permitted range for a given load. The load, height and range values are detected by three sensors linked by electrical connection. Increased safety on construction sites is expected by using these tools. However, these sensors are designed to give the physical condition of the equipment and cannot be directly used for collision avoidance.

2.4.2.2 UWB Technology

UWB is a wireless technology for transmitting large amounts of digital data over a wide spectrum of frequency bands at very low power (less than 0.5 milliwatts) (Ghavami et al., 2004). UWB has the ability to carry signals through doors and other obstacles that tend to reflect signals at more limited bandwidths and at a higher power. The Federal Communication Commission of the USA has limited communication coverage zone for pulse UWB systems by implementing power density mask on all transmitting devices. As a result, the UWB coverage area cannot exceed 100 meters (Bunin and Valikov, 2006). As shown in Figure 2-22, with conventional RF, the reflections in in-building environments distort the direct path signal, making accurate pulse timing difficult, whereas if UWB is used, the direct path signal can be distinguished from the reflections, making pulse timing easier. Thus, the accuracy of the UWB system can be up to 15 cm in good conditions. These conditions include the absence of multi-path problems because of the availability of direct line-of-sight signals from tags to sensors and the absence of metallic objects in the vicinity of the UWB system (Muthukrishnan and Hazas, 2009). In addition, UWB works better with metals than do other RF devices. These advantages make it possible to attach UWB tags to construction equipment and to other moving objects on site and collect accurate location data.



(a) Multi-path problem

(b) Conventional RF

(c) UWB

Figure 2-22: Multi-path problem with conventional RF and UWB (Ubisense, 2010)

In a UWB RTLS system, a sensor cell is composed of several sensors connected together into a single operating unit, which captures the location of tracked objects. Sensors are synchronized using a timing signal (distributed by timing cables) from each sensor to the timing source. The master sensor receives and synchronizes the timing data from the other slave sensors. Each tag registers with its containing sensor cell and is inserted into the schedule for that cell. The schedule determines when the tag should emit UWB signals to be located by the cell. The schedule is optimized to give attention to each tag as close as possible to its requested quality of service, while maintaining enough space in the schedule for new tags to register. The data radio channel is used to manage this location schedule. The UWB channel is used to transmit radio signals from tags to sensors at the scheduled time for each tag. These signals are used in calculating the locations of tags. When a tag emits a signal, this signal is picked up by one or more sensors in the cell, as shown in Figure 2-23. The slave sensors decode the UWB signal and send the angle of arrival and the timing information back to the master sensor through an Ethernet connection. The master sensor accumulates all sensed data and computes the location based on trilateration.

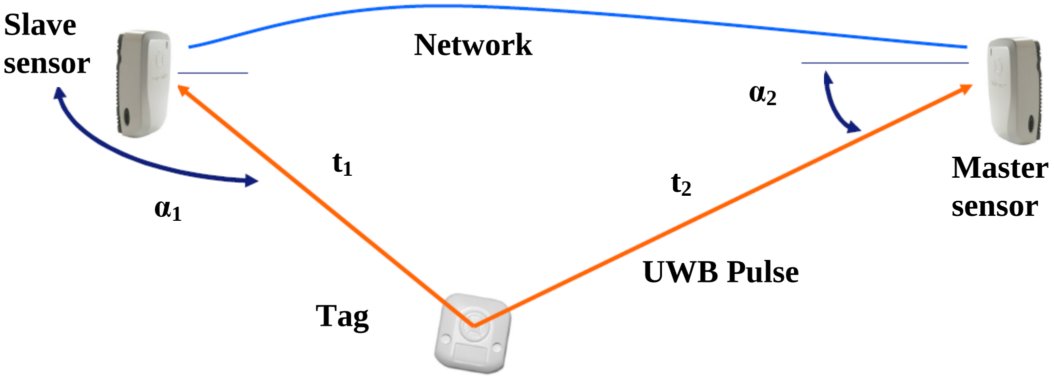


Figure 2-23: Signals sent to sensors are used to calculate the 3D position of a tag (Ubisense, 2010)

Several methods are used to measure the distance between the sensors and the tags, such as Time Difference of Arrival (TDOA) and Angle-of-Arrival (AOA). With a known position of the sensor, the objects on site with tags attached to them can be located. In the TDOA method, the difference in time at which the signal from the tag to be positioned arrives at different receivers is measured. Each time difference is then converted into a hyperboloid with a constant distance difference between two receivers. The position is found by solving equations to find the intersection of the corresponding hyperboloids. In 3D, at least four receivers are required because this technique requires the synchronization of the receivers' clocks (Ghavami et al., 2004). In AOA method, the angle of arrival of the signal sent by the tag is measured at several stationary receivers. Each measurement forms a radial line from the receiver to the tag. In 2D positioning, the position of the tag is defined at the intersection of two directional lines of bearing. This method has the advantage of not requiring synchronization of the receivers nor an accurate timing reference. On the other hand, receivers require regular calibration in order to compensate for temperature variations and mismatches (Ghavami et al., 2004). Location estimate of the tag is calculated at the intersection of these lines. In theory, direction-finding systems require only two receiving sensors to locate a tag, but in practice, to improve accuracy and compensate for finite angular resolution, multipath and noise, more than two sensors are needed (Munoz et al., 2009). Commercial products, such as Ubisense (2010) and Multispectral Solutions (2010) are available for evaluating the usability of UWB technology.

Research about the UWB tracking technology has been carried out in different domains. For example, a prototype UWB tracking system is under development at NASA Johnson Space Center (Ni et al., 2010). The system is being studied for use in tracking of lunar/Mars rovers and astronauts during early exploration missions when satellite navigation systems are not available. Field tests demonstrated that the prototype system is feasible for providing positioning-awareness information in a 3D space to a robotic control system.

2.4.2.3 Applications of UWB Tracking in Construction

Fontana (2007) has proposed that UWB could be used for improving crane safety. Teizer et al. (2007) have investigated the usability of a UWB tag attached to a crane hook to track the position of the hook. The tag is attached to the top of the hook, as indicated by the arrow in Figure 2-24.

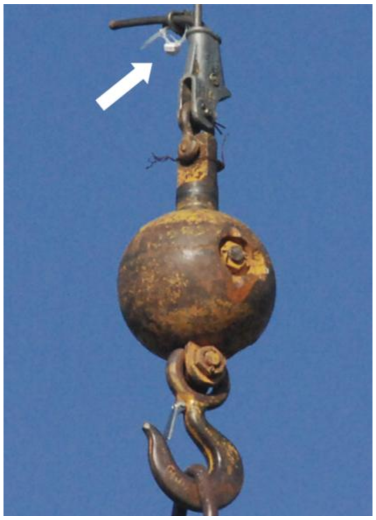


Figure 2-24: UWB tag on hook (Teizer et al., 2007)

Giretti et al. (2009) have indicated that UWB behaviour is rather constant during most parts of the construction progress. They note that, in an open area, tests confirm an accuracy of about 30 cm. They have also discussed a safety management system that gives an alarm when a worker is approaching a static, known dangerous area. Fullerton et al., (2009) have proposed using UWB for *proactive safety*, which works in real time to alert personnel of the dangers arising, and for *reactive safety*, which collects data to be analyzed in order to determine the best practices and to make process improvements. Carbonari et al. (2009) have proposed safety management systems for tracking workers' trajectories to prevent accidents. Cho et al. (2010) have discussed error modeling for an untethered UWB system for indoor construction asset tracking. Based on their experiment, elevated tags give a better line-of-sight path between the tag and the sensors. The average accuracy is 17 cm, while the tethered system gives 10 cm accuracy in open space. They conclude that the accuracy seems sensitive mainly to the location and the facing angle of sensors, which affect the chance of having a line-of-sight transmission path from mobile tags.

However, previous research did not investigate the requirements of using UWB RTLSs for improving crane and other construction equipment safety, where the dynamic and complex aspects of construction sites need to be considered. For example, tracking only the hook of a crane is not enough for collision avoidance. The full kinematic configuration of the crane should be identified to prevent collision with other equipment or workers. Therefore, the raw location data of tags should be processed to calculate the poses of the crane. Furthermore, the setting of the sensor's location and orientation and the number and location of tags should be investigated in detail to get more visibility of tags and to improve the accuracy of object locations.

2.4.3 AUTOMATION OF CONSTRUCTION EQUIPMENT

Unmanned construction is work performed by remotely operated construction machinery that corresponds to an operator-controlled robot. Unmanned construction was used in civil engineering work for the first time in Japan in 1969 when an underwater bulldozer was used to excavate and move deposited soil during emergency restoration work at the

Toyama Bridge that had been blocked by the Joganji River disaster. Since then, unmanned construction by excavators inside pneumatic caissons and by backhoes has been carried out. However, the restoration works following the volcanic eruptions that began in 1994 at the Unzen-fugendake Volcano and the eruption of the Usuzan Volcano in 2000 were the first examples of large-scale unmanned construction and have spurred rapid progress in unmanned construction technologies and encouraged their wide use (Ban, 2002).

Much research about construction automation has been carried out at the National Institute of Standards and Technology (NIST, 2010) in the U.S. The *Construction Metrology and Automation Group (CMAG)* is involved in the development of position/orientation tracking systems and sensor interface protocols. The *Computer Integrated Construction (CIC)* group is doing research on the visual representation and simulation of construction models (Furlani et al., 2002). *Intelligent Systems Division* with *CMAG* are researching a robotic structural steel placement project called *Automated Steel Construction Testbed* (Lytle et al., 2002; 2004). *CMAG* has been conducting research in crane automation since the mid 1980's. A robotic crane (*RoboCrane*) based on an inverted, cable-actuated Stewart-Gough platform (Angeles, 1997) principle was invented at NIST at that time. Since then several versions of the *RoboCrane* concept have been developed for various applications. Recently, *CMAG* is developing a generic crane controller using NIST real-time control system methodology in order to test and evaluate various automated crane control schemes. In addition, *CMAG* is working on methods and algorithms to identify construction components from high-resolution 3D laser scanning data and to determine their position and orientation. The use of low-resolution 3D range cameras for obstacle avoidance and crane load docking are also being investigated (Saidi and Lytle, 2008). *Computer Integrated Road Construction* (*CIRC*) project has been aiming at introducing a new generation of control and monitoring tools for road pavement construction. Two prototypes have been developed: *CIRCOM* for compactors (Bouvel et al., 2001), and *CIRPAV* for asphalt pavers (Peyret et al., 2000). Figure 2-25 shows a compactor instrumented with a GPS antenna, a gyro, a radar, etc.

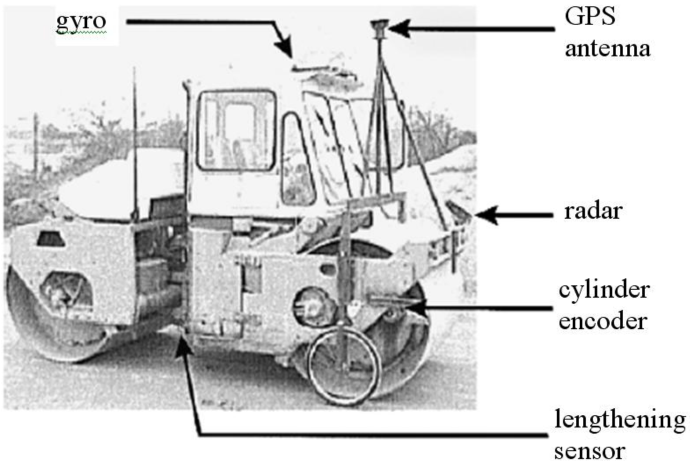


Figure 2-25: Instrumented compactor (Bouvel et al., 2001)

Unmanned and semi-automated construction systems could be used not only at disaster restoration sites, but also to increase safety and efficiency at ordinary construction sites. However, it is mentioned that the efficiency of unmanned construction is roughly 60% to 70% of that of manned construction, but sharply decreases in cases where the machinery moves or high precision work is necessary (Ban, 2002). Therefore, the full automation of heavy equipment is unnecessary in construction projects. Artificial intelligence (AI) methods can be used as an auxiliary tool to support the equipment operators, as is explained in Section 2.5.

There are other applications of construction automation for a variety of purposes. For example, the RC Truck Club (RC-Truck-Club, 2010) in Switzerland has been working on automated construction equipment models. The Moon Outpost Project of NASA requires small robots to level sites and roads, to clear obstacles, and to construct berms (Moon construction, 2010). A lunar surface manipulation system (LSMS) is being developed by NASA's Exploration Technology Development Program Office. It is designed for lifting purposes on the moon, where it can operate autonomously, can be remotely operated from a base, or can be operated manually (NASA, 2010). The Canadarm (2009), which is a crane used in space, is significantly more sophisticated than many of the ones found on Earth. It was built in Canada and sponsored by the Canadian Space Agency and used on the space shuttle to transfer cargo and to release satellites. A remote control system for large track-type tractors has been developed by Caterpillar to improve safety and productivity (Caterpillar, 2010). Critical information normally displayed in the cab (in gauge clusters and on display panels) is replicated on the remote console. An immediate stop is applied in emergency cases to enhance safety.

2.5 AGENT TECHNOLOGIES

The concept of agent comes from developing a thinking machine with the capability of solving a problem on its own. An agent can be a piece of software that is capable of accomplishing tasks on behalf of its user. AI has provided the foundation for computers to deal with complex tasks, such as monitoring and controlling industrial processes, assisting in medical diagnoses, or designing new machines. As Russell and Norvig (2003) have described, agents are relatively independent and autonomous entities that operate within communities in accordance with complex modes of cooperation, conflict, and competition in order to survive and perpetuate themselves.

Agents are capable of perceiving their environment, but only to a limited extent. By exchanging information with other agents, they can acquire more information about the environment. Actions are taken by the agent to satisfy its objectives based on some satisfaction/survival function which it tries to optimise using its skills. The actions carried out by an agent change the agents' environment and thus its future decision making. Agents are endowed with autonomy, which means that they are not directed by commands coming from a user, but by a set of tendencies, which can take the form of individual goals to be achieved or of satisfaction or survival functions which the agent attempts to optimise.

Subsection 2.5.1 briefly introduces Multi-Agent Systems (MAS) research. Subsection 2.5.2 reviews planning in MAS, focusing on path planning. Subsection 2.5.3 briefly introduces communications between agents, and agent research in construction is reviewed in Subsection 2.5.4.

2.5.1 MULTI-AGENT SYSTEMS (MAS)

MAS is a branch of AI that aims to answer the following questions: How do agents cooperate? What methods of communication are required for them to distribute tasks and coordinate their actions? What architecture can they be given so that they can achieve their goals? The term *multi-agent system* is applied to a system comprising the following elements (Ferber, 1999):

(1) An environment space, *E*, which generally has a volume; e.g., a construction site;

(2) A set of objects, *O*. These objects are situated, which means it is possible at a given moment to associate any object with a position in *E*. These objects can be perceived, created, destroyed, and modified by the agents; e.g., equipment on the construction site;

(3) An assembly of agents, *A*, which are specific objects ($A \subseteq O$), representing the active entities of the system; e.g., a *Crane Agent* and a *Coordinator Agent*;

(4) An assembly of relations, *R*, which link objects (and thus agents) to each other; e.g., priority among different equipment and relationships among groups;

(5) An assembly of operations, *Op*, making it possible for the agents of *A* to perceive, produce, consume, transform and manipulate objects from *O*; e.g., sensing, data processing, message exchanging, and control; and

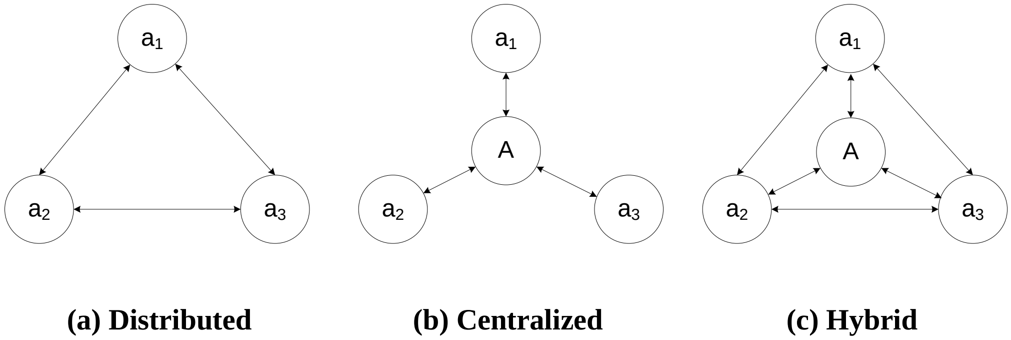
(6) Operators with the task of representing the application of these operations and the reaction of the world to this attempt at modification; e.g., commands of executing a motion plan.

MAS can be applied to numerous areas, such as problem solving, multi-agent simulation, building artificial worlds, collective robotics, and so on. Among them, problem solving actually concerns all situations in which software agents accomplish tasks that are of use to human beings (Ferber, 1999).

2.5.2 PLANNING IN MULTI-AGENT SYSTEMS

There are several ways of planning for MAS either in a centralised or a distributed manner. The centralised method treats the entire team as a single complex agent and then generates plans for this agent whereas the distributed method generates plans for individual agents and uses coordination techniques to combine these plans. Due to the intelligence of agents, each agent can generate a partial plan independently and the coordination of these partial plans can be centralized or distributed to form a single coherent overall plan (Ferber, 1999).

Figure 2-26(a) shows a distributed approach in which three agents communicate with each other and make decisions based on the result of their negotiation. Distributed problem solving involves multiple agents that combine their knowledge, information, and capabilities so as to develop solutions to problems that are difficult to solve by a single agent. An agent is unable to accomplish its own tasks alone, or it can accomplish its tasks better (more quickly, completely, precisely, or certainly) when working with others. Durfee (1999) has discussed the motivations for using a distributed problem-solving approach. These motivations are the following: (1) using parallelism, problem solving can be accelerated, (2) expertise or other problem-solving capabilities can be inherently distributed, (3) data are distributed, and (4) the results of problem solving or planning might need to be distributed in order to be acted upon by multiple agents (Durfee, 1999).



(a) Distributed **(b) Centralized** **(c) Hybrid**

Figure 2-26: Different approaches for planning

Figure 2-26(b) shows a centralized approach where *A* is acting as a team coordinator to communicate with the team members and is responsible for producing an overall plan.

The team members transmit data to the coordinator to form a global view. However, this centralized approach may cause tremendous amounts of unnecessary communication compared to allowing the exchange of information directly among team members. Moreover, the complexity of the problem increases rapidly with the size of the team or the DoFs of the equipment; therefore, centralized approaches are typically used in dealing with small teams or simple problems. A variation of the centralized approach is that team members draw up their own partial plans independently and send them to the coordinator. Then the coordinator tries to synthesize all the partial plans into an overall plan by solving the contradictions among the partial plans. An example is that of using a distributed approach to generate paths for each individual agent. Then a centralised planner schedules the movement of all the agents along their respective paths to ensure there are no collisions (O'Donnell and Lozano-Pérez, 1989). Taking advantage of both centralised and distributed approaches contributes to the hybrid approach, as shown in Figure 2-26(c). Kalra et al. (2005) have discussed that the team can work faster if the team members make decisions more locally and achieve coordination via a mechanism that is light on both communication and computation. More complex interactions between teammates and a more complex coordination mechanism are needed for more complex scenarios. Two types of coordination mechanism are proposed: one for teammates acting in a self-interested manner and another for a team plan that consists of actions that its teammates could take. By doing so, the robots efficiently vet candidate solutions and choose the coordination mechanism that best matches the current demands of the task. Clark et al. (2003) have proposed a complex hybrid approach where centralised planning is performed for dynamically-formed subgroups of agents. This approach enables the agents to decide which subgroups to form based on the relative positioning of the agents in the team.

2.5.2.1 Path Planning in MAS

One of the major and complex planning problems in MAS is the path planning problem. Ferguson (2006) has addressed the problem of path planning and re-planning in realistic scenarios in the case of single agent and multiple agents. Centralized algorithms are explored to deal with the planning problems for teams of agents. To efficiently cope with the high-dimension state spaces involving multiple agents, RRT algorithms have been selected for path planning and re-planning. Tavakoli et al. (2008) have proposed a cellular automata-based algorithm for path planning in MAS with a centralized approach. Several geographically distributed agents with the same priorities move towards a common goal location. The proposed algorithm distributes the agents to avoid long queues where only a few possible paths are available towards the goal. These researchers have claimed that the new proposed algorithm is faster than the traditional A* when several agents have a common goal. Marsh et al. (2005) have introduced a simulation to test real-time path planning in a road network. Distributed architecture has been adopted to avoid system failure caused by the central agent failure. Each agent broadcasts its sensed data to other agents to reduce the completion time. Gireesh and Vijayan (2007) have proposed a fuzzy logic approach to secure a collision-free path avoiding multiple dynamic obstacles. A robot is equipped with several sensors, and decisions are taken at each step in the pre-defined path in the environment. Sud et al. (2007) have presented a novel approach by introducing a new data structure, called Multi-agent Navigation Graph, which is constructed from Voronoi diagrams. Simulation scenarios consisting of hundreds of moving agents, each with a distinct goal, are used to test the proposed approach for real-time multi-agent planning. In the above-mentioned literature, it can be seen that researchers have been actively exploring different methods and trying to effectively solve path planning problems by considering specific applications. There is no one method that can be thought of as superior in general; therefore, it is important to select a suitable method and improve it to solve our own research problem efficiently.

2.5.3 COMMUNICATION BETWEEN AGENTS

Communication between agents is essential for the coordination of the behaviour of the agents in time and space. Such communication basically requires the exchange of messages between agents. Wireless communication technologies are needed for agents to communicate with each other on site. Many types of wireless networks are available, such as wireless personal area networks (WPANs), wireless metropolitan area networks, and wireless local area networks (WLANs), also called Wi-Fi. Wi-Fi networks are able to solve many of the communication problems caused by the "islands of information" in construction (Lee and Bernold, 2008). As shown in Figure 2-27, the dotted lines show wireless communication between different components of an agent-based crane alert model.

One of the new Wi-Fi standards, known as 802.11n, supports actual data rates up to 100 Mbps. Another WLAN technique is ad-hoc wireless networking, in which some mobile devices are part of the network only for the duration of a communication session while they are in close proximity to the rest of the network. Yang and Hammad (2007) have investigated problems related to deploying ad-hoc wireless networks that support communication and onsite data collection. Hammad et al. (2009) have designed an outdoor distributed Augmented Reality (AR) system to support the interaction of two users operating two virtual cranes and communicating with each other by using an ad-hoc network as shown schematically in Figure 2-28.

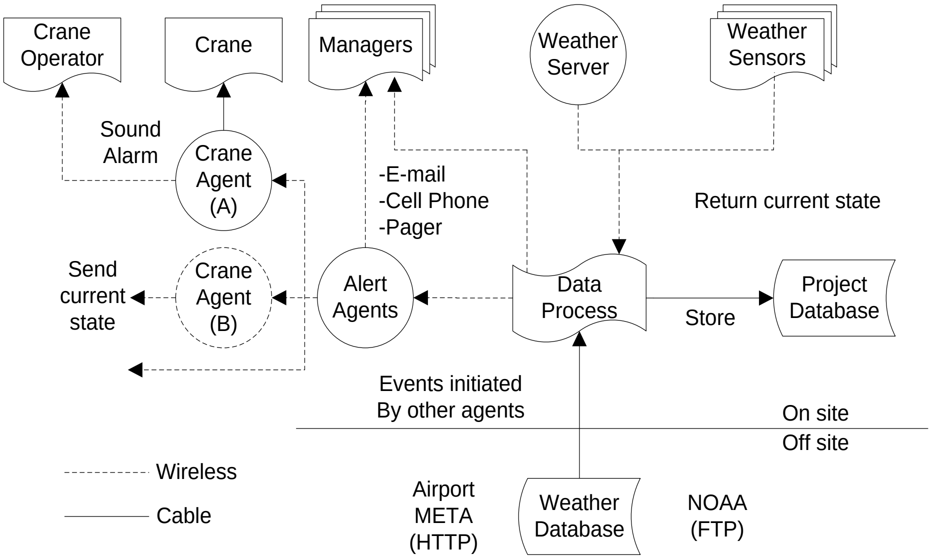


Figure 2-27: Agent-based crane alert model (Lee and Bernold, 2008)

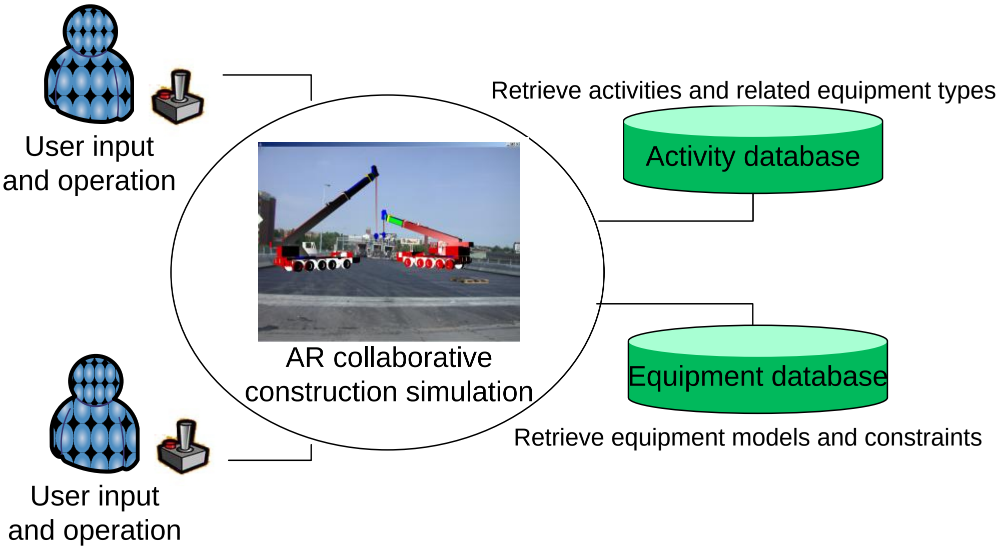

User input and operation
Retrieve activities and related equipment types
Activity database
AR collaborative construction simulation
Equipment database
Retrieve equipment models and constraints
User input and operation

Figure 2-28: Distributed augmented reality system for supporting multi-user interaction using ad-hoc wireless networking (Hammad et al., 2009)

2.5.4 AGENT SYSTEMS IN CONSTRUCTION

Some research involving agents has been done to enhance communication between team workers and to solve problems in the construction industry. For example, agent systems have been used for construction claims negotiation (Ren and Anumba, 2002) and dynamic rescheduling negotiation between subcontractors (Kim and Paulson, 2003). Bilek and Hartmann (2003) have presented an agent-based approach to support complex design processes in Architecture, Engineering, and Construction (AEC). Wing (2006) has presented some research on the application of software agents together with RFID technology in construction. Lee and Bernold (2008) have presented an agent-based communication system on site for collecting weather information and sending warning messages. To the best knowledge of the author, no research has focused on near real-time path planning of construction equipment operation using agents.

2.6 SUMMARY

A wide range of literature in civil engineering, computer science, and robotics areas are reviewed in this chapter, including simulation in construction, path planning for cranes, automation of construction equipment, available field data capturing technologies, and current research trends. Simulation in construction has the limitation of not supporting real-time applications, and the simulated environment is static and does not consider dynamic objects. Motion planning algorithms are studied in terms of discrete or continuous space searching to select the most suitable one for motion planning of cranes. Not much research has been done in motion planning for construction equipment. Safety issues have not been discussed enough in motion planning. Furthermore, real-time support is not available when potential collisions may occur. In this case, unknown and dynamic objects should be monitored and taken into account to ensure collision-free movement of the equipment. Therefore, environment perception technologies are reviewed to find the feasibility of tracking static and dynamic objects on site. Several RTLSs are investigated including RFID and UWB to track moving objects on site. UWB technology can provide a relatively high accuracy and applying UWB in construction is in a preliminary stage. Finally, agent technology is reviewed for the management of the collaboration of path planning of construction equipment. Different aspects of multi-agent systems have been reviewed. There are not much agent applications in construction to support decision-making.

CHAPTER 3 OVERVIEW OF PROPOSED METHODOLOGY AND REQUIREMENTS DEFINITION

3.1 INTRODUCTION

The present chapter introduces the methodology used in the present research. Section 3.2 provides a summary of the proposed methodology. Section 3.3 discusses the criteria used in the selection of the motion planning algorithm to meet the requirements of cranes, especially the re-planning phase, which needs a quick response on site to avoid obstacles and to re-plan a new path. In Section 3.4, the requirements of applying UWB technology for crane safety are defined to collect the necessary and accurate data in near real-time considering the advantages and limitations of the technology. The framework of a multi-agent system is proposed in Section 3.5 to aid the equipment operators with near real-time information exchange and decision-making.

3.2 PROPOSED METHEDOLOGY

Figure 3-1 shows the concept of the proposed methodology for the near real-time environment updating, motion planning and re-planning of cranes. During the planning stage, a 3D model of the static environment is available, and a collision-free motion plan is generated for the crane, taking into account engineering constraints and operation rules (Zhang et al., 2009). During the actual construction work, a UWB RTLS is used to capture on-site data. Multiple UWB tags with identification numbers (IDs) are attached to the different components of cranes and other equipment and workers, at predefined locations, to monitor their positions and orientations. Updated environment information is used to check the motion plan for any potential collision. In the case an obstacle is detected, the equipment involved is stopped to ensure safety and a new motion plan is generated in near real time according to the updated environment. Near real-time re-planning is defined as finding a new collision-free path of the crane based on the sensed data in a short period of time (a few seconds) after detecting the obstacles by the sensors. The short delay is caused by certain relatively low update rates of tags and the calculation time. To improve the safety of crane operations and to provide more awareness on site, the following important issues should be considered using the data collected from the RTLS.

(1) Identifying the poses of obstacles by using multiple tags attached to different components of equipment. Consequently, moving objects can be tracked, identified, and modeled in such a way that the full geometry, speed, moving direction, and all the related information of the task are used to prevent collision accidents. Buffers are added to the obstacles for collision detection. The size of the buffer can be adjusted according to the accuracy and the update frequency of the UWB system, and to the moving velocity of the crane. Less accurate data, lower update frequency and higher velocity require selecting a bigger buffer around obstacles.

(2) Checking the compliance with safety regulations and engineering constraints to prevent accidents, as explained in Chapter 1. By providing the feasible configurations of cranes within the capacity limits, the risk of tip over can be reduced; by applying collision-detection for the next movement of the crane, the risk of electrocution can be reduced.

(3) Providing an advanced intelligent support by integrating motion planning algorithms to generate a collision-free path. Once a potential collision is detected, re-planning of the equipment motion can be carried out according to the updated environment information. The present research focuses on the crane boom movement when the crane base is fixed.

The present research focuses on near real-time data collection and processing using UWB technology. However, the proposed methodology can be extended to capture the progress of the construction work by accommodating other types of sensing technologies, such as laser scanners (Gordon and Akinci, 2005).

To provide effective near real-time intelligent support to the crane operators, agent-based technology (Ferber, 1999) is proposed that encapsulates knowledge, organizes the information, makes decisions and translates the motion plan into actions that can be applied by the operators. In a multi-agent system, agents carry out separate, but interdependent tasks to meet their final objective. Every agent needs to send and receive messages and to make decisions (such as changing priorities for motion planning and re-planning) based on near real-time on site situations. Communication between agents expands the perceptive capacities of agents by allowing them to benefit from the information and the know-how of other agents (Ferber, 1999). Each construction staff is represented as an agent in the system and gets support for the near real-time decision making.

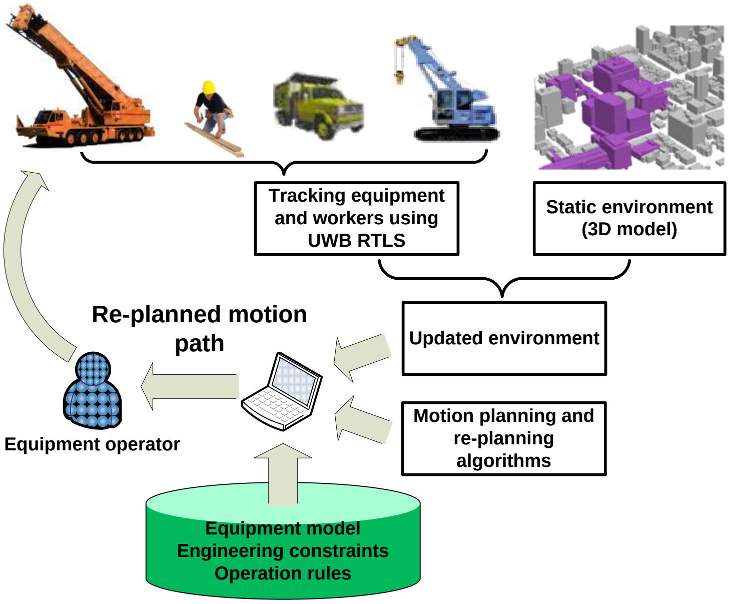

Tracking equipment and workers using UWB RTLS
Static environment (3D model)
Updated environment
Re-planned motion path
Motion planning and re-planning algorithms
Equipment operator
Equipment model
Engineering constraints
Operation rules

Figure 3-1: Conceptual near real-time environment updating and intelligent assistance for cranes

The motion planning and re-planning algorithms selection criteria are discussed in Section 3.3, and the details of the algorithms are described in Chapter 4. The requirements of using UWB technology in construction for crane safety are defined in Section 3.4. The details involved in the application of the UWB system for near real-time data collection and processing are given in Chapter 5. The details about the proposed framework of the agent-based system are given in Section 3.5.

3.3 MOTION PLANNING AND RE-PLANNING FOR CRANES

As discussed in Section 2.3, most implementations of motion planning algorithms are assisted by appropriate domain heuristics to find a good/optimal path within a reasonable time, and no industry-wide standard exists for heavy lift path planning practices. Consequently, experts rely primarily on experience to develop the plans or to perform optimization. The present research focuses on investigating a new method for crane motion planning and re-planning by adapting robotic motion planning algorithms, taking into consideration the engineering constraints of cranes.

3.3.1 CRANE MODELING

To apply motion planning algorithms, a crane is modeled as a robot composed of a series of links (rigid bodies) connected by joints that allow the relative motion of neighboring links. Using this robotic model, kinematic properties are defined including the following: the hierarchal structure of the links, the local coordinate systems (frames), and the joint type, which is either a sliding joint (prismatic joint) or a rotational joint (revolute joint). These properties can be defined mathematically in a homogeneous transformation matrix using DH-notation (Denavit and Hartenberg, 1955), as shown in Figure 3-2.

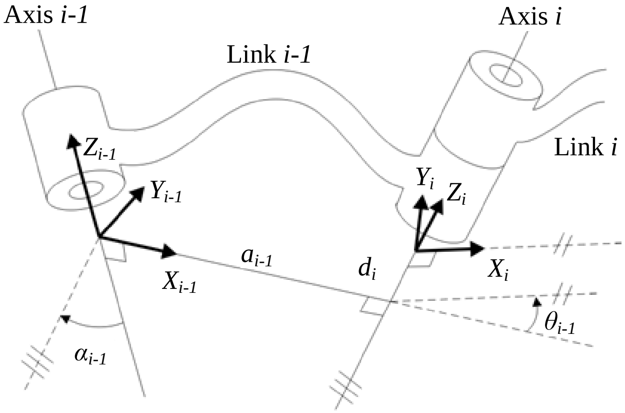


Figure 3-2: Relationship between links in D-H notation (Craig, 2004)

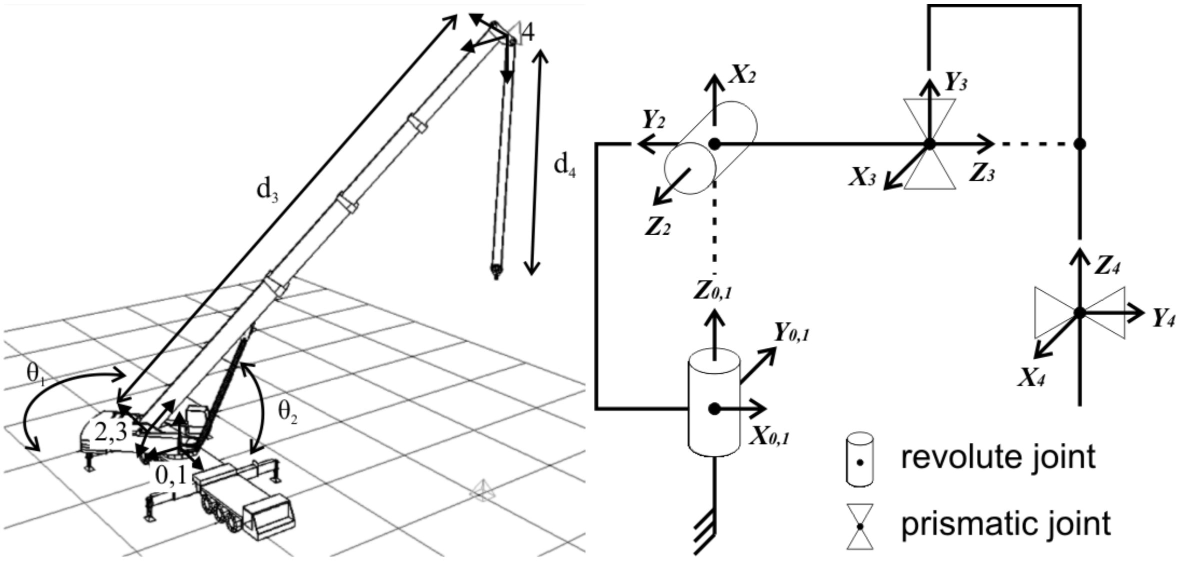
The relationship between Link i-1 and Link i is essentially the transformation matrix between coordinate system $\{i\text{-}1\}$ and coordinate system $\{i\}$. This matrix can be represented in a homogeneous transformation matrix ${}^{i\text{-}1}T_i$ as follows:

$$ {}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3\text{-}1) $$

where $c\theta_i$ represents $cos(\theta_i)$, $s\theta_i$ represents $sin(\theta_i)$, $s\alpha_{i-1}$ represents $sin(\alpha_{i-1})$, and $c\alpha_{i-1}$ represents $cos(\alpha_{i-1})$. Once the transformation matrix for each link has been developed, the forward kinematics function of the robot can be found by multiplying all the link transformation matrices. The result is a computational model that is used to control the simulation model.

As discussed in Subsection 2.3.1, the number of DoFs of a crane defines the dimensions of the *C-space*. Therefore, the greater the number of DoFs considered, the more complex the *C-space* is. For example, a loaded crane has a maximum of eight DoFs, and a path planning for manipulators having more than four DoFs is considered to be complex (Hwang and Ahuja, 1992). Figure 3-3 shows the kinematic modeling of a hydraulic crane for which four DoFs are defined and for which there are two revolute joints (the swing of the boom θ_1 and the angle to the ground θ_2) and two prismatic joints (the boom extension d_3 and the cable extension d_4) (AlBahnassi and Hammad, 2010). A local coordinate system is attached to each joint. This kinematic model includes an additional motion constraint that re-orients the cable along the gravity vector as the boom rotates up and down. This motion constraint avoids having an additional revolute DoF for controlling the orientation of the cable with respect to the boom (θ_4); in that case, the *C-space* becomes a five-dimension space, and the configuration vector is q = (θ_1, θ_2, d_3, θ_4, d_5) while the controllable DoFs are only (θ_1, θ_2, d_3, d_4). This leads to solving the motion planning problem as if it is non-holonomic, a condition which is not true because the cable motion constraint can be expressed as a configuration constraint, as follows:

$$\theta_4 = 90^{\circ} - \theta_2 \tag{3-2}$$



(a) Frames attached to the crane components

(b) Schematic for the hydraulic crane based on DH-notation

Figure 3-3: Defining the kinematic structure for a hydraulic crane (AlBahnassi and Hammad, 2010)

Once the transformation matrix for each link has been developed, the forward kinematics function of the hydraulic crane can be found by multiplying all four joint transformation matrices. Since the matrix is essentially transferring the coordinate system from $\{0\}$ to $\{4\}$, we denote it by ${}^{0}T_{4}$ as shown in Equation (3.3).

$${}^{0}T_{4} = \begin{bmatrix} c\theta_1 c\theta_2 & c\theta_1 s\theta_2 & -s\theta_1 & -d_4 s\theta_1 + d_3 c\theta_1 s\theta_2 \\ s\theta_1 c\theta_2 & s\theta_1 s\theta_2 & c\theta_1 & d_4 c\theta_1 + d_3 s\theta_1 s\theta_2 \\ s\theta_2 & -c\theta_2 & 0 & -d_3 c\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

where $c\theta_1$ represents $cos(\theta_1)$, $c\theta_2$ represents $cos(\theta_2)$, $s\theta_1$ represents $sin(\theta_1)$, $s\theta_2$ represents $sin(\theta_2)$, θ_1 represents the swing of the boom, θ_2 represents the angle to the ground, d_3 represents the length of the boom, and d_4 represents the length of the cable. The matrix ${}^{0}T_4$ is a homogeneous transformation matrix, which can be used to represent both the orientation and the position of the cable with respect to the coordinate system $\{0\}$.

3.3.2 CRITERIA FOR SELECTING THE MOTION PLANNING ALGORITHM

Several criteria are taken into consideration in selection of the motion planning algorithm of the cranes. There are four major criteria that are taken into account in the present research. They are the following: efficiency, optimality, reusability, and safety.

(1) ***Efficiency***: Path planning has been proven to be a hard problem (Reif, 1979). In the last decade, more interest has grown in developing practical path planners (Latombe, 1991; Barraquand et al., 1997). These planners embed weaker notions of completeness (e.g., probabilistic completeness) and/or can be partially adapted to specific problem domains in order to boost performance in those domains. In re-planning, efficiency is the most important factor because decisions usually need to be taken in near real time to cope with the dynamic nature of the environment. Based on the literature review in Subsection 2.3.2, Rapidly-Exploring Random Trees (RRTs) have been shown to be effective for solving single-shot path planning problems in complex *C-spaces* by combining random sampling of the *C-space* with biased sampling around the goal configuration. RRTs efficiently provide solutions to problems involving vast, high-dimensional *C-space*. These solutions would be intractable using deterministic approaches.

(2) ***Optimality***: Optimality is considered as the ability to find an optimal path with respect to some metrics. Single-query sampling-based algorithms are not able to guarantee the generation of an optimal path based on pre-defined criteria. An optimization updates are required to address this point (LaValle, 2006). The basic RRT algorithm does not take path quality into account during its search, which may produce paths that are grossly suboptimal (Ferguson, 2006). To improve the quality of the solution path, Urmson and Simmons (2003) have proposed modified RRT algorithms that take the cost of the path into account. Berg et al. (2006) have considered adding the cost of the path in navigating a mobile robot.

(3) ***Reusability***: This requirement is specific to motion re-planning when a new obstacle appears. An efficient re-planning algorithm should be able to plan optimal traverses in near real time by incrementally repairing the paths of the equipment as new information is discovered. Re-planning algorithms should focus on the repairs to significantly reduce the total time required for the initial path calculation and subsequent re-planning operations (Stentz, 1995). Deterministic re-planning algorithms such as D* efficiently repair previous planning solutions when changes occur in the environment (Choset et al., 2005). They do this by determining which parts of the solution are still valid and which parts need to be recomputed. However, as the number of the dimensions of the search space increases, for example, in the case of multiple cranes working together, deterministic algorithms simply cannot cope with the size of the corresponding state space. On the other hand, RRT algorithms abandon the original tree and grow a new RRT path from scratch if the environment is dynamic. This can be a

very time-consuming operation, particularly if the planning problem is complex. In such a case, researchers have started implementing the Dynamic RRT (DRRT) algorithm as a probabilistic analog to D* for navigation in unknown or dynamic environments (Ferguson, 2006). DRRT depends on repairing the current RRT when new information concerning the *C-space* is received instead of abandoning the current RRT entirely. DRRT removes the invalid part of the path and grows the remaining tree until a new solution is found (Ferguson et al., 2006).

(4) ***Safety***: To ensure the generation of safe paths for cranes, the algorithm needs to consider the engineering constraints of the crane. Taking the motion planning of hydraulic cranes as an example, the working range, the load charts, and the rules of action should be followed to ensure on-site safety.

3.3.3 PROPOSED MOTION PLANNING ALGORITHM

Based on the criteria for selecting the motion planning algorithm discussed in Subsection 3.3.2, RRTs have been shown to satisfy most of those criteria. For that reason, RRT algorithms have been selected as the basic algorithms for motion planning in the present research, but modifications have been made to improve the performance and to take into consideration the specific requirements of cranes.

There are many variations of RRT in terms of improved efficiency and path quality. For example, multiple trees are used to reduce the planning time and to improve the success rate of finding a path. The aspect of cost function for evaluating path quality is applied to produce a path with better quality. Clifton et al. (2008) have evaluated the performance of multiple RRTs. They have indicated that, when many obstacles are present, the time required to find a complete path can increase significantly; hence, in the present research, a dual-tree structure is used for the motion planning algorithm, thereby reducing the planning time. Building two trees starting from the initial and goal states separately aims to precisely select the goal state instead of assuming that the goal state will be reached by generating nodes randomly, especially when a manipulator must reach a specific end-effector position and orientation.

One of the advantages of using an RRT algorithm is that it does not need an explicit representation of the *C-space* (Choset et al., 2005). It is enough to define the range of each DoF of the equipment based on the engineering constraints, for example, the load charts and working ranges of a crane. As introduced in Subsection 2.3.1, the configuration of a crane is a complete specification of the position of every point of that crane. The *C-space* of the crane is the space of all possible configurations of the crane; and a configuration is simply a point in this abstract *C-space*. Unlike robotics, the engineering constraints of a crane further narrow down the *C-space* into a feasible space, thereby fulfilling the feasibility of the movement according to the load charts and the working ranges of the crane. Therefore, the feasible *C-space* can be defined according to the crane-specific conditions, e.g., the lift weight, the counterweight, and the outrigger radius. In this way, safe and realistic motion plans for cranes are generated by taking into account the engineering constraints.

Path quality improvement is also investigated in the present research by evaluating the smoothness of the path and execution time. The smoothness value of each node can be calculated. The nodes with better smoothness are selected and connected to the tree. The details of the proposed motion planning algorithm are given in Chapter 4.

When a random node is generated in the feasible *C-space*, the position of each component of the crane in the work space is defined, and then a collision detection algorithm is used to detect whether there is any collision between the crane and the obstacles on site. Current collision detection methods applied in robotics and computer graphics are generally more complex than necessary for construction purposes and are relatively difficult to implement efficiently (Kang and Miranda, 2006). Therefore, bounding boxes of the components of the crane are considered enough for collision detection with objects in the environment. Buffers added to the bounding boxes can ensure the safety on site in near real-time planning, as mentioned in Section 3.2.

3.3.4 PROPOSED MOTION RE-PLANNING ALGORITHM

Planning motions for navigating the crane on the actual construction site is more challenging than planning motions off-line. The main challenges come from incomplete or imperfect information, limited deliberation time, and the dynamic environment (Ferguson, 2006). In the present research, multiple tags are attached to different components of objects, as explained in Section 3.4; therefore, the poses of an obstacle are assumed to be fully known and are used for collision detection.

As discussed in Subsection 3.3.2, DRRT depends on repairing the current RRT when new information concerning the *C-space* is received instead of abandoning the current RRT entirely. DRRT efficiently removes just the invalid part of the path and grows the remaining tree until a new solution is found (Ferguson et al., 2006). However, the DRRT algorithm is based on growing one tree, and the direction of growing the tree is reversed to reuse the previous tree when the current configuration changes. This algorithm does not support the case of growing two trees rooted at the initial configuration and the goal configuration. Furthermore, when an obstacle is detected, the whole tree is checked to mark invalid nodes, and then the solution path is checked to see if it contains any of the invalid nodes. If invalid nodes are found, the whole tree is trimmed and re-grown. It is time consuming to trim the entire tree for all new obstacles that may affect far later nodes on the path. Some of the detected obstacles can become non-obstacles at a later time because they are continuously moving in space.

In the present research, a dynamic motion re-planning algorithm is proposed, which starts by executing a motion plan generated during the planning phase. Each movement of the crane is checked for potential collision. If it is collision free, the crane is moved to the next configuration. Otherwise, the remaining path is checked to find collision-free nodes and rebuild two trees between the current crane configuration node and the collision-free node. If a path between these two nodes is found successfully, this partial path is combined with the remaining path and the crane resumes the movement to execute the new motion plan. This algorithm ensures the safety of the crane throughout its movement on the path considering the near obstacles and it ignores other obstacles that are far away from the current node on the path since these obstacles may move before the crane reaches that part of the path. Thus, this algorithm reduces the time for trimming the whole tree. Details of the motion re-planning algorithm are given in Section 4.6.

3.4 REQUIREMENTS OF UWB TECHNOLOGY FOR CRANE SAFETY

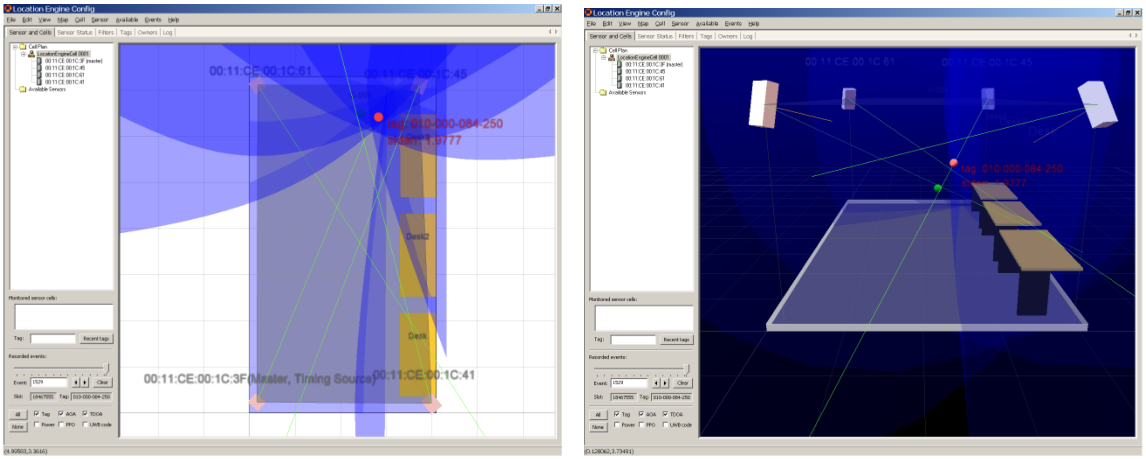
As explained in Subsection 2.4.2.2, UWB is becoming available for precise RTLS, which offers several distinct advantages over traditional tracking systems. The UWB RTLS is superior to other location systems in providing a long reliable readability range, offering accurate near real-time positioning, and being more robust to the multipath problems.

Based on the literature review and our experimentation with the Ubisense system (Ubisense, 2010), the present research examines the following requirements for the application of UWB in construction to improve safety: accuracy, visibility, scalability and real-time, tag form factors, power consumption, and networking requirements. The number of tags, the number of sensors, and the location and orientation of sensors should be decided to satisfy these requirements, as is discussed in Chapter 5.

(1) Accuracy requirement: Accuracy is the most important requirement guaranteeing that valuable data are collected. AOA and TDOA are used in UWB RTLS to locate tags based on trilateration. If only the AOA method is used, two sensors are theoretically enough to locate a tag in 3D; however, to improve accuracy, more sensors are needed in practice to reduce the influence of multipath and noise (Munoz et al., 2009). If only the TDOA method is used, at least three sensors are required for 2D positioning and four sensors are required for 3D positioning (Ghavami et al., 2004). Table 3-1 summarizes the combinations of the location methods and the results. The combination of AOA and TDOA yields the highest possible accuracy (Abdul-Latif et al., 2007). With this combination, two sensors deliver a robust localization with an accuracy of up to 15 cm under ideal conditions (Ubisense, 2010). In practice, more sensors enable both a greater confidence in the accuracy and a higher availability, leading to a more robust solution. Figure 3-4 shows an example in which four sensors receive signals from a tag and locate its position. The AOA vectors originating from the tag towards the sensors are drawn in green, while the TDOA curves are drawn in blue. The intersection marks the position of the tag, shown as a red dot.

Table 3-1: Combinations of the location methods and the results (adapted from Ubisense, 2010)

Location method	Number of sensors detecting tag	Other information required	Result
Single-sensor AOA	1	Known height of tag	2D horizontal position (+ known height)
AOA	2 or more	None	3D position
TDOA only	3	Known height of tag	2D horizontal position (+ known height)
TDOA only	4 or more	None	3D position
TDOA+AOA	2 or more	None	3D position (highest accuracy)



(a) 2D view

(b) 3D view

Figure 3-4: Four sensors locating a tag in a room

To gain accurate location data, calibration of the sensors is essential. A local coordinate system is defined and the coordinates of each sensor are measured precisely using surveying tools, such as a total station. Each sensor should be levelled after the installation with a zero roll angle. One tag should be placed at a location with known coordinates in the local coordinate system. As a result, the pitch and yaw angles (Figure 3-5) of each sensor can be calculated and recorded in the system.

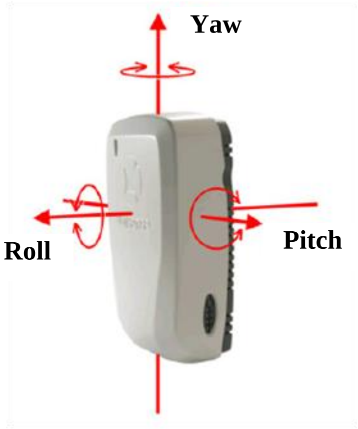


Figure 3-5: Yaw, Roll and Pitch angles of a sensor (Ubisense, 2010)

Moreover, data filtering should be applied in real time to reduce errors and improve the accuracy. This filtering can validate the individual AOA and TDOA measurements against a predicted position, and then these measurements can be used to calculate a new estimate of the position. The motion model for the filter has to be defined by specifying the constraints on the motion that the tracked object can undergo. For example, a tag can be free to move in 3D or can be constrained to move horizontally with a certain motion model of position and velocity and Gaussian noise on velocity (Ubisense, 2009). Filtering can be also applied on the location data resulting from the trilateration. For example, Cho et al. (2010) have claimed that the total accuracy is improved by 25% after applying an error model using a Kalman smoother. However, the application of these filters is based on several assumptions about the motion model. Applying these assumptions is not easy in the case of the movement of cranes.

(2) Visibility requirement: The sensors should be set in a way to utilize their field of view (FoV) both in the azimuth and the elevation. The FoV may be different from one UWB system to another. The maximum range of sensors can be potentially up to 100 m; therefore, a reasonable monitoring area should be defined taking into consideration the coverage of the sensor cell. If the area to cover is very big, more sensors should be installed to cover the whole area divided into two or more cells. In addition, attaching multiple tags to the same object should be considered as a way to improve the visibility of that object by increasing the probability of detecting the tags. For example, multiple tags can be attached to a worker's hardhat or the boom of a crane as is explained in Subsections 5.4.2 and 5.4.3.

(3) Scalability and real-time requirements: Since in commercial UWB systems there is only a single UWB channel used in time division mode, only one tag can be located at a time in each sensor cell. As mentioned in the visibility requirement, multiple tags can be used even for an individual object; therefore, the suitable number of tags attached to an object should be decided based on the frequency of the system and the size of the sensor cell. The number of time slots per second depends on the cell frequency of the UWB system. For the Ubisense system with a nominal cell frequency of R = 160 Hz, one second is divided into 153 time slots. Each has a duration of 6.5 ms. Different slot intervals can be selected to determine how often the tags' locations are updated, i.e., how often the system listens for data and schedules messages from the master sensor. The shortest slot interval can be set to 4 slots, which means the update interval is 26 ms, corresponding to a maximum update rate per tag of approximately 38 Hz (Ubisense, 2009).

With a large number of tags in a sensor cell, the update rate of tags decreases to allow the system to cover all tags with the fixed total number of time slots. For example, if the time slot is set to 4 and only 4 tags are in the cell, the four tags are updated every 26 ms (38 Hz). When more tags are detected in the cell, e.g., 8 tags, the update rate is decreased to 19 Hz. The more tags included in the cell, the bigger the required slot interval, and the lower the update rate. Figure 3-6 shows how the system assigns updates for 4 tags with a slot interval of 4 time slots. Table 3-2 shows the update intervals and rates for different slot intervals for Ubisense 160 Hz system (Ubisense 2010).

A specific update rate can be set for an individual tag or a group of tags. One consideration when setting the update rate is the moving velocity of the object. Objects with high velocity need more frequent updates to accurately track their traces. Therefore, it is essential to select a suitable number of tags with an appropriate update rate based on their velocity in order to achieve a balance between the conflicting requirements of visibility and accuracy in near real time.

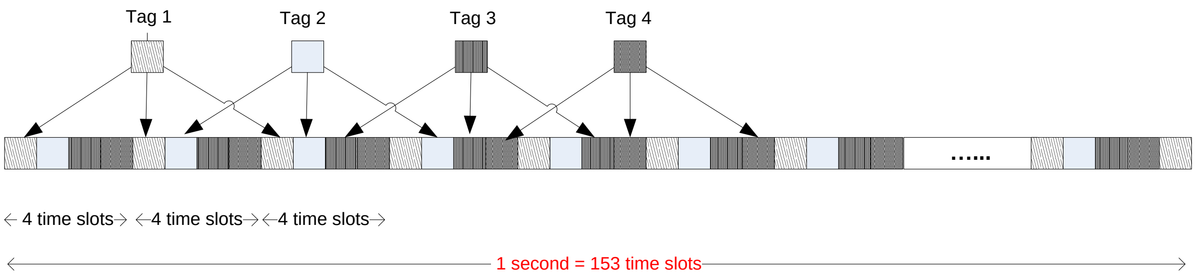


Figure 3-6: Tag updates for a 160 Hz system with slot interval of 4 time slots

Table 3-2: Update intervals and rates for different slot intervals for Ubisense 160 Hz system (adapted from Ubisense, 2010)

Slot interval	Update interval (ms)	Update rate for each tag (Hz)
4	26	38
8	52	19
16	104	10
32	208	5
64	416	2.4
128	832	1.2
…	…	…

To maximize the update rate for the Ubisense system, one heuristic rule can be defined as follows:

$$m \leq 4 \times 2^n \quad (3\text{-}4)$$

where m is the number of tags in the cell, n is the minimum value that meets the inequality, and 4×2^n is the time slot interval that should be set. For example, if there are 10 tags in the cell, the minimum value of n is 2; therefore, the time slot interval should be set to 16 (i.e., update rate of 10 Hz).

On the other hand, if the update rate is defined, another heuristic rule can be established as follows:

$$R/r \geq 4 \times 2^n \quad (3\text{-}5)$$

where R and r are the update rates of the cell and the tags, respectively, n is the maximum value that meets the inequality, and 4×2^n is the time slot interval that should be set. For example, if an update rate of r = 8 Hz is required for the tags in a 160 Hz system, the maximum value of n is 2, and the time slot interval can be set to 16. According to inequality (3-4), a maximum of 16 tags can be used in the system to obtain this update rate. Similar inequalities can be derived for other UWB systems.

As mentioned above, *r* should be set according to the velocity of the objects. For example, in the case of tracking a crane boom, if the velocity of the tip of the boom is 0.6 m/s, with a UWB system that has an accuracy of 15 cm, at least 4 Hz is needed to update the location of the boom's tip to avoid potential collisions.

(4) Tag form factor and function requirements: Even if the basic functionality of the tags is the same, tags come with different form factors. Some tags are specifically designed to be worn by a person as a badge; others are ruggedly designed to be attached to objects in a harsh environment. In addition to their tracking function, tags can include a buzzer or an LED to provide basic messaging functions and push buttons that trigger events. These tags can be used in safety applications in which, for example, a buzzer signal indicates that a worker is entering a dangerous zone. Specific examples of tags are given in Section 5.4.

(5) Power requirement: The sensors must be connected to a stable power source for precision measurements. Tags require a battery, the life of which depends on the update rate established for the system. The tag's update rate can be dynamically and automatically varied depending on the activity of the tag. If the tag moves quickly, a high update rate can be assigned for best tracking; if it moves slowly, the update can be reduced for best battery lifetime. When stationary, tags go into sleep mode to conserve power, and an in-built motion detector ensures that the tag transmits again as soon as it is moved.

(6) Networking requirement: The sensors can be connected by cables or wirelessly to the location server. Both data cables and timing cables are needed for a wired system. The length of the cables should not exceed the maximum length recommended by the manufacturer to avoid noise problems (Ubisense, 2009). The wireless system depends only on AOA calculations since wireless communication is not fast enough to support TDOA calculations. The choice of the type of the network (wired vs. wireless) has a direct impact on accuracy (Cho et al., 2010).

3.5 FRAMEWORK OF AGENT-BASED SYSTEM

The concept of the proposed methodology discussed in Section 3.2 can benefit from agent technology as discussed in Section 2.5. The framework of the proposed agent-based system is shown in Figure 3-7. This figure demonstrates the concept of a hybrid planning approach, according to which two *Crane Agents* plan their paths separately and a *Coordinator Agent* is used to coordinate the two cranes.

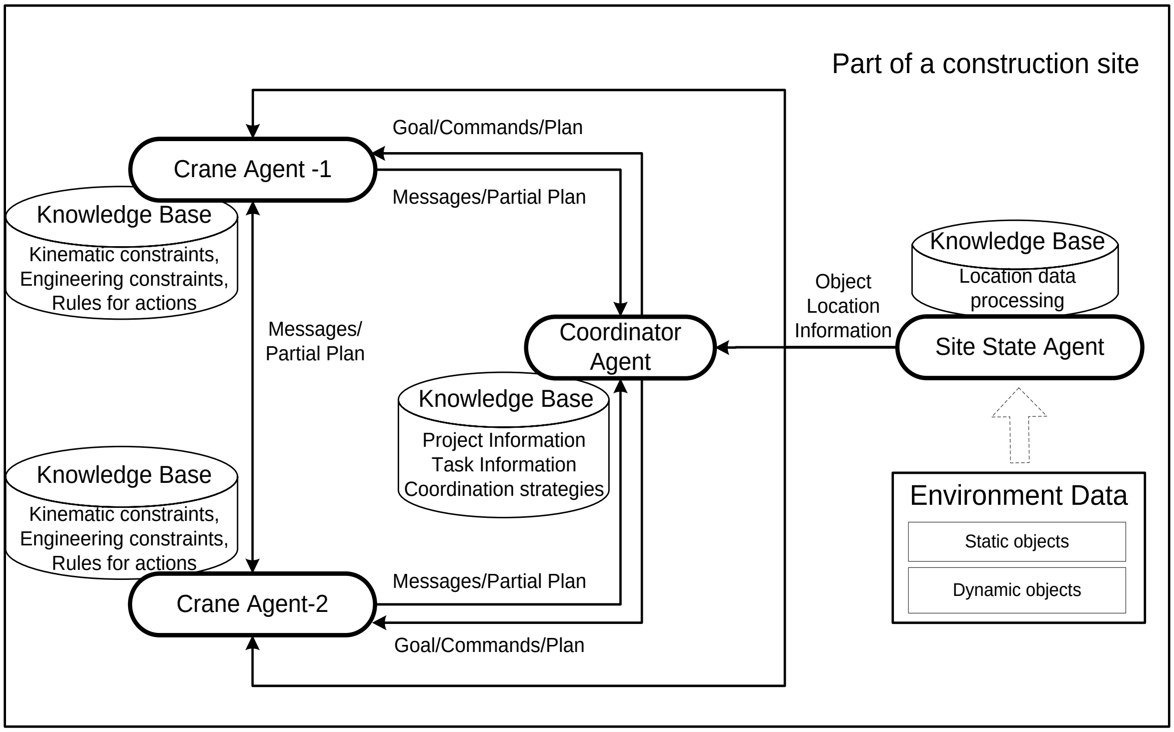


Figure 3-7: Framework of agent-based system

In a part of the construction site, several agents are involved in one or more tasks: *Crane Agent-1*, *Crane Agent-2*, *Coordinator Agent*, and *Site State Agent*. Each agent has a knowledge base, which consists of domain-specific knowledge that supports decision-making. The design of this framework assumes that the agents can be activated or deactivated by the system based on the physical locations of the objects they represent, i.e., inside or outside the monitored area, as discussed in Subsection 3.5.4.

3.5.1 CRANE AGENTS

A *Crane Agent* has the knowledge base that includes the kinematic constraints, the engineering constraints, and the rules for actions of the crane. Taking hydraulic cranes as an example, the kinematic constraints, i.e., the degrees of freedom (DoFs), can be defined according to the specifications. Engineering constraints are based mainly on the working range and load charts. The working range shows the minimum and maximum boom angle according to the length of the boom and the counterweight. Load charts give the lifting capacity based on the boom length, the boom angle to the ground, and the counterweight. Crane manufacturers and large construction companies usually have databases of the different cranes used in their work. These databases include the specifications about the different models of certain types of cranes. The D-Crane has a database that serves as a good example (Al-Hussein, 1999). The rules of actions are based on expert rules. For example, in the case of two cranes lifting together the same object, the combinations of hoisting and swinging or hoisting and luffing at the same time should be avoided (Shapiro et al., 2000). Tags are attached to different components of the crane (e.g., the boom, the hook, and the lift object) to monitor the poses (i.e., the position and orientation) of those components. These poses are used by the *Crane Agents* to detect potential collisions with obstacles on the path to ensure safety. The *Crane Agents* can communicate with each other and with the *Coordinator Agent* by exchanging messages or partial plans.

3.5.2 SITE STATE AGENT

The *Site State Agent* is responsible for collecting and processing data about static and dynamic objects on the construction site. Information about static objects includes the 3D model of the site created during the planning stage. The information can be updated when necessary. For example, newly built structures become obstacles for the next operation. Information about dynamic objects includes the positions of tags attached to moving objects on site, such as cranes, workers, and materials transported by the equipment. Location data are collected by the *Site State Agent* and processed into useful information to update the state of the environment model. As discussed in Section 2.4, several field data capture technologies have been proposed in recent years to create the 3D model of a construction site in near real time. Field data capture technologies include 3D imaging technologies (e.g., 3D scanners, 3D range cameras and photogrammetry) and radio-based identification and tracking technologies (e.g., RFID and UWB technologies). The quality of field data and the ability to capture in near real time decide the accuracy and feasibility of the system. The knowledge base of the *Site State Agent* includes location data processing algorithms. The *Site State Agent* classifies information for each object based on the tag IDs. Raw location data from the sensors are processed to describe the full geometry and poses of objects, as is explained in Chapter 5. For example, a simplified bounding box can be generated according to location data transmitted from multiple tags attached to the boom of a crane. Furthermore, based on the location of each object, the

Site State Agent decides to which agent the information of that object should be sent so that each agent gets the information necessary to ensure safety and to avoid overwhelming the communication bandwidth.

3.5.3 COORDINATOR AGENT

The knowledge base of the *Coordinator Agent* includes information about the project and task schedules (macro and micro levels), the operating cost of equipment, and the safety regulations. The knowledge base also includes coordination strategies to guide the movement of two cranes. Several strategies can be used including the leader-follower strategy (Zheng, 1989), time delay strategy (Chang et al., 1994), and speed alteration strategy (Hwang et al., 2003; Kamezaki et al., 2009). The *Coordinator Agent* works differently in the following two cases: (1) two or more cranes working together to lift one object; and (2) two or more cranes working on different tasks in the same area, where coordination is needed to avoid conflicts. In the first case, collaborative requirements limit the possible movement of each crane; accordingly, in a centralized approach to reduce collaboration complexity, the *Coordinator Agent* generates plans for the cranes based on the data sent. One important rule that should be considered in this case is that the distance between the two hooks should be equal to the length between the two attachment points, and crane load lines must be kept plumb at all times for multiple crane lifts (Shapiro et al., 2000).

In the second case, the *Coordinator Agent* is not responsible for motion planning. It only coordinates the work by deciding the priorities of the cranes. Once a potential conflict occurs, the *Crane Agents* communicate with the *Coordinator Agent* by exchanging messages and they make decisions based on negotiation in a hybrid approach. One efficient method to avoid collision is to adjust velocity instead of re-planning the path. As introduced in Subsection 2.3.3, Kang (2005) has proposed a decoupled method of planning for multiple cranes. Plans are generated for each crane separately by ignoring the other one during a short period δ, and then by coordinating the two cranes by tuning their relative velocities to avoid collisions. If successful, the system plans the next time period δ until the entire project is finished. Otherwise, a new δ is considered and steps are repeated for the entire project. The same method can be applied to re-planning when two *Crane Agents* can negotiate with each other to adjust their velocity and avoid collision. If there is no way to avoid collision by adjusting the velocities, re-planning should be done by one of the *Crane Agents*. In this case, the priority is decided by the *Coordinator Agent*, and the agent that has the lower priority has to re-plan the path for the corresponding crane. In the present research, only the second case (i.e., the hybrid approach) is considered.

During the plan execution stage, obstacles not taken into account in the planning stage are detected in real time. Agents are used to dynamically guide the actions of the equipment and to find collision-free paths respecting the engineering constraints and action rules. If re-planning is necessary, the priority of the *Crane Agents* should be decided according to the following scenarios in order to select which agent should re-plan the path:

(1) Safety-based priority: The equipment with critical safety issues has a higher priority. For example, the crane with the heavier load or the narrower work space should be given higher priority.

(2) Task-based priority: The tasks on the critical path have priority over other tasks. Based on the project schedule, the tasks on the critical path cannot be delayed because the whole project will be delayed. In this case, the *Coordinator Agent* should give the priority for the use of the work space to the equipment that executes the tasks on the critical path.

(3) Time-based priority: The equipment that has a shorter time in a given work space for its task should be given a higher priority for movement. For example, compared with the equipment working intensively in one workspace, the equipment that has a one-time access to the area is not to appear again in the same area has the priority to finish its task.

(4) Cost-based priority: The equipment with a higher operating cost has the higher priority in order to optimize the budget of the project.

(5) Alternating priority: If all the conditions are the same for both pieces of equipment, priority can be circulated between them so that each piece of equipment has the priority for a certain time period, e.g., one hour.

These priority scenarios have been identified according to our discussion with crane engineers. In certain cases, more than one priority rule can be applied resulting in a conflict between these priorities. However, these cases are beyond of the scope of the present research. Figure 3-8 describes a scenario of re-planning when two cranes have potential collision on their paths.

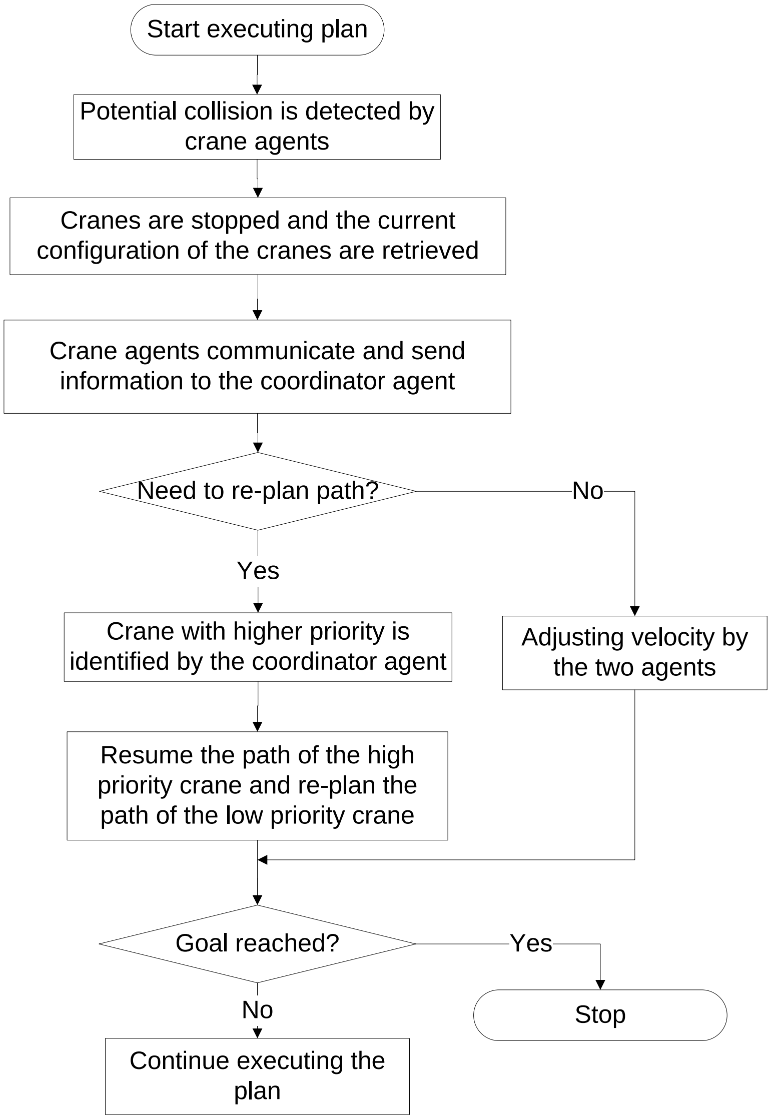

Start executing plan
Potential collision is detected by crane agents
Cranes are stopped and the current configuration of the cranes are retrieved
Crane agents communicate and send information to the coordinator agent
Need to re-plan path?
No
Yes
Crane with higher priority is identified by the coordinator agent
Adjusting velocity by the two agents
Resume the path of the high priority crane and re-plan the path of the low priority crane
Goal reached?
Yes
No
Stop
Continue executing the plan

Figure 3-8: Flowchart of re-planning scenario

The *Crane Agents* start executing their plans and continuously detect potential collisions for the next movements. Once a potential collision is detected, the *Crane Agents* send signals to stop the cranes and retrieve the current cranes' configurations. The *Crane Agents* communicate with the *Coordinator Agent* to get information about the obstacle and to send information about their paths and tasks to the *Coordinator Agent*. The *Coordinator Agent* decides whether re-planning is needed or adjusting velocity can solve the conflict. If there is no need for re-planning, the two *Crane Agents* negotiate with each other and with the *Coordinator Agent* to avoid collision by adjusting their velocities. In the extreme scenario of the case one crane fully stops and waits till the other crane leaves the conflict area. If re-planning is needed, the *Coordinator Agent* decides which crane has higher priority. The agent of the higher-priority crane resumes its path and the other agent re-plans the path of the other crane.

3.5.4 AGENT ACTIVATION

Tags with IDs are attached to moving objects, such as cranes, and are linked to the agents representing the specific objects they are attached to. Different parts of the construction site are monitored using different sensor cells. All the activities scheduled during a specific time period within a cell are retrieved from the project database. Accordingly, all the workers and equipment expected within the cells are identified and represented by agents in the system. Object identification is important since safety rules are generally applied differently to different object types (Chi and Caldas, 2009). In addition, the system monitors each object within the monitored area and initializes their agents when they are detected for the first time. Once the object leaves the monitored area, the corresponding agent is deactivated from the system, and the next time it enters the area, the agent is activated again. Information about an object can be retrieved from its agent, such as its ID, its tasks, and the duration of its task, and possibly the path of the object.

3.5.5 COMMUNICATION AND NEGOTIATION BETWEEN AGENTS

The communication is limited to agents within a part of the construction site where a task is carried out. This partitioning of the site space is necessary to avoid communication bottlenecks. Furthermore, because the dynamic agent system activates and deactivates agents based on the boundary of the monitored area, ad-hoc wireless networking is a good solution for the proposed method, as explained in Subsection 2.5.3. Based on the location of each object, the *Site State Agent* decides to which agent the information of that object should be sent. The *Coordinator Agent* communicates with all *Crane Agents* that are under its control and receives messages and partial plans from these agents. In addition, it decides the priorities for movement if any conflict occurs, and it sends commands to the *Crane Agents* to avoid collisions. A *Crane Agent* can also communicate with the other *Crane Agent* to inform the path of the crane for re-planning.

Negotiation between agents occurs in two scenarios: (1) if potential collision is detected between the two cranes, the two *Crane Agents* negotiate with each other and adjust the velocity of the crane boom to avoid collisions; (2) negotiation also happens when a *Crane Agent* rejects the decision made by the *Coordinator Agent*. The *Crane Agent* may suggest other options based on its own interest. The *Coordinator Agent* selects the best one or adjusts it using coordination strategies.

3.5.6 ACTIONS BASED ON MOTION PLAN

The motion plan is represented by a series of configurations that the crane needs to take in a sequence to achieve the goal. The initial configuration and the goal configuration of the crane should be defined according to the task (i.e., the initial and goal locations of the lift object). In order to help the crane operator, the configuration of each step on the path should be translated into a series of actions that can be understood by the operator, such as the instruction to swing the boom clockwise by 10 degrees. Taking a hydraulic crane as an example, the movement of the crane during lifting includes the following actions:

Boom movement: BoomUp, BoomDown, BoomExtend, BoomRetract, BoomSwingClockwise, BoomSwingCounterclockwise, BoomStop;

Hook movement: HookUp, HookDown, HookAttach, HookRelease, HookStop.

Each of these actions is quantified by a value, e.g., BoomSwingClockwise (10 °).

Based on the actions taken by the equipment, the configurations of the crane can be translated into states in the physical work space for collision detection. For example, at time *j*, the state of crane *i* can be represented as CraneState$_{ij}$ = [P_{ij}, Φ_{ij}, θ_{ij}, α_{ij}, l_{ij}, H_{ij}], which means that crane *i* is at location P_{ij}, with base orientation Φ_{ij}, boom swing angle θ_{ij}, boom angle to the ground α_{ij}, boom length l_{ij}, and hook position H_{ij}. Figure 3-9 shows a simple example for the movement of crane *i* from one state to another by raising its boom by $\Delta\alpha$ and by swinging its boom clockwise by $\Delta\theta$.

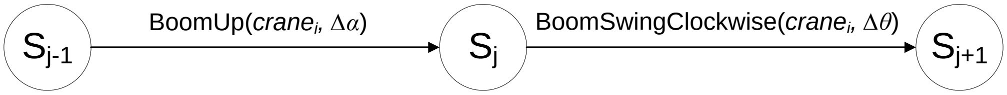

S_{j-1}
BoomUp(*crane*$_i$, $\Delta\alpha$)
S_j
BoomSwingClockwise(*crane*$_i$, $\Delta\theta$)
S_{j+1}

Figure 3-9: Actions and states changes

3.6 SUMMARY AND CONCLUSIONS

The present chapter has presented an overview of the proposed methodology and defined several requirements for the use of this methodology in construction to improve crane safety. The following issues have been discussed for the motion planning and re-planning of cranes: (1) Crane modeling has been described in detail following the robotic kinematic properties to apply motion planning algorithms; (2) Four criteria, which are efficiency, optimality, reusability, and safety, have been defined for the selection of the appropriate motion planning algorithm for cranes. Based on these criteria, RRT algorithms have been selected as the basic algorithms used in this research due to their quick calculation time and the ability of dealing with high dimensional problems; (3) Motion planning and re-planning for cranes have been described briefly to give an overview of the proposed methodology.

On the other hand, the requirements of the deployment of UWB RTLS on construction sites have been defined. These requirements are accuracy, visibility, scalability, and real-time requirements, tag form and function requirements, power, and networking requirements. These requirements have been defined based on the literature review and on our experience acquired by using a UWB system. Heuristic rules have been defined to balance the requirements of visibility, scalability, and real-time by clarifying the relationship between the number of tags, the update rate, and the velocity of objects.

Crane motion planning and real-time environment updates have been integrated by developing a framework of an agent-based system to improve the safety of cranes. This framework has several agents supporting the crane operations. The functionalities of the *Crane Agents*, *Coordinator Agent*, and *Site State Agent* have been described in detail for sensing, communication and decision-making. The main characteristics of the agent-based system have been described as the following: (1) A hybrid approach has been used in the system to gain the flexibility of distributing motion planning to each *Crane Agent* based on the priorities decided by the *Coordinator Agent*; (2) Priority patterns have been defined to decide which agent should re-plan the equipment’s path to avoid potential collisions; and (3) In order to guide the crane operators, motion plans can be translated into actions based on the crane’s configurations.

CHAPTER 4 MOTION PLANNING AND RE-PLANNING

4.1 INTRODUCTION

Based on the discussion in Section 3.3, RRT algorithms have been selected as the basic algorithms used in this research for crane motion planning and dynamic re-planning in near real time. In the present chapter, a more detailed review of the RRT algorithms and their different variations for improving planning time and path quality is presented. In this research, dual-tree RRT algorithms have been used to gain efficiency while ensuring safety by taking into account crane-specific engineering constraints. In addition, path smoothness is considered in this research to provide a realistic path for cranes and to reduce unnecessary movements. The proposed algorithms are compared with other available algorithms to evaluate their performance in terms of planning and re-planning time and the cost of the path. Based on the literature review, this is the first time that the dual-tree RRT algorithm has been applied to crane motion planning.

4.2 BASIC RRT ALGORITHM

RRT algorithms incrementally construct a search tree rooted either at an initial configuration q_{init} or a goal configuration q_{goal}. As shown in Figure 4-1, at each iteration from 1 to m (lines 2 to 4), a random configuration, q_{rand}, is sampled uniformly in the search space. The nearest configuration, q_{near}, to q_{rand} in the tree is found and an attempt is made to extend the tree, and finally connect q_{init} and q_{goal}. This method was originally developed by LaValle (1998).

```
Build-RRT($q_{init}$)
  $T$.add($q_{init}$);
  for $i$ = 1 to $m$ do
    $q_{rand}$ = Choose-Target();
    Extend($T$, $q_{rand}$);
  return $T$;

Choose-Target()
  return Random-Node();

Extend($T$, $q$)
  $q_{near}$ = Nearest-Neighbor($q$, $T$);
  $q_{new}$ = New-Node($T$, $q_{near}$, $q$);
  if Collision-Free($q_{new}$) == true
    $T$.add($q_{new}$);
    if $q_{new} \approx q$ then
      return Reached;
    else
      return Advanced;
  return Trapped;
```

Figure 4-1: The basic RRT algorithm (adapted from LaValle and Kuffner, 2001)

In this basic RRT algorithm, the *Extend* function selects the node q_{near} in the tree that is nearest to the sampled node q_{rand}, as shown in Figure 4-2. Then a motion toward q_{rand} with some fixed incremental distance ε is applied. If this motion is collision-free, a new node q_{new} is added to the tree and the *Extend* function returns one of the following two values: *Reached* if q_{rand} is reached or *Advanced* if q_{rand} is not reached. If q_{new} is not collision-free, a *Trapped* value is returned.

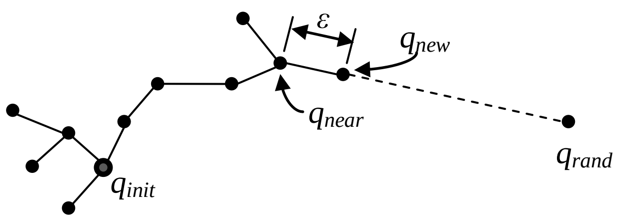


Figure 4-2: The Extend operation (LaValle and Kuffner, 2001)

4.3 DUAL-TREE ALGORITHMS

Several variations of RRT have been developed to reduce the planning calculation time. For example, Clifton et al. (2008) have evaluated the performance of multiple RRTs, and they have indicated that when many obstacles are present, the time required to find a complete path can increase significantly; therefore, having more than one tree growing simultaneously can greatly reduce the calculation time. Two of these algorithms are discussed in the following.

4.3.1 RRT-CONNECT ALGORITHM

Kuffner and LaValle (2000) have proposed the RRT-Connect algorithm, in which two trees rooted at the initial configuration and the goal configuration are built, as shown in Figure 4-3.

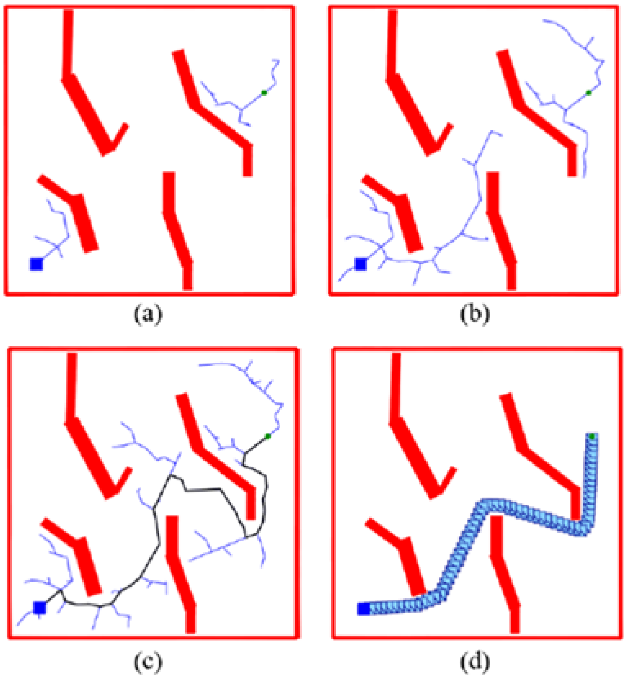


Figure 4-3: Growing two trees towards each other (Kuffner and LaValle, 2000)

The procedure of developing two trees consists of the following steps: (1) growing tree T_{init} (rooted in the initial state) toward a random sample node q_{rand}, resulting in a new node q_{new}; (2) growing tree T_{goal} (rooted in the goal state) toward the new node q_{new} in T_{init}; (3) growing T_{goal} toward another random sample node q'_{rand}, resulting in a new node q'_{new}; (4) growing tree T_{init} toward the new node q'_{new} in T_{goal}. This method is a kind of biasing the direction of the tree generation. It is expected that growing the two trees towards each other is a faster way of finding a solution. Furthermore, building two trees from the initial and goal configurations aims to precisely select the goal state instead of assuming that the goal state will be reached by generating nodes randomly, especially when a manipulator must reach a specific end-effector position and orientation.

In RRT-Connect, instead of attempting to extend the tree by an ε step, the *Connect* heuristic function iterates the *Extend* step until q_{rand} is reached or an obstacle is detected. Figure 4-4 shows the RRT-Connect algorithm. In each iteration, one tree is *Extended*, and an attempt is made to *Connect* the nearest node of the other tree to the new node, and then vice versa.

```
RRT-Connect($q_{init}$, $q_{goal}$)
  $T_a$.add($q_{init}$); $T_b$.add($q_{goal}$);
  for $i = 1$ to $m$ do
      $q_{rand}$ = Choose-Target();
      if not (Extend($T_a$, $q_{rand}$) == Trapped) then
          if (Connect($T_b$, $q_{new}$) == Reached) then
              return Path($T_a$, $T_b$);
      Swap($T_a$, $T_b$);
  return Failed;

Connect($T$, $q$)
  repeat
      Status = Extend($T$, $q$);
  until not (Status == Advanced)
  return Status;
```

Figure 4-4: RRT-Connect algorithm (Kuffner and LaValle, 2000)

Figure 4-5 shows the merging of two trees. The configuration q_{rand} is generated randomly; the configuration q_1 has been extended to q_{new}; q_2 is the closest configuration to q_{new} in T_{goal}. If it is possible to connect q_2 to q_{new}, T_{init} and T_{goal} are merged and a path is successfully found.

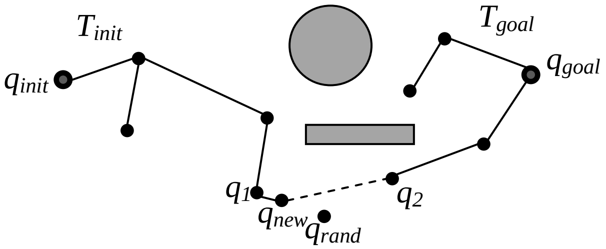


Figure 4-5: Merging two RRTs (Adapted from Choset et al., 2005)

This greedy heuristic results in reducing calculation time by a factor of three or four, especially in an uncluttered environment (Kuffner and LaValle, 2000). It works most effectively in a relatively open space for the majority of planning queries. In very cluttered environments, the *Connect* heuristic slightly increases the calculation time in comparison to using the *Extend* function to construct the two trees. Therefore, when the advantages and disadvantages of the *Connect* heuristic are weighed, using the *Connect* heuristic is worthwhile. Furthermore, this algorithm is theoretically proved to be probabilistically complete. The vertices tend to follow a uniform distribution over the free configuration space (Kuffner and LaValle, 2000).

4.3.2 RRT-CONNECT-CONNECT ALGORITHM

The greediest dual-tree RRT algorithm is RRT-Connect-Connect (RRT-Con-Con) (Kuffner and LaValle, 2000), which replaces the *Extend* function in line 4 of the RRT-Connect algorithm (Figure 4-4) with the *Connect* function and leads to a greedier heuristic. However, this greediness may result in an increased randomness of the whole path and makes the movement unrealistic. Additionally, an error tolerance value is set to directly connect the two trees when they are close enough to each other. This value should be selected carefully; if it is too large, collision may not be detected for the edge of the connection of the two trees; if it is too small, calculation time is wasted or a path that is unnecessarily long is generated.

4.4 PATH QUALITY IMPROVEMENT

As mentioned in the *Optimality* critieron in Subsection 3.3.2, the original RRT algorithm does not take the cost of the path into account during its search. It almost always converges to a suboptimal solution (Karaman and Frazzoli, 2010). To improve the quality of the path, it is often the case that a feasible path is found quickly. Additional computational time is devoted to improving the solution with heuristics depending on how much time is allowed in the application, which is called anytime algorithm. Examples can be found in Zilberstein and Russell (1995) and Berg et al. (2006). Berg et al. (2006) have considered adding the cost of the path in navigating a mobile robot. Time taken to execute the path (t_{path}) and the cost of the path (c_{path}), based on all relevant metrics other than time, are considered to contribute to the overall cost of the path. Examples of the metrics included in c_{path} are the proximity to adversaries or friendly agents and communication access. Equation (4-1) shows the cost function of the path.

$$C_{path} = w_t * t_{path} + w_c * c_{path} \tag{4-1}$$

where w_t and w_c are weights which add up to 1. It should be noted that the term *cost function* is equivelent to *objective function* used in optimization. However, although the quality of the path is improved using the cost function, the time needed for finding a solution increases significantly when the costs of different paths are calculated and compared (Ferguson, 2006). In recent research, Karaman and Frazzoli (2010) have claimed that the algorithm they proposed, Rapidly-exploring Random Graph (RRG), is asymptotically optimal in the sense that it converges to an optimal solution almost surely as the number of samples approaches infinity. However, the authors did not discuss the planning time needed to reach the optimal path. Thus, that research is impractical and not applicable for solving problems involving the motion planning of cranes in near real time. Consequently, in this research, the advantage of the greediest dual-tree algorithm RRT-Con-Con is taken and a cost function is used to improve the path quality.

4.5 DYNAMIC RRT

During re-planning, the equipment must either wait for the new path to be computed or move in an uncertain direction; therefore, rapid re-planning is essential. As described in the reusability criterion in Subsection 3.3.2, Ferguson et al. (2006) have proposed the DRRT algorithm, which depends on repairing the current RRT when new information is received. It removes just the invalid part of the path and grows the remaining tree until a new solution is found. As shown in Figure 4-6, the steps of DRRT are the following: (a) an initial RRT is generated from a start position to a goal position, (b) a new obstacle is detected in the *C-space*, (c) parts of the previous tree that are invalidated by the new obstacle are marked, (d) the tree is trimmed and invalid parts are removed, (e) the trimmed tree is grown until a new solution is generated. Whenever an obstacle is detected, the whole tree is checked to mark invalid nodes. The solution path is then checked to see if it contains any of the invalid nodes. If invalid nodes are found, the whole tree is trimmed and re-grown.

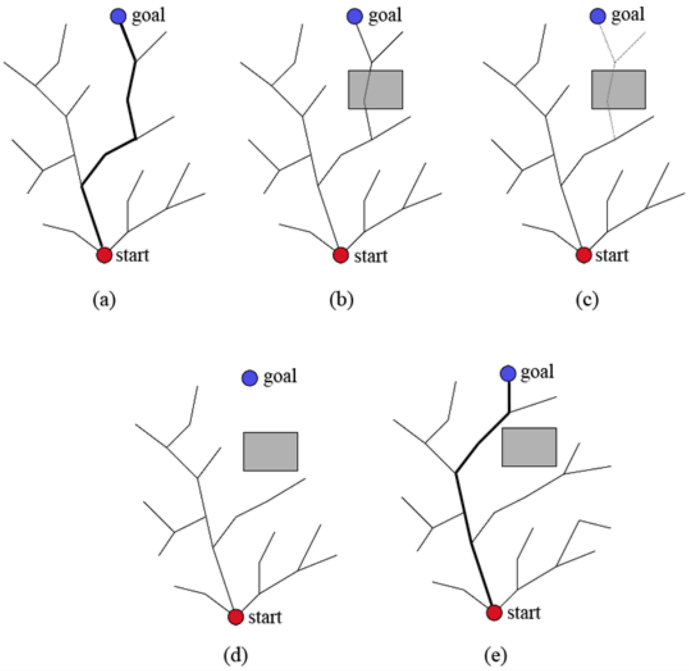


Figure 4-6: DRRT steps for re-planning (Adapted from Ferguson et al., 2006)

This DRRT is specific to one tree structure. The direction of growing the tree is reversed to avoid changing the root of the tree when doing re-planning. The new direction is from the goal configuration to the current configuration of the robot. Otherwise, the root of the tree changes constantly and the entire tree needs to be re-grown. For this reason, this DRRT cannot be directly used in the dual-tree algorithm. Furthermore, it is time consuming to trim the entire tree for all new obstacles, which may affect the far later nodes on the path. Some of the detected obstacles can become non-obstacles at a later time because they continuously move in the space. Therefore, in the proposed algorithm for re-planning, only the next several nodes on the path are checked for collision to ensure immediate safety.

4.6 PROPOSED MOTION PLANNING AND RE-PLANNING ALGORITHMS

The proposed motion planning algorithm is named RRT-Connect-Connect-Modified (RRT-Con-Con-Mod). This algorithm is a modified version of the RRT-Con-Con algorithm, which generates two trees from the initial and goal configurations. A new sampled node is selected to be connected to one of the trees based on its smoothness value as explained in the next subsection. An anytime algorithm is applied to find a better solution by using a cost function to evaluate the quality of the path. Furthermore, engineering constraints are taken into account to generate a feasible path. In the case of re-planning, a dynamic re-planning algorithm is proposed to efficiently repair the path when obstacles are detected.

4.6.1 PATH SMOOTHNESS

Due to the randomness nature of sampling-based algorithms and the greediness of the *Connect* function of RRT-Con-Con, the jaggedness of the path is inevitable and some movements of the crane may be unnecessary. These problems make the RRT-Con-Con algorithm impratical for crane operation. Therefore, in this research, the smoothness of the path is used as a metric to evaluate the path. Smoothness can be improved by keeping the difference between the coordinates of two subsequent nodes at a minimum or relatively small. Ali et al. (2005) have introduced a fitness function in a Genetic Algorithm for crane path planning. The fitness function calculates the total angular displacement of the robotic joints while the crane moves from a pick-up location to a place location. In the present research, the smoothness of a path is calculated by introduing a function $s(q_i, q_{i+1})$ representing the smoothness of the movement from node q_i to the next node q_{i+1} on the path, as shown in the following equation (4-2).

$$s\ (q_i,\ q_{i+1}) = \sqrt{\sum_{j=1}^{m}\left(d_{q_{i+1}}^{j} - d_{q_i}^{j}\right)^2} \qquad (4\text{-}2)$$

where m is the number of DoFs of the equipment; $d_{q_i}^{j}$ and $d_{q_{i+1}}^{j}$ are the normalized values of the movement along the jth DoF of nodes q_i and q_{i+1}, respectively. The purpose of normalization is to consider the effects of all movements of the boom represented by polar coordinates that have different units and different ranges. For example, in the case study discussed in Section 4.7, the swing angle of the boom of the hydraulic crane is in the range of [-180, 180] degrees and the boom entension is in the range of [10.97, 33.53] meters. However, after normalization, both movements are in the unitless range of [0,1]. The smoothness of the path s_{path} is the sum of the smoothness of movements between all couples of consecutive nodes on the path, as shown in Equation (4-3).

$$s_{path} = \sum_{i=1}^{k-1} s(q_i, q_{i+1}) \qquad (4\text{-}3)$$

where k is the number of nodes on the path. s_{path} should be minimized to improve the smoothness of the crane movement.

4.6.2 COST FUNCTION

Considering plan execution time and path smoothness, the overall cost of a path can be calculated using a cost function, as shown in Equation (4-4), which is used to evaluate each path and to select a better path. A path with less executing time and less smoothness value is considered to be a better path.

$$C_{path} = \alpha\ \cdot w_t\ \cdot t_{path} + \beta\ \cdot w_s\ \cdot s_{path} \qquad (4\text{-}4)$$

t_{path} is calculated based on the path and velocity along each DoF. For example, the velocities of extending the boom and swinging the boom are different; therefore, the time of executing a path should be calculated according to the equipment specifications.

The weights of execution time w_t and path smoothness w_s can be based on a preference of shorter time or better smoothness. For example, if better smoothness is preferrable, $w_t = 0.4$ and $w_s = 0.6$ can be selected. Other weights (α and β) are used to adjust the values of t_{path} and s_{path} to the same order of magnitude. For example, if the range of the values of t_{path} is within 100 to 1000, and the range of the values of s_{path} is within 1 to 10, $\alpha = 0.01$and $\beta =1$ can be applied to t_{path} and s_{path}, respectively.

4.6.3 SELECTIVE NODE SAMPLING

After randomly sampling nodes over the whole *C-space*, only nodes that are expected to result in a smoother path are selected as nodes that will be connected to the tree. The smoothness of a newly sampled node q is evaluated by calculating the summation of the smoothness values from this node to the roots (q_{init} and q_{goal}) of the two trees T_{init} and T_{goal}.

$$S\,(q) = s\,(q, q_{init}) + s\,(q, q_{goal}) \qquad (4\text{-}5)$$

A threshold is set by multiplying the smoothness value between the initial and the goal nodes ($s(q_{init}, q_{goal})$) by an amplifier (γ). The first sampled node that meets the following inequality is selected as the node to be connected to the tree.

$$S\,(q) \leq \gamma\ \ s\,(q_{init}, q_{goal}) \qquad (4\text{-}6)$$

However, to restrain the calculation time, if the algorithm failed to find a node satisfying this condition after k trials, the node with the minimum smoothness value is selected.

Figure 4-7 shows a 2D schematic example of nodes with different smoothness values. Each γ value defines an area where the nodes inside a smaller area are expected to contribute to a better solution. According to this criterion, the smaller the γ value, the better the smoothness of the node.

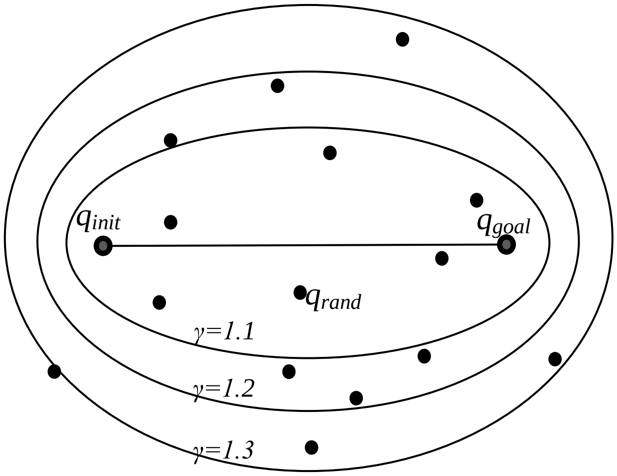
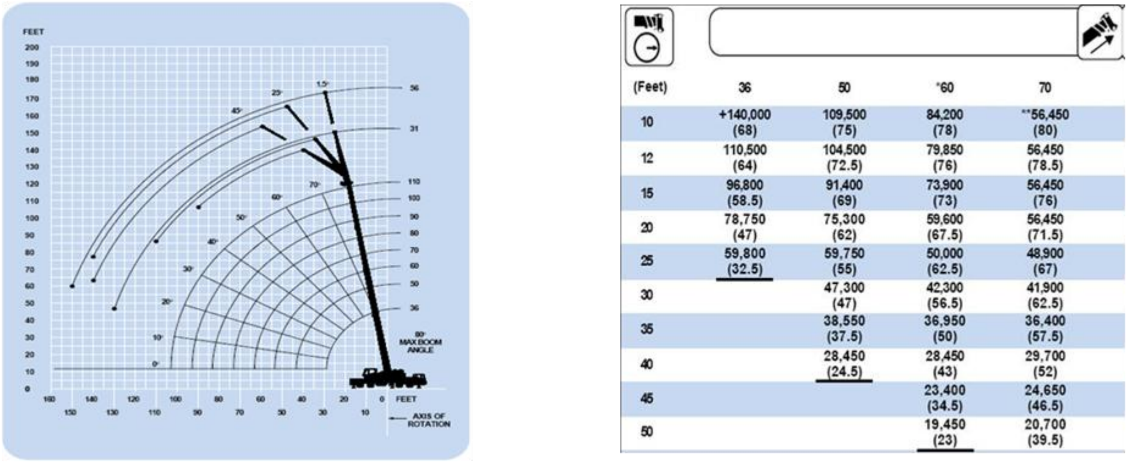


Figure 4-7: Nodes with different smoothness values

4.6.4 ENGINEERING CONSTRAINTS

For cranes, the engineering constraints are mainly created by the working range and by the load charts in order to avoid tip-over problems. Figure 4-8 shows examples of a working range graph and of a load chart. The working range shows the minimum and maximum boom angles according to the length of the boom and to the size of the counterweight. Load charts give the lifting capacity based on the boom length, the boom angel to the ground and the size of the counterweight. For example, for a Groove crane TSM870 (Groove Crane, 2008), if the lift object is 15,000 lbs (6.8 metric tons) and the counterweight is 18,000 lbs (8.2 metric tons), the ranges of the three DoFs for this lifting task are the following: (1) boom length: 36 to 110 ft (10.97 to 33.53 m); (2) luffing angle: 23 to 80 degrees; (3) swing angle: -180 to 180 degrees, as shown in Table 4-1.



(a) Working range

(b) Load chart

(Feet)	36	50	"60	70
10	+140,000 (68)	109,500 (75)	84,200 (78)	**56,450 (80)
12	110,500 (64)	104,500 (72.5)	79,850 (76)	56,450 (78.5)
15	96,800 (58.5)	91,400 (69)	73,900 (73)	56,450 (76)
20	78,750 (47)	75,300 (62)	59,600 (67.5)	56,450 (71.5)
25	59,800 (32.5)	59,750 (55)	50,000 (62.5)	48,900 (67)
30		47,300 (47)	42,300 (56.5)	41,900 (62.5)
35		38,550 (37.5)	36,950 (50)	36,400 (57.5)
40		28,450 (24.5)	28,450 (43)	29,700 (52)
45			23,400 (34.5)	24,650 (46.5)
50			19,450 (23)	20,700 (39.5)

Figure 4-8: Examples of working range and load chart of a crane (Groove Crane, 2006)

Table 4-1: Ranges of crane boom length and luffing angles

Boom length (ft)	**Luffing angle limits (degree)**
36	32.5-68
50	24.5-75
60	23-78
70	32-80
80	37.5-78.5
90	46.5-80
100	53-79
110	54.5-80

The original load chart is given in Appendix B. Furthermore, the range of luffing angles varies according to the boom length; therefore, node sampling should be constrained within these ranges. When connecting the sampled node to the tree, the intermediate nodes are also checked to make sure that they meet these engineering constraints. The *Connect* function discussed in Section 4.3 is modified to check these constraints for the intermediate nodes on the connection edge between the sampled node and the tree.

4.6.5 PROPOSED MOTION PLANNING ALGORITHM

The proposed RRT-Con-Con-Mod is a modified version of RRT-Con-Con, which uses selective node sampling and considers the engineering constraitnts introduced above. Figures 4-9 and 4-10 show the flowchart and the pseudo code of the RRT-Con-Con-Mod algorithm, respectively. First of all, the load charts and the working range data are read after the parameters of the lift task are defined including the lift weight, the counterweight of the crane, and the initial and goal configurations of the crane. Based on the required capacity, the working range of the crane is defined. A node q_{rand} is sampled randomly within the range of the engineering constraints. Then the smoothness value of q_{rand} is calculated based on Equation (4-5). If this value is less than the threshold value γ *s* (q_{init}, q_{goal}), q_{rand} is selected as the node that one tree should grow towards. If the value is bigger than the threshold, this value is compared with the smallest smoothness value of other sampled nodes. Then, after several trials, the node with the smallest smoothness value is returned and connected to the tree. Next, the nearest node on one tree T_a is found to connect to q_{rand} and to extend the tree using a modified *Connect* function shown in Figure 4-4. This *Connect-Mod* function checks if every intermediate node is within the working range. After T_a is extended to q_{new}, the node on T_b nearest to q_{new} is found and the gap between the two trees is checked to see if it is less than the predefined error tolerance. In that case, the two trees are connected directly and the path between the initial and the goal states is returned. Otherwise, T_b tries to extend towards q_{new}. If the two trees are connected successfully, a path is returned, otherwise, T_a and T_b are swapped and another round of sampling and connecting starts untill the maximum number of path generation trials *m* is reached.

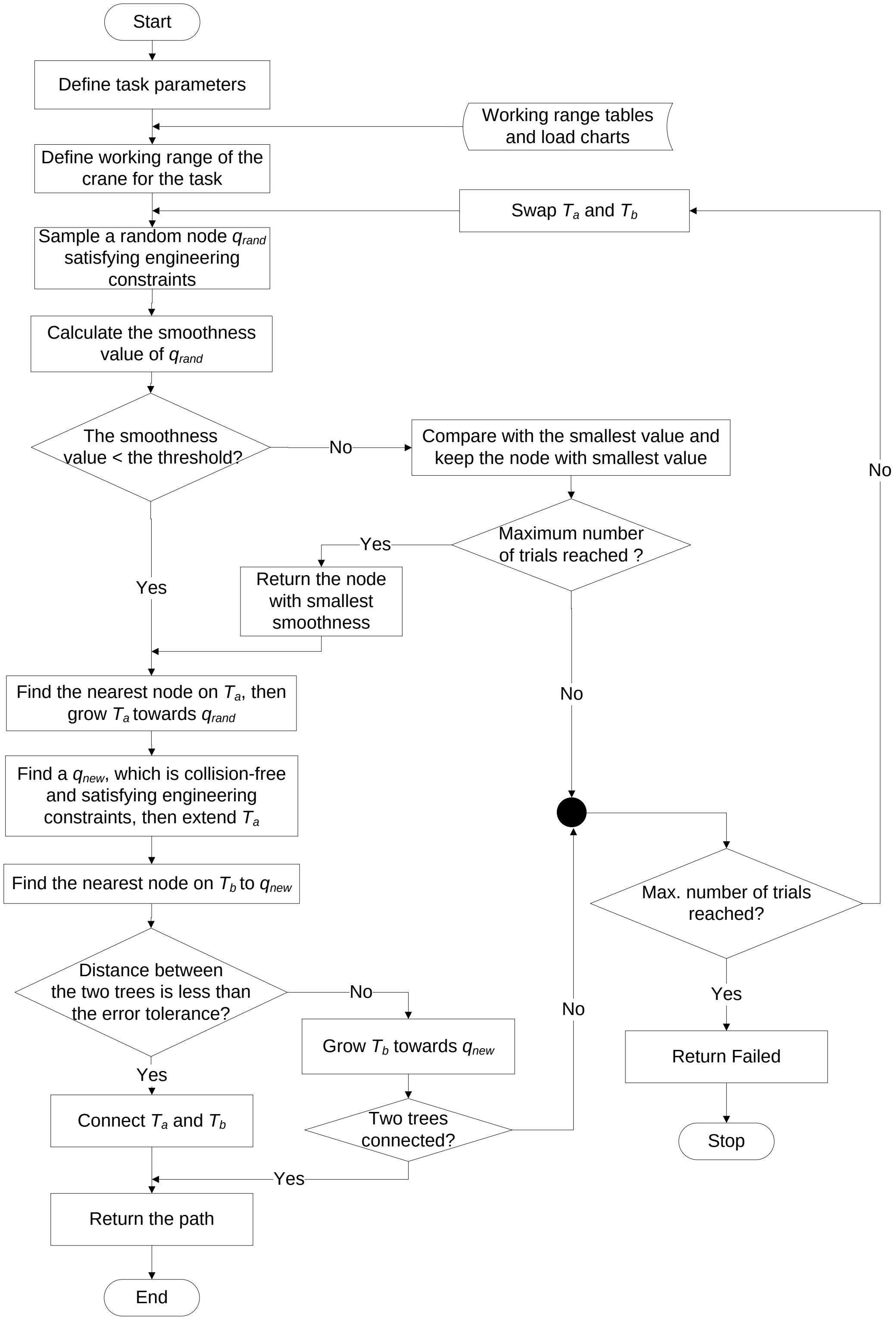

Start
Define task parameters
Working range tables and load charts
Define working range of the crane for the task
Swap T_a and T_b
Sample a random node q_{rand} satisfying engineering constraints
Calculate the smoothness value of q_{rand}
The smoothness value < the threshold?
No
Compare with the smallest value and keep the node with smallest value
No
Maximum number of trials reached ?
Yes
Return the node with smallest smoothness
Yes
Find the nearest node on T_a, then grow T_a towards q_{rand}
No
Find a q_{new}, which is collision-free and satisfying engineering constraints, then extend T_a
Find the nearest node on T_b to q_{new}
Max. number of trials reached?
Distance between the two trees is less than the error tolerance?
No
No
Yes
Grow T_b towards q_{new}
Return Failed
Yes
Connect T_a and T_b
Two trees connected?
Stop
Yes
Return the path
End

Figure 4-9: Flowchart of motion planning algorithm RRT-Con-Con-Mod

```
RRT-Con-Con-Mod($q_{init}$, $q_{goal}$)
  $T_a$.add($q_{init}$); $T_b$.add($q_{goal}$);
  for $i$ = 1 to $m$ do
      $q_{rand}$ = Choose-Better-Node();
      if not (Connect-Mod($T_a$, $q_{rand}$) == Trapped)
          if (Connect-Mod($T_b$, $q_{new}$) == Reached)
              return Path($T_a$, $T_b$);
      Swap($T_a$, $T_b$);
  return Failed;

Choose-Better-Node()
  $S_o$ = s($q_{init}$, $q_{goal}$);
  $S_{less} = \infty$;
  $q_{better}$ = null;
  for $i$ = 1 to $k$
      $q$ = Random-Node-Satisfy-Eng-Const();
      $S$ = s($q$, $q_{init}$ ) + s($q$, $q_{goal}$);
      if $S \leq \gamma \cdot S_o$
          return $q$;
      else
          if $S < S_{less}$
              $q_{better} = q$;
  return $q_{better}$;

Connect-Mod($T$, $q$)
  repeat
      Status = Extend($T$, $q$);
  until not (Status == Advanced and Eng-Const-Satisfied($q$) == true)
  return Status;

Extend($T$, $q$)
  $q_{near}$ = Nearest-Neighbor($q$, $T$);
  $q_{new}$ = New-Node($T$, $q_{near}$, $q$);
  if Collision-Free($q_{new}$) == true
      $T$.add($q_{new}$);
      if $q_{new} \approx q$
          return Reached;
      else
          return Advanced;
  return Trapped;
```

Figure 4-10: Pseudo code of motion planning algorithm RRT-Con-Con-Mod

Due to the randomness of the RRT-Con-Con-Mod algorithm, it is difficult to guarantee that better solutions can be found using a specific γ value with one trial using one specific random seed number. A large number of tests have been carried out by changing the random seed number and the γ value. It was found that the paths generated by using the same random seed number are the same when γ value is large, e.g., $\gamma = 2.0$ in the first case study introduced in Section 4.7. These paths are the same as the paths generated by the RRT-Con-Con algorithm. It was also found that when the γ value is gradually reduced the smoothness of the path in the majority of the cases decreases to some extent and then starts to increase again at a certain value of γ. This γ value varies depending on the value of the random seed number. This behavior occurs because small γ values can result in over-constraining the dual-tree generation mechanism. Consequently, in order to obtain the best feasible path, an anytime algorithm is applied to investigate the improvement of the proposed RRT-Con-Con-Mod algorithm by changing the random seed number and the γ value. Figure 4-11 shows the flowchart of this algorithm. Instead of reading the load charts and the working range data every time when calling RRT-Con-Con-Mod, they are read only once after defining the task parameters. Based on this information, the working range of the task is defined. A path P_{best} is initialized by calling RRT-Con-Con to find a feasible path. By reducing the γ value, nodes with less smoothness value are selected and connected to the tree; in this way, a better path is expected. After several loops of reducing γ, no more improvement can be realized with the same random seed number. If there is enough time to search for a better path, the random seed number is changed to find other feasible paths, and the loop of improving the smoothness of the path is repeated until time is over.

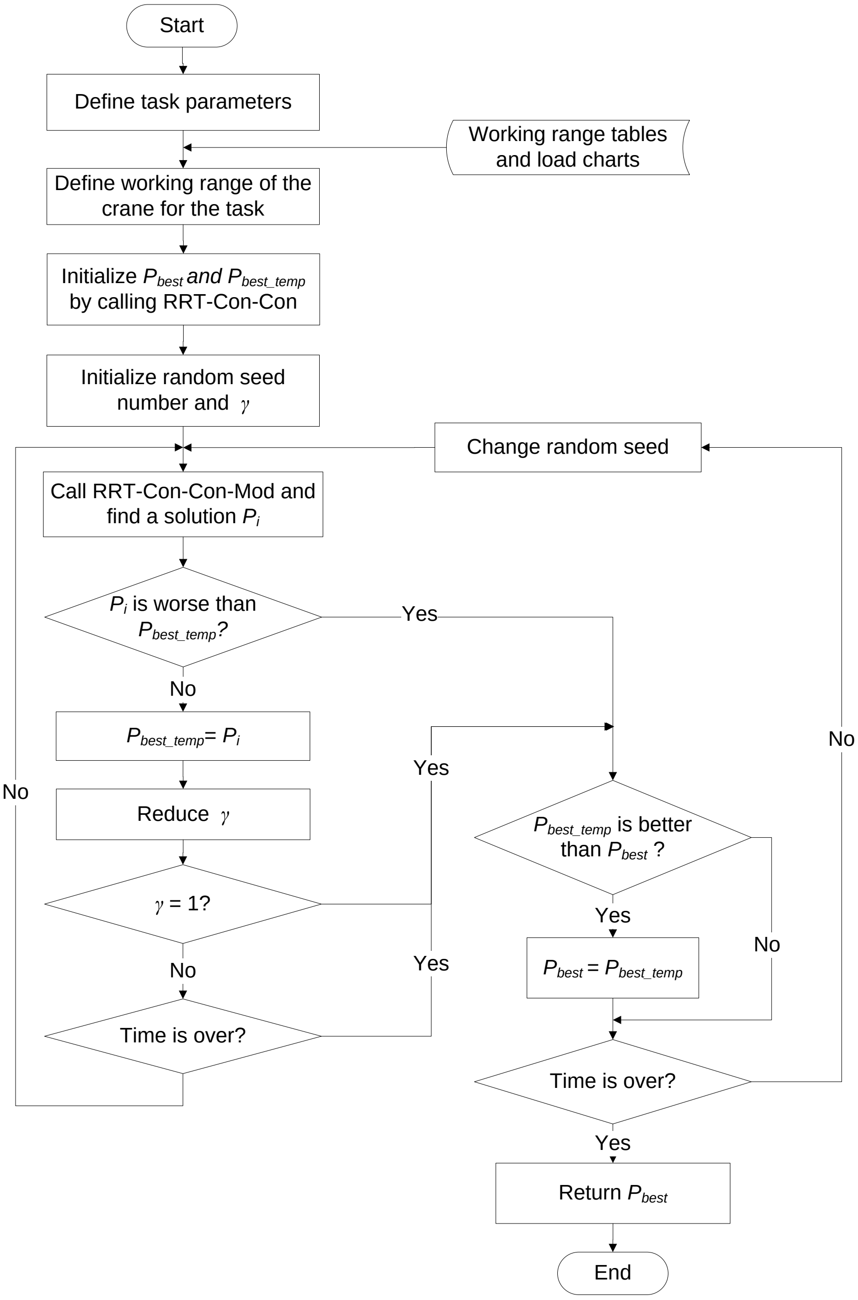

Start
Define task parameters
Working range tables and load charts
Define working range of the crane for the task
Initialize P_{best} and P_{best_temp} by calling RRT-Con-Con
Initialize random seed number and γ
Change random seed
Call RRT-Con-Con-Mod and find a solution P_i
P_i is worse than P_{best_temp}?
Yes
No
$P_{best\_temp} = P_i$
Reduce γ
γ = 1?
Yes
No
Time is over?
Yes
No
P_{best_temp} is better than P_{best} ?
Yes
No
$P_{best} = P_{best\_temp}$
Time is over?
No
Yes
Return P_{best}
End

Figure 4-11: Flowchart of anytime motion planning algorithm

4.6.6 PROPOSED MOTION RE-PLANNING ALGORITHM

During the execution of the motion plan of a crane, static and dynamic obstacles not considered in the planning, such as other equipment in the vicinity of the crane, necessitate the re-planning of its path. In the framework discussed in Subsection 3.5, the

Coordinator Agent decides the priorities of movements based on different criteria. As discussed in Section 4.5, an efficient re-planning algorithm should be able to plan optimal traverses in near real time by incrementally repairing paths of the equipment as new information is discovered. Reusing a part of the previously generated plan is a good option to reduce the time of re-planning. Since RRT-Con-Con-Mod has been proposed in the present research for rapid motion planning for cranes, a dynamic version of this algorithm is proposed to solve the re-planning problem.

Figures 4-12 and 4-13 show the dynamic re-planning algorithm called DRRT-Con-Con-Mod. RRT-Con-Con-Mod is used twice: first to generate the initial plan (Line 1 in Figure 4-12) and, then, to repair the path if necessary (Line 15 in Figure 4-12). The path is repaired by re-growing two trees rooted at the current node q_{crane} (current configuration of the crane) and the first collision-free node $q_{newgoal}$ on the remaining path (Line 14 in Figure 4-12) while detecting collisions for the remaining path based on updated environment information.

The main function starts by growing two trees from the initial q_{init} configuration and the goal q_{goal} configuration taking into account the engineering constraints and the smoothness value. If the two trees connect successfully, a path P is obtained and is executed using a loop until the equipment reaches the goal (Lines 4 to 25). Before the next movement, environment information is updated (Line 8) to check whether the next movement is collision-free or not. If there is no obstacle, the crane is moved to the next configuration (Lines 10 and 11); otherwise, the remaining path $P_{remaining}$ is evaluated to remove the affected nodes on it (Line 13). Once a collision-free node is found, this node is recorded as a temporary goal node $q_{newgoal}$, and together with the crane's current configuration q_{crane} as an initial node, two new trees are developed to connect these two configurations. If the connection succeeds, the new partial path P_{new} is merged with the remaining path to form a new path (Line 24). If the connection fails, the following nodes on the remaining path are evaluated and attempts are made until successful connection is made between that node and the current configuration node.

```
DRRT-Con-Con-Mod($q_{init}$, $q_{goal}$)
1. $P$ = RRT-Con-Con-Mod ($q_{init}$, $q_{goal}$);
2. if $P$ = Failed
3.     return Failed; //There is no feasible path
4. while ($q_{crane} \neq q_{goal}$)    //Start to execute the path
5.     $q_{next}$ = Read-Next-Node($P$);
6.     while ($q_{crane} \neq q_{next}$)
7.         $q_{inter}$ = Interpolate($q_{crane}$, $q_{next}$); //Calculate intermediate node
8.         Update-Environment();
9.         if Collision-Free($q_{inter}$) == True
10.            Move-Crane($q_{inter}$);
11.            $q_{crane}$= $q_{inter}$;
12.        else
13.            $P_{remaining}$ = Validate-Nodes($P_{remaining}$);
14.            $q_{newgoal}$ = First-Node($P_{remaining}$);
15.            $P_{new}$ = RRT-Con-Con-Mod($q_{crane}$, $q_{newgoal}$);
16.            while $P_{new}$ == Failed
17.                if $q_{newgoal}$ = $q_{goal}$
18.                    return Failed;
19.                else
20.                    $P_{remaining}$.Remove($q_{newgoal}$);
21.                    $P_{remaining}$ = Validate-Nodes ($P_{remaining}$);
22.                    $q_{newgoal}$ = First-Node($P_{remaining}$);
23.                    $P_{new}$ = RRT-Con-Con-Mod ($q_{crane}$, $q_{newgoal}$);
24.            $P$ = Merge-Path($P_{new}$, $P_{remaining}$);
25.            $q_{next}$ = Read-Next-Node($P$);

Validate-Nodes (path)
26.  repeat for all the nodes $q$ on path
27.      if Collision-Free($q$) == False
28.          path.Remove($q$);
29.      else
30.          return path;
31.  return Stop;
```

Figure 4-12: Dynamic re-planning algorithm DRRT-Con-Con-Mod

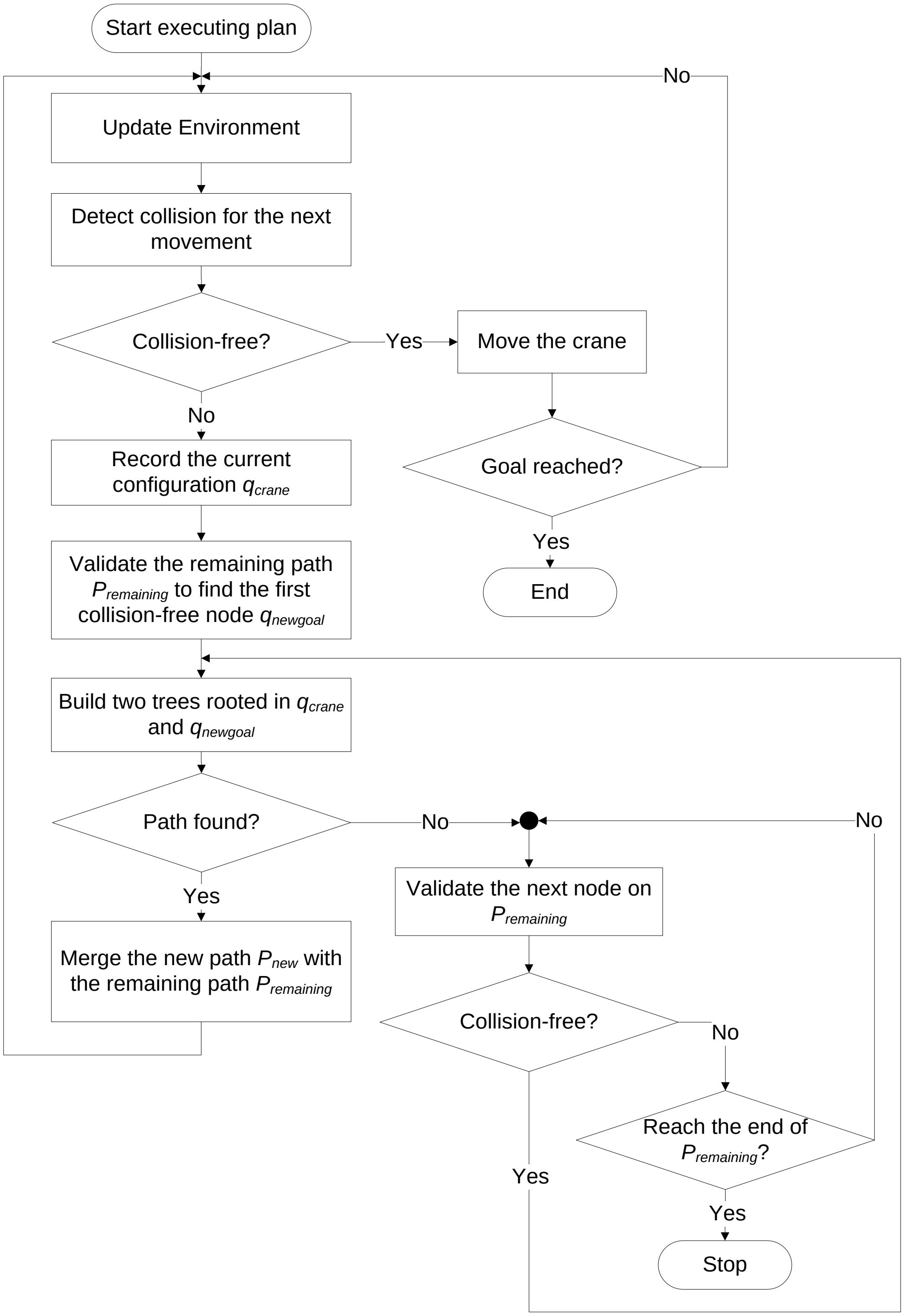

Start executing plan
No
Update Environment
Detect collision for the next movement
Collision-free?
Yes
Move the crane
No
Record the current configuration q_{crane}
Goal reached?
Validate the remaining path $P_{remaining}$ to find the first collision-free node $q_{newgoal}$
Yes
End
Build two trees rooted in q_{crane} and $q_{newgoal}$
Path found?
No
No
Yes
Validate the next node on $P_{remaining}$
Merge the new path P_{new} with the remaining path $P_{remaining}$
Collision-free?
No
Reach the end of $P_{remaining}$?
Yes
Yes
Stop

Figure 4-13: Flowchart of re-planning algorithm DRRT-Con-Con-Mod

The differences between the proposed dynamic motion planning algorithm and the DRRT algorithm (Ferguson et al., 2006) discussed in Subsection 3.3.4 are the following: (1) a dual-tree algorithm is used in the present research to find a solution. This algorithm is faster than the one-tree DRRT algorithm; (2) the proposed algorithm ensures immediate safety and ignores obstacles that are far away from the current node on the path since these obstacles may move before the crane reaches that part of the path. Thus, the proposed algorithm reduces the calculation time compared with the DRRT algorithm since it does not trim the entire tree. Due to the greediness of RRT-Con-Con-Mod, regenerating a small part of the path is expected to be very quick. To the best of our knowledge, no research has considered dual-tree RRT algorithms for dynamic motion planning.

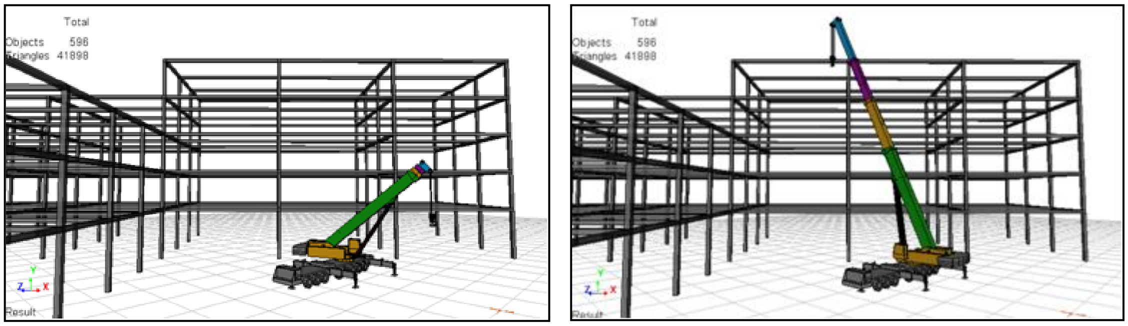
4.7 CASE STUDIES

A simulation environment is built in Autodesk Softimage (2010), where a scene is created with two identical hydraulic cranes (TMS870/TTS870) (Groove Crane, 2008). CAD models of the cranes are imported into Softimage. A hierarchy of components and kinematics is created in a 3D environment. Four DoFs are considered in the current work, as shown in Figure 3-3(a). In addition, a model of a steel structure with 596 elements is created in the simulation environment. Static objects are defined by grouping them under a specific model with the name *obstacles*. The algorithm then considers all objects in this model as static obstacles and performs collision checking during the planning stage. Dynamic objects are defined by applying motion information to 3D objects (AlBahnassi, 2009). Details about the prototype system are given in Chapter 6.

The ranges of the dimensions of the *C-space* vary according to the size of the counterweight and the size of the lift weight. The example shown in Subsection 4.6.4 is used in the present case study. The length of the cable could vary between 0 and the distance from the tip of the boom to the ground. In the present research, a critical volume under the boom is considered for collision detection, taking into account the cable and the hook. The initial and goal configurations of the crane are shown in Figure 4-14. It should be noticed that the fourth DoF does not contribute to the smoothness of the path; therefore, it is not used in the calculation of the cost.

4.7.1 CASE STUDY – 1: MOTION PLANNING

To evaluate the proposed motion planning algorithm, two major criteria are considered: calculation time and path quality. Calculation time should be as short as possible in near real-time applications. Path quality is evaluated in terms of smoothness and plan execution time, as discussed in Subsection 4.6.2.



(a) Initial configuration **(b) Goal configuration**

Figure 4-14: Initial and goal configurations of the crane

Fifty-one random seed numbers (0-50) were used to compare the planning time for each algorithm. As shown in Table 4-2, times spent for finding a feasible path using RRT, RRT-Con-Con and RRT-Con-Con-Mod are listed. It was found that RRT took more time to find a path, whereas the dual-tree algorithms found a path much faster due to the greediness of the dual-tree structure and the *Connect* function. The RRT-Con-Con-Mod spent slightly more time than RRT-Con-Con to check the threshold of the smoothness of the nodes.

Table 4-2: Comparison of calculation time of three algorithms

Algorithm	**Average calculation time (s)**	**Shortest time (s)**	**Longest time (s)**	**Std. dev. (s)**
RRT	16.95	3.56	39.53	8.302
RRT-Con-Con	1.59	0.61	3.16	0.594
RRT-Con-Con-Mod	1.91	0.61	4.11	0.718

The same random seed numbers were used sequentially in the anytime algorithm to evaluate the improvement that can be achieved using RRT-Con-Con-Mod. The γ value was initialized with the value of 1.7 and was reduced by a step of 0.1. No time limit was set in order to evaluate the performance of the anytime algorithm. The program stopped after finishing the calculations using all of the 51 random seed numbers. The results were sorted according to smoothness values. It was found that in the top 14 cases, where the paths are smoother, there was no improvement when applying the threshold. By contrast, in the remaining 37 cases, an average improvement of 11.51% better smootheness occurred in 24 cases compared with the paths found using RRT-Con-Con. The best solution among these cases was 29.89% better than the path generated by RRT-Con-Con. Table 4-3 shows the results of the improved cases and the average improvement percentage with the corresponding γ value. Figure 4-15 shows the smoothness improvement during the time taken for running 51 cases with 320 run times (447.5 seconds). An original path is found with a smoothness value of 3.025. After 9 run times

(16.31 seconds) by reducing γ value and changing the random seeds, a better path is found with a smoothness value of 2.563. Then at run time 89 (146.75 seconds), a better path is found with a smoothness value of 2.372. Table 4-4 shows the smoothness values and the execution times of these three paths (P_1, P_2, P_3) and the improvement pecentages of P_3 compared to P_1. The execution time is calculated by assuming the speeds of the movements of the boom as 3.75 °/second for swinging, 1.04 °/second for luffing, and 0.124 meter/second for extending the boom (Manitowoc, 2010). It can be seen that the smoothness and the execution time are improved by 21.59% and 16.86%, respectively. Using $\alpha = \beta = 1$ and $w_t = w_s = 0.5$ in Equation (4-4), the costs of the three paths are calculated and shown in the same table. The improvement of the path cost is 18.32%. As explained in Subsection 4.6.2, other weights can be used in Equation (4-4).

Table 4-3: Path smoothness using different threshold values

Threshold Multiplier (γ) used in RRT-Con-Con-Mod	1.6	1.5	1.4	1.3	1.2	1.1
Cases improved	2	3	5	4	7	3
Averaged improvement (%)	9.603	6.729	10.565	6.769	11.927	16.183

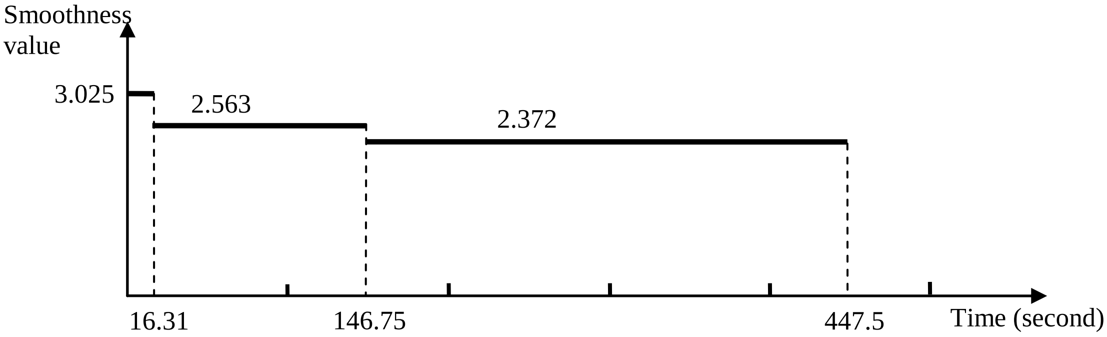


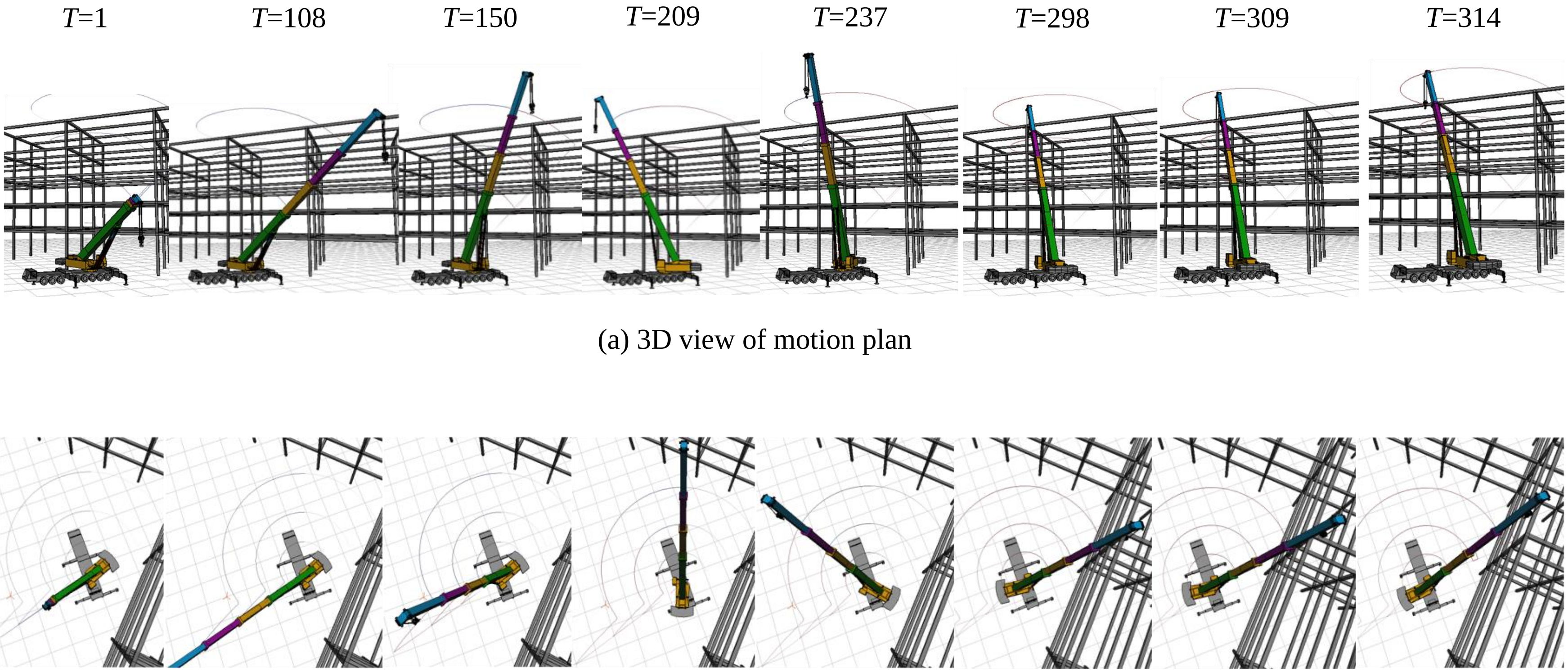
Figure 4-15: Smoothness improvement

Table 4-4: Smoothness values, execution times, and costs of three paths

Path Path cost values	P_1	P_2	P_3	Improvement of P_3 compared to P_1 (%)
Smoothness value	3.025	2.563	2.372	21.59
Execution time (min)	6.82	6.54	5.67	16.86
Cost of the path	4.923	4.55	4.021	18.32

Figure 4-16 shows eight snapshots of path P_1 at sereval frames of the simulation. This path includes several backward movements due to the randomness of the algorithm. *T* represents the time frame used in the simulation where 32 frames equal to 1 second. Between frame 1 and 108, the crane extends its boom, and then raises its boom until frame 150. After that, the crane starts rotating the boom clockwise until frame 209. However, a counterclockwise rotation occurs between frames 209 and 237. Next, the crane continues rotation clockwise until frame 298 and raises the boom until frame 309. Then, again, the crane rotates the boom counterclockwise to reach the goal. Because of the counterclockwise rotation, the smoothness value is large and redundant movements occur.

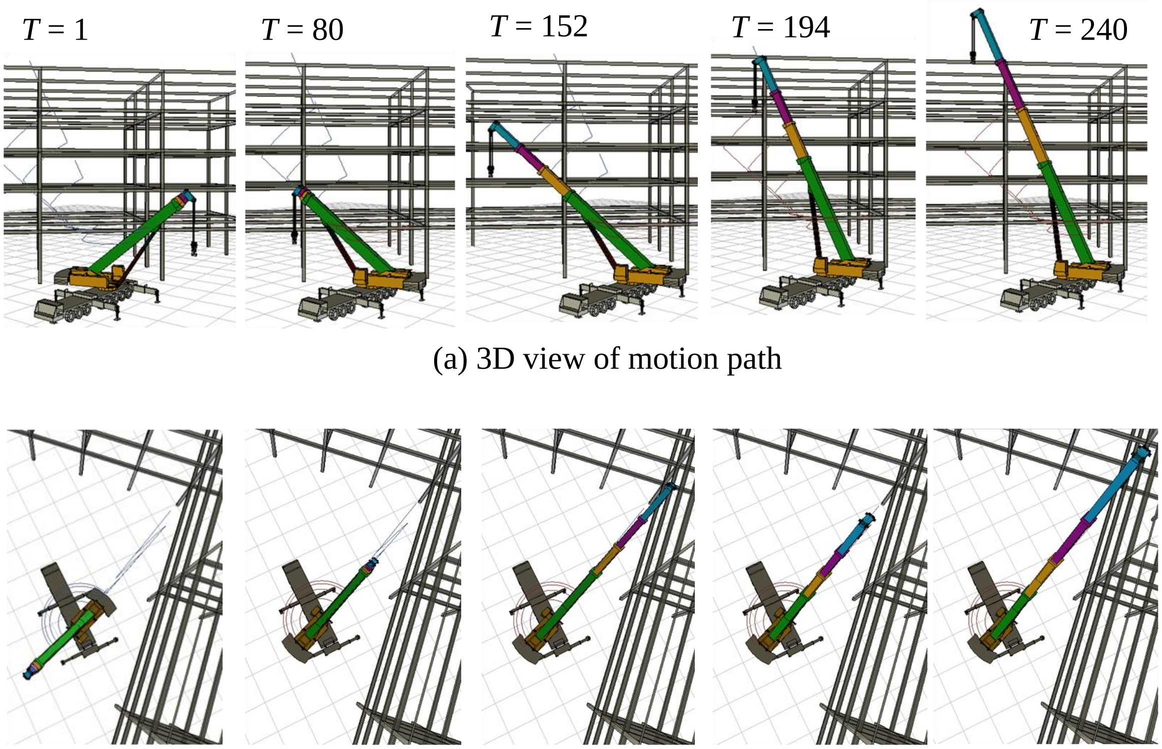
After applying the anytime algorithm, the improved path (P_3) is shown in Figure 4-17. The smoothness value of this path is 2.372, which is the best path found by using RRT-Con-Con-Mod. The motion path starts by rotating the boom clockwise; and, then, at frame 80, the boom is extended. After that, at frame 152, the boom is raised until frame 194, and finally the boom is extended to reach the goal.



(a) 3D view of motion plan

(b) Top view of motion plan

Figure 4-16: The initial path (P_1)



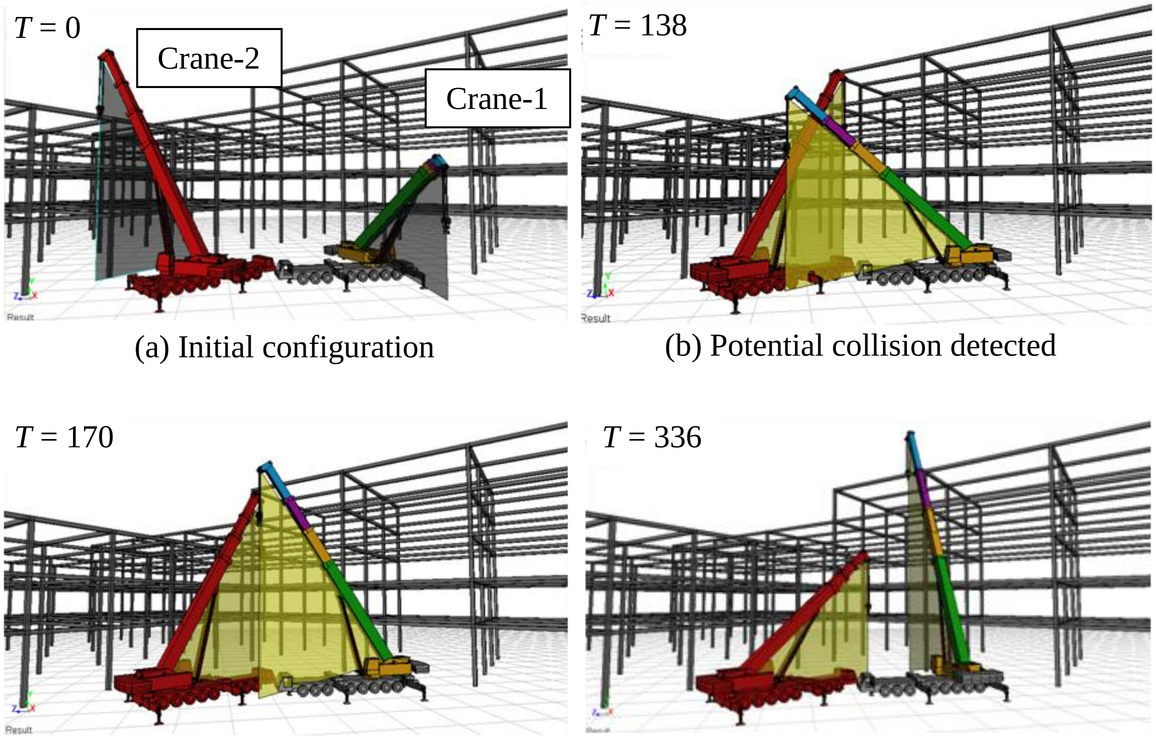
(a) 3D view of motion path

(b) Top view of motion path

Figure 4-17: The improved path (P_3)

4.7.2 CASE STUDY – 2: MOTION RE-PLANNING

The present case study focuses on testing the motion re-planning algorithm introduced in Subsection 4.6.6. In this case, as shown in Figure 4-18, a second crane (Crane-2) is located in the same area of Crane-1 used in Case Study-1. It is assumed that Crane-2 has a higher priority than Crane-1. While swinging its boom 90 ° clockwise, Crane-2 may become an obstacle for Crane-1. When a potential collision is detected as shown in Figure 4-18(b), re-planning is triggered to re-plan the path for Crane-1 by raising its boom to avoid collision as shown in Figures 4-18(c) and (d).



(a) Initial configuration

(b) Potential collision detected

(c) Collision avoided

(d) Goal configuration

Figure 4-18: Two cranes working in the same area

A comparison is carried out between the proposed dynamic algorithm DRRT-Con-Con-Mod and the basic DRRT algorithm based on the concept of Furguson (2006). The same random seed numbers are used for both algorithms to generate initial motion plans for Crane-1. Due to the randomness of the algorithms, the potential collision of the two cranes does not always occur. However, in some cases, more than one re-planning occured during the task when Crane-2 was detected several times as an obstacle. In these cases, the calculation time of each re-planning was taken into account in calculating the average re-planning time. The average re-planning time of the DRRT-Con-Con-Mod algorithm was 1.47 seconds compared with 5.98 seconds for DRRT, resulting in a reduction in re-planning time by a factor of four. Furthermore, DRRT cannot guarantee that a feasible path will be found in each re-planning case. In the 22 cases in which re-planning occured, there were 10 cases in which the DRRT failed to find a new path. Whereas in all the re-planning cases (23 cases) of DRRT-Con-Con-Mod, the algorithm sucessfully found a new path. Table 4-5 shows the comparison of re-planning times produced by DRRT and DRRT-Con-Con-Mod.

Table 4-5: Comparison of re-planning times

Algorithm	Average calculation time (s)	Shortest time (s)	Longest time (s)	Std. dev. (s)
DRRT	5.98	0.17	26.97	9.63
DRRT-Con-Con-Mod	1.47	0.11	5.27	1.40

4.8 SUMMARY AND CONCLUSIONS

In this chapter, RRT algorithms have been investigated in depth and several variations have been reviewed, such as dual-tree algorithms and algorithms for path quality improvement. Based on these reviews, we have developed a new algorithm called RRT-Connect-Connect-Modified (RRT-Con-Con-Mod) for crane motion planning. The main charastistics of this algorithm are the following: (1) It is a dual-tree RRT algorithm, which geneartes two trees from the initial and goal configurations; (2) A cost function is used to evaluate the quality of the path by taking into account the smoothness of the path and the time taken to execute the path; (3) Engineering constraints are considered to generate safe paths for the cranes and to avoid tip over due to overloading.

In the case of re-planning, a dynamic re-planning algorithm has been proposed to efficiently repair the path when the environment is updated. Compared with the DRRT algorithm proposed by Furguson (2006), the proposed dynamic algorithm maintains a focus on the path instead of the entire tree by regenerating a partial path to replace the part that is not collision-free due to new obstacles. The advantage of the proposed algorithm is that the time for trimming the entire tree is eliminated and an immediate collision-free movement is ensured for the crane, thereby reducing safety risks. Regenerating the trees and finding a feasible partial plan are carried out quickly thanks to the greediness of the dual-tree structure and the *Connect-Mod* function.

Comparisons have been made between RRT, RRT-Con-Con and RRT-Con-Con-Mod for motion planning, and between DRRT and DRRT-Con-Con-Mod for re-planning. The results show that the path smoothness is improved by applying the anytime algorithm while gradually narrowing the area for node sampling based on the smoothness value. An average improvement of 11.51% better smoothness has been obtained compared with the paths found by using RRT-Con-Con. The calculation time of RRT-Con-Con-Mod was much less than that of the RRT algorithm. The best path found using the anytime algorithm shows an improvement in smoothenss and execution time of 21.59% and 16.86%, respectively. The cost of the path is consequently reduced by 18.32%. As for re-planning, a reduction in re-planning time by a factor of four is achieved by using the proposed dynamic algorithm. Furthermore, the DRRT-Con-Con-Mod algorithm was always able to find a new path during re-planning.

CHAPTER 5 TRACKING CRANE POSES USING UWB SYSTEM

5.1 INTRODUCTION

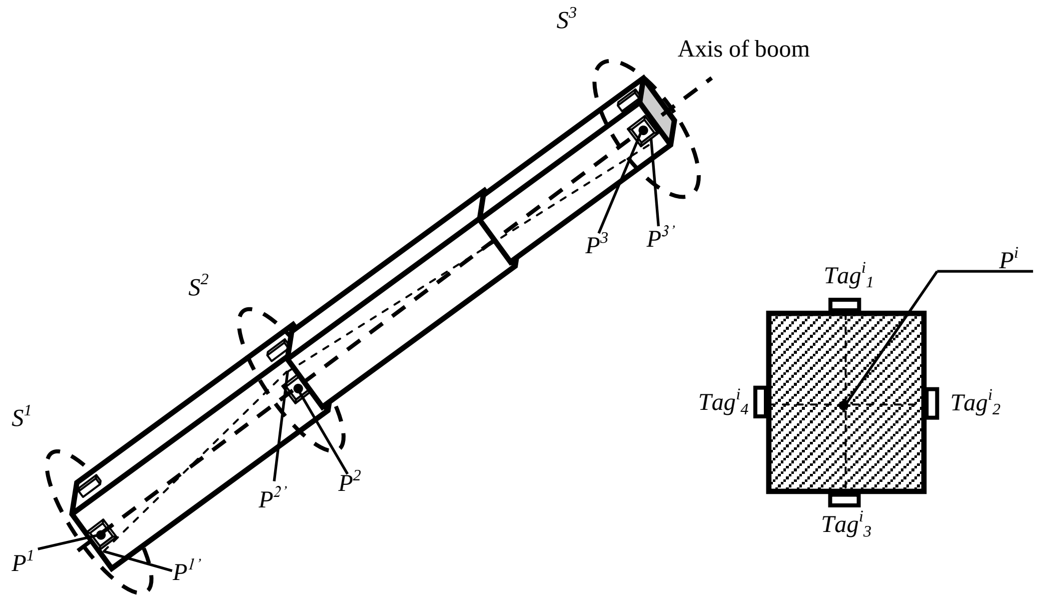
To apply the proposed approach to construction, the UWB RTLS needs to be tested and the collected data need to be processed in near real-time to calculate the crane poses used for collision detection. According to the system requirements defined in Section 3.4, the UWB system setting is explored in depth in Section 5.2. A systematic method of data processing in near real time is discussed in Section 5.3 to investigate how to improve data accuracy and usability. Several tests were applied to investigate the performance of the system and to collect data in different indoor or outdoor environments, as will be discussed in Section 5.4. In Section 5.5, the detailed data analysis of an outdoor crane test is described to calculate crane boom poses.

5.2 UWB SYSTEM SETTING

As discussed in Section 3.4, the requirements of a UWB system for crane safety include: accuracy, visibility, scalability, and real-time, etc. The present section investigates the setting method of the UWB system to satisfy these requirements (Hammad et al., 2010).

5.2.1 TAG LOCATIONS

As discussed in Chapter 3, in the case of monitoring the movement of a hydraulic crane, multiple tags should be attached to its different components to identify its poses. Tags can be attached to the base of the first part of the boom and its tip for easy installation and to avoid damaging the tags. Figure 5-1 shows a schematic boom with three sets of tags (S^1, S^2, S^3) attached to it. Each set S^i includes four tags ($Tag_1^i, Tag_2^i, Tag_3^i$, and Tag_4^i) fixed on each side of the boom. This redundancy improves the visibility of the tags attached to the boom by the sensors when the boom rotates. The approximate location of the center point of a cross section $P^{i'}$ can be calculated by averaging the locations of all or some of the four tags of set S^i. The orientation and the length of the boom can be obtained by connecting the two axis points $P^{1'}$ and $P^{3'}$. The purpose of having an additional set of tags S^2 is to get a third point $P^{2'}$ on the axis of the boom so as to increase the accuracy by having more points along the axis, thereby allowing for the interpolation of the line representing the axis.



(a) Schematic representation of tags on a boom **(b) Boom cross section with set of tags**

Figure 5-1: Locations of tags on the boom and the cross section of the boom

5.2.2 SENSOR COVERAGE

The four sensors of a cell are usually located at the corners of a rectangular monitoring area at a high position facing down towards the center of the area. In the case of monitoring the movement of a large hydraulic crane, the sensors should be fixed at a high position and their pitch angle should be adjusted to cover all the tags attached to the crane, as shown in the upper set of sensors in Figure 5-2. Furthermore, a second cell could be necessary to monitor workers working on the ground because tags attached to them may not be detected by the upper cell due to obstruction by the crane or the limited FoV of the sensors. This two-cell setting to monitor the same area at two elevations is needed in sites where a large vertical area should be monitored.

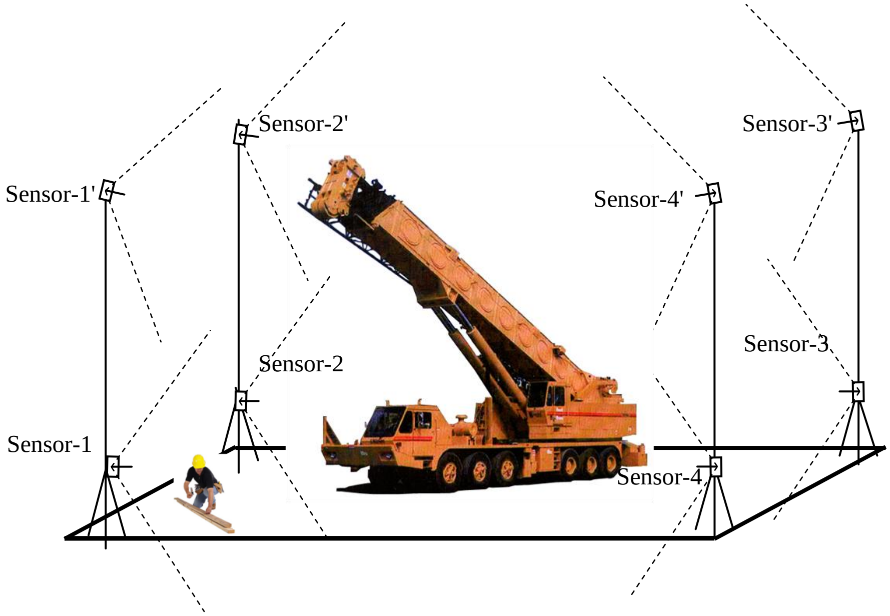


Figure 5-2: Vertical coverage of sensors at two elevations

However, in other cases, using only one cell at the ground level is suitable. In this case, the pitch angle of the sensors should be set to cover all the tags on site. Figure 5-3 shows a 2D projection of a sensor facing a crane. The sensor having a vertical FoV of ±β is mounted on a tripod at a height H_s with a pitch angle of θ. The areas out of the lines of sight are not covered by this sensor but can be covered by other sensors in the same cell. The working range of the crane should be considered to decide the appropriate position and the orientation of the sensors based on the maximum boom length and maximum tip height.

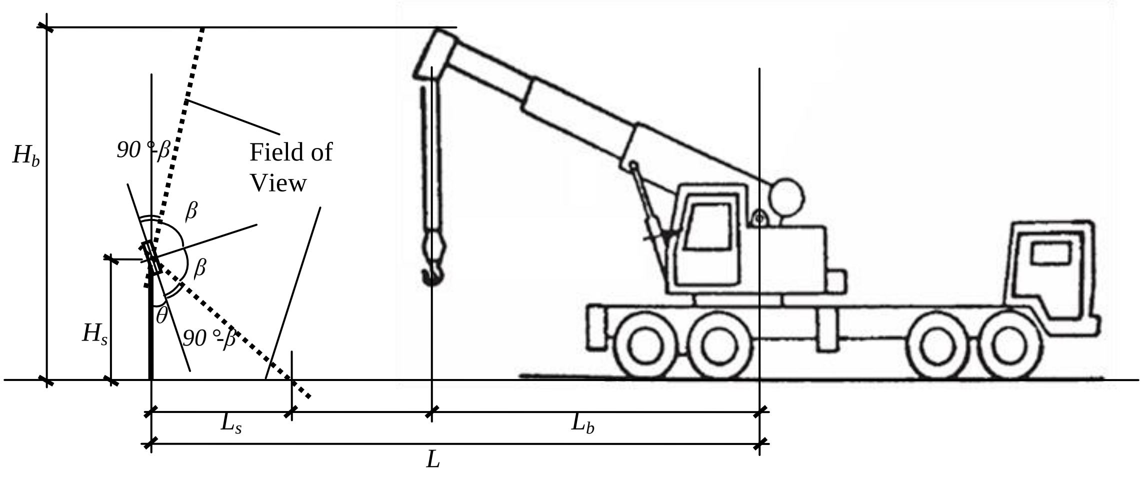


Figure 5-3: Parameters for defining sensor position and orientation

The horizontal distance between the sensor and the base of the boom *L* should satisfy the following condition:

$$L \geq H_s \tan(90° - \beta + \theta) + L_b \qquad (5\text{-}1)$$

where L_b can be simplified as the maximum boom length L_{max} when the boom is almost horizontal. The height of the sensor H_s should meet the requirement of covering the height of the boom tip H_b:

$$H_s + (L - L_b) / \tan(90° - \beta - \theta) \geq H_b \qquad (5\text{-}2)$$

H_b is based on the maximum angle to the ground of the boom according to the working range of the crane. In an extreme scenario, H_b can be replaced by the maximum height of the boom H_{max}. Based on these conditions, a set of *L*, H_s and θ can be defined to improve the coverage of the sensors. In practice, the system setting will start by assuming the initial values for H_s and θ and the value of *L* will be the larger of the two values calculated from inequalities (5-1) and (5-2). Furthermore, the size of a cell should satisfy the conditions of the maximum range of the UWB system and the length of the data and timing cables used for networking between the sensors as explained in Section 3.4.

If information about the crane working range is not available, the maximum coverage above the sensor is needed, and therefore the upper line of sight should be vertical. The simulation software (Autodesk Softimage) used for crane path planning can be also used to visually check the sensor coverage with respect to tags attached to the crane before the actual setting of the UWB system.Figure 5-4 shows an example of a test setting for monitoring a crane. This setting includes the locations of the sensors and the tags with respect to an accurate model of the crane. The lines, connecting the sensors to the tags attached to the boom and the hook, are used for visibility checking.

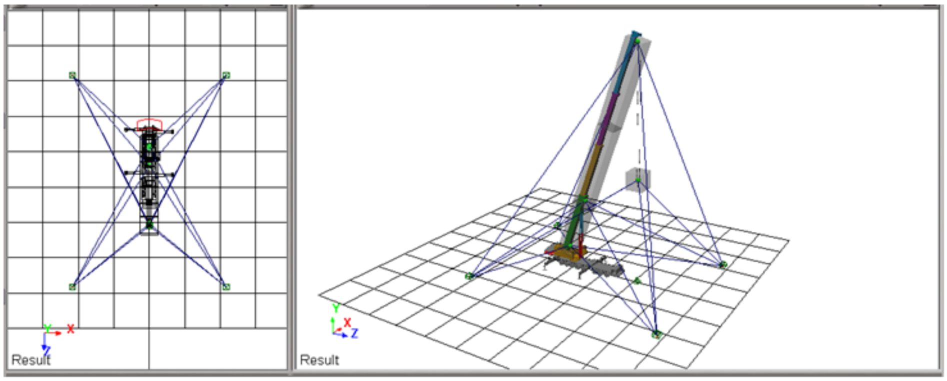


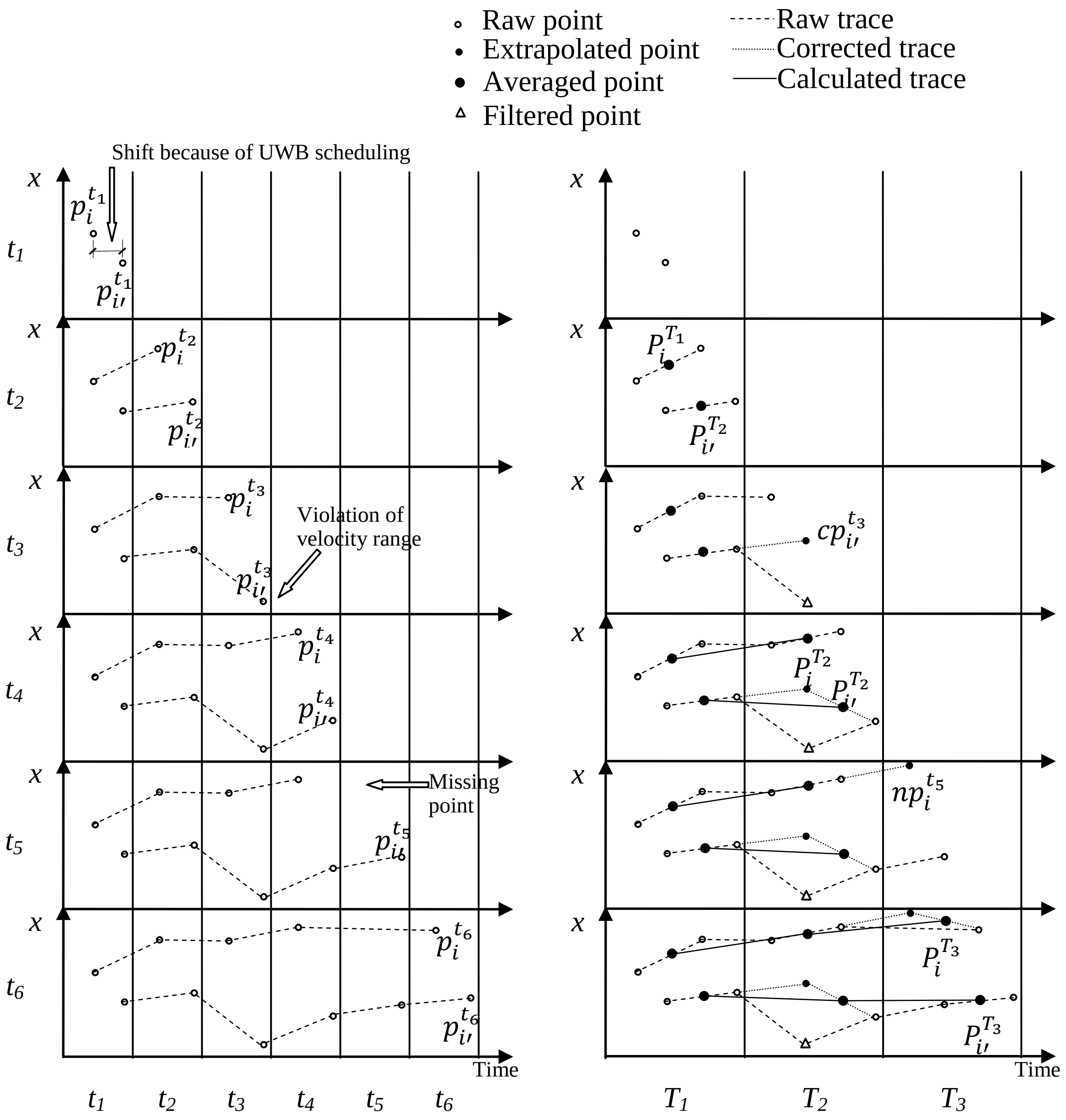
Figure 5-4: Example of a test setting for monitoring a crane

5.3 UWB DATA PROCESSING

Raw location data captured from the UWB system cannot be directly used to compute the pose of the monitored object they are attached to for the following reasons:

(1) **Lack of synchronization:** The data from different tags are not synchronized. As explained in Section 3.4, the UWB radio signals are emitted from each tag based on a precise schedule where only one tag can emit a signal at any point in time in a predefined cycle. To approximately synchronize the locations of different tags, we define a small time period *T* based on the actual update rate *r* of a tag. Assuming $t = 1/r$, *T* should be equal to *t* or a multiplication of *t* big enough to capture at least one reading of each tag in the UWB cell and small enough for near real-time applications (e.g., if *r* is high enough and the application can tolerate a delay of $\delta t = 2t$, then *T* could be set to $2t$). If more than one location is captured for the same tag within *T*, these locations can be averaged to obtain a single reading for that period. Figure 5-5 shows the near real-time location data processing of two tags (Tag_i and $Tag_{i'}$) in the simplified case of tags allowed to move only on the *x* axis, and where $t = 1/r$ and $T = 2t$. Figure 5-5(a) shows the raw traces where points $p_i^{t_j}$ and $p_{i'}^{t_j}$represent the locations of Tag_i and $Tag_{i'}$ at time $t_{j,}$ respectively. Figure 5-5(b) shows the processed traces where points $P_i^{T_k}$ and $P_{i'}^{T_k}$represent the average locations of Tag_i and $Tag_{i'}$ at time T_k, respectively. As shown in the upper-left part of Figure 5-5(a), there is a shift in the timing of the readings of the two tags because of the scheduling of the UWB system.

(2) **Accuracy errors:** Each tag location has certain errors because of radio reflections, etc. These accuracy errors could be filtered in two stages: First, they could be filtered from raw locations captured by the UWB system based on the data of a single tag before synchronization. Second, they could be filtered after synchronization by exploiting geometric constraints between the tags attached to the same object.



(a) Raw data **(b) Processed data**

Figure 5-5: Near real-time data processing of two tags

Filtering based on the data of a single tag can be done by using one of the following methods: (a) Applying the filters provided by the UWB system during trilateration (as discussed in Section 3.4); (b) Checking if the reading of a tag location is outside the monitored area or outside the expected height range of the object to which it is attached; (c) Checking the patterns of movement where a location does not satisfy certain heuristic rules. For example, assume that the maximum expected velocity v_{max} of an object is known and that the measured velocity based on the distance between the past captured location $p_i^{t_{j-1}}$ of Tag_i at time t_{j-1} and the new one $p_i^{t_j}$ is out of range. These conditions indicate that $p_i^{t_j}$ has an accuracy error and should be eliminated. This elimination results in a missing-point error that is processed as explained below. Other heuristic rules can be applied based on the specific constraints of the movement of tags, such as the acceleration of movement.

Filtering based on comparing data from different tags can be achieved by applying geometric constraints between multiple tags attached to a solid object at known locations. These constraints can be used to check the accuracy of the location data. For example, in Figure 5-1(b), the calculated distance between the two tags attached to the top and bottom sides of a section of the boom should be almost equal to the actual distance. Figure 5-6 shows a two dimensional example of the actual paths of two tags (Tag_i and $Tag_{i'}$) attached to the same object. These paths are parallel with a fixed distance $D_{ii'}$. The figure also shows the traces based on the locations of tags at time T_k after averaging. It is noticed that all points $P_i^{T_k}$and $P_{i'}^{T_k}$have a certain number of accuracy errors. However, the distances between the traces $d_{ii'}^{T_k}$are expected to be within the range of [$D_{ii'}$ - 2ε, $D_{ii'}$ + $2\ \varepsilon$], where ε is the nominal accuracy of the UWB system (e.g., 30 cm). If $d_{ii'}^{T_k}$ is outside this range, then $P_i^{T_k}$ and/or $P_{i'}^{T_k}$ should be checked for possible elimination. For example, in Figure 5-6, if $d_{ii'}^{T_5}$ is out of range compared with $D_{ii'}$, and $P_{i'}^{T_5}$ has been calculated based on an extrapolated point, there is a higher probability that $P_{i'}^{T_5}$ should be eliminated.

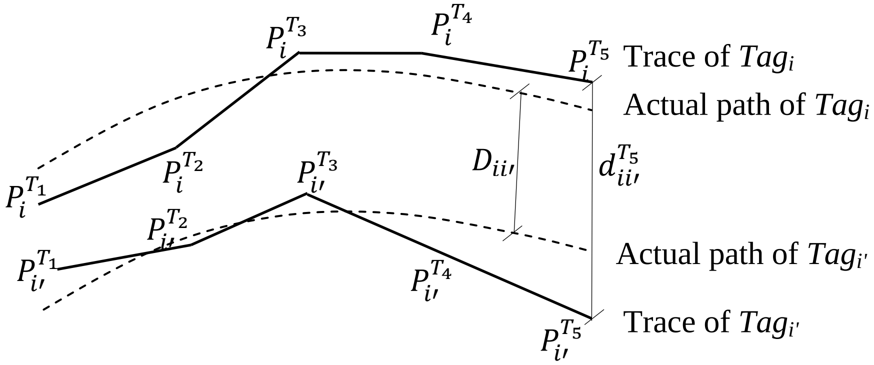


Figure 5-6: Example of using geometric constraints to detect errors

(3) **Missing-point errors:** There can be some missing data because of lack of visibility (i.e., the tag is not detected during a certain period t_j because of the lack of a direct line of sight to some sensors) or because of the filtering of data. Extrapolation can be used after filtering to fill in the missing points for one or more periods assuming that the object is moving with a known velocity. However, this can affect the quality of the location data if several points are missing in a row. Another type of extrapolation/interpolation can be carried out based on a geometric relationship between the tags attached to the same object. For example, in Figure 5-1(a), if the locations of the tags Tag_1^2 and Tag_1^3 at the upper side of cross sections S^2 and S^3 are known at time t_j, the location of Tag_1^1 in cross section S^1 can be calculated by extrapolation. As an example of missing-point errors, Figure 5-5 shows that an accuracy error occurred in t_3, where the velocity of $Tag_{i'}$ exceeded the maximum expected velocity. Extrapolation is used to calculate a new location $cp_{i'}^{t_3}$ for $Tag_{i'}$

based on its previous locations and on the assumption that the tag is moving with the same velocity. Another missing-point error occurred in t_5 for Tag_i. Extrapolation is also used here to calculate a new location $np_i^{t_5}$ for Tag_i at time t_5.

Based on the discussion above, the steps shown in Figure 5-7 can be applied to improve data quality and compute the pose of an object in near real time: (1) The tag IDs are identified and grouped according to their geometric relationship with respect to the objects they are attached to (e.g., tags attached to three sections of the boom). T is defined according to the updated rate for the purpose of synchronization; (2) Readings of each tag are filtered within time t_j according to the methods described in Accuracy Errors; (3) Missing data for each tag caused by missing-point errors or accuracy errors are calculated using extrapolation according to the tag's previous locations; (4) Tag locations are averaged during T_k to synchronize different tags; (5) Errors are filtered according to geometric constraints of multiple tags; (6) After filtering, missing data can be calculated based on extrapolation/interpolation of the data of other tags either in the same group or in different groups as explained above; (7) Locations of multiple tags in the same group are averaged (e.g., averaging the locations of the tags shown in Figure 5-1(b) to get the center point of the cross section); and (8) The pose of the object is calculated according to the positions of the tags attached to it. For example, the pose of the boom can be found according to the calculated center points on the axis of the boom. This pose is used for near real-time motion re-planning as explained in Section 3.3.

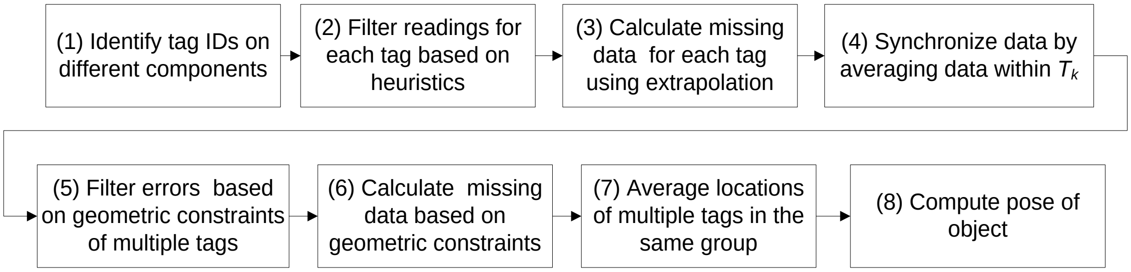
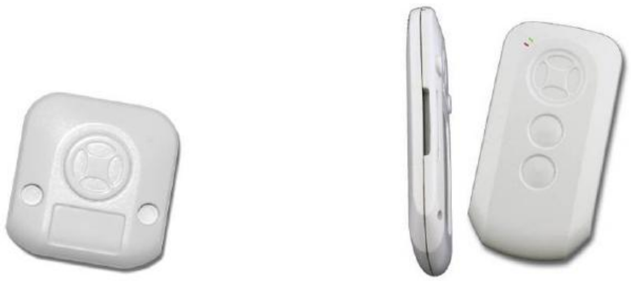

(1) Identify tag IDs on different components
(2) Filter readings for each tag based on heuristics
(3) Calculate missing data for each tag using extrapolation
(4) Synchronize data by averaging data within T_k
(5) Filter errors based on geometric constraints of multiple tags
(6) Calculate missing data based on geometric constraints
(7) Average locations of multiple tags in the same group
(8) Compute pose of object

Figure 5-7: Steps of data processing

5.4 UWB TESTS

In the present research, Ubisense system is used to evaluate the performance of the UWB technology. The system consists of four sensors and 100 tags with an academic price of CA$20,000. A sensor cell is created using these four sensors and several tags. The power of the sensors is supplied by a Power over Ethernet (PoE) switch. Timing cables are used to connect the sensors to synchronize the signals from a tag to different sensors. The sensors are calibrated using a tag as a reference point with known position. Specifications of sensors can be found in Appendix C. Tags are attached to the objects to be tracked. There are two types of tags available, which are slim tags and compact tags. Slim tags are designed to be worn by a person. Compact tags are specially designed for use in harsh industrial environments and include several advanced features: a Light Emitting Diode (LED) for easy identification, a motion detector to instantly activate a stationary tag, and a push button to trigger events. Compact tags are selected in the present research because of their omni-directional antennas and rugged design, which make them more suitable for the tests with cranes. Figure 5-8 shows the two types of tags. The specifications for compact tags can be found in Appendix D.



(a) Compact tag **(b) Slim tag**

Figure 5-8: UWB Tags (Ubisense, 2010)

The sensor cell network should be installed and calibrated properly following the procedures defined in Appendix E. A work sheet for organizing the data of the system calibration is shown in Appendix F, where the data of the system setting are recorded. Location data can be recorded using either a Cell Monitor File (.xcm) or an application called Ubisense Logger, which records the local coordinates (3D) of the target tags. The samples of the Cell Monitor file and the Logger file are shown in Appendices G and H, respectively. The initial cell update rate (*R*) was 40 Hz when the UWB system was purchased in 2008. Later on, it was upgraded to 160 Hz to meet the requirements of this research, for which a high update rate is needed.

5.4.1 LABORATORY TESTS

The laboratory tests were designed for the preliminary testing of the sensors' networking and collecting of near real-time data. Figure 5-9 shows the floor plan of the lab and the connections between four sensors. The solid lines show the data cables connecting the sensors with the PoE switch, whereas the dotted lines show the timing cables. Sensor-1 acts as the master sensor.

A test was applied to a scaled model of a tower crane to simulate a lifting task with tags attached to it. Figure 5-10 shows the tower crane model with one tag attached to the hook and two tags attached to the counterweight. A box was placed near the crane to represent an obstacle. A remote controller was used to swing the boom of the crane and to lift the hook (two DoFs), to avoid collision with the obstacle.

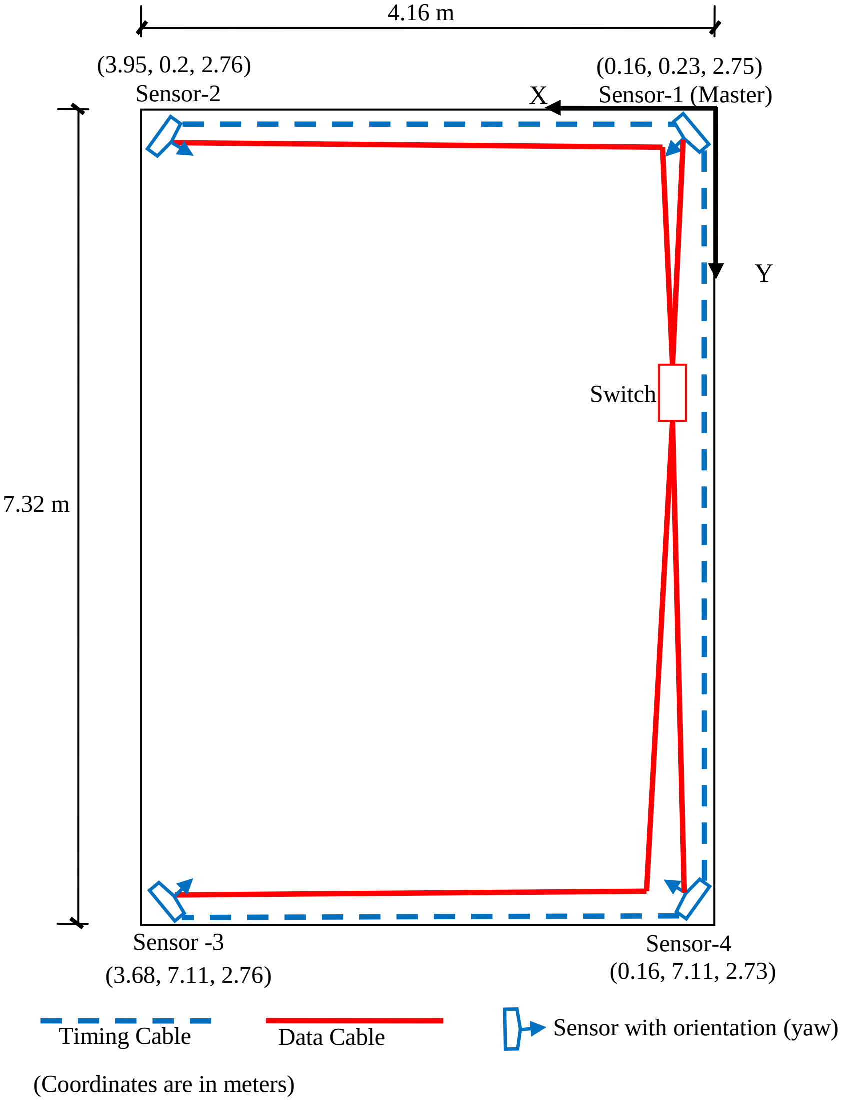


Figure 5-9: Lab settings

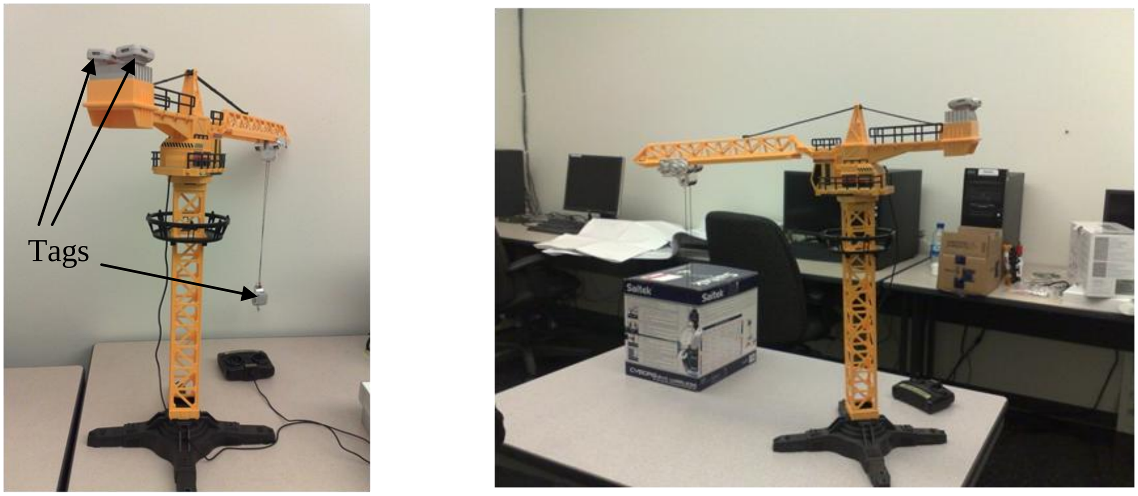


Figure 5-10: Tower crane model with tags

Raw data were collected using the Logger including the tag ID, date, time, and the x, y, z coordinates. Data were collected for about 30 seconds with a tag update rate of 9 Hz and a cell update rate of 40 Hz. Figure 5-11 shows a screen shot of the visualization environment of Softimage, where the path of the hook is shown. The path is smoothed by applying a curve fitting function.

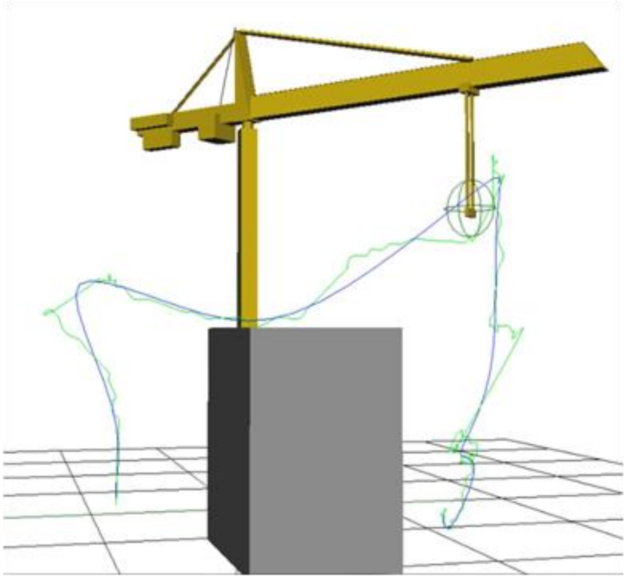


Figure 5-11: Path trace of the tower crane hook

5.4.2 INDOOR TEST

On April 30, 2009, an indoor test was carried out on the 7th floor of the new building of the John Molson School of Business of Concordia University, where the installation of heating, ventilating, and air conditioning (HVAC) ducts was being carried out. The objective of this test was to ensure that the UWB system can be used in an actual construction environment to track workers and equipment. The dimensions and the layout of the ducts are shown in Figure 5-12. The sensor cell was designed as shown in Figure 5-13. Four sensors were fixed at four corners using tripods, and they are connected with data cables and timing cables.

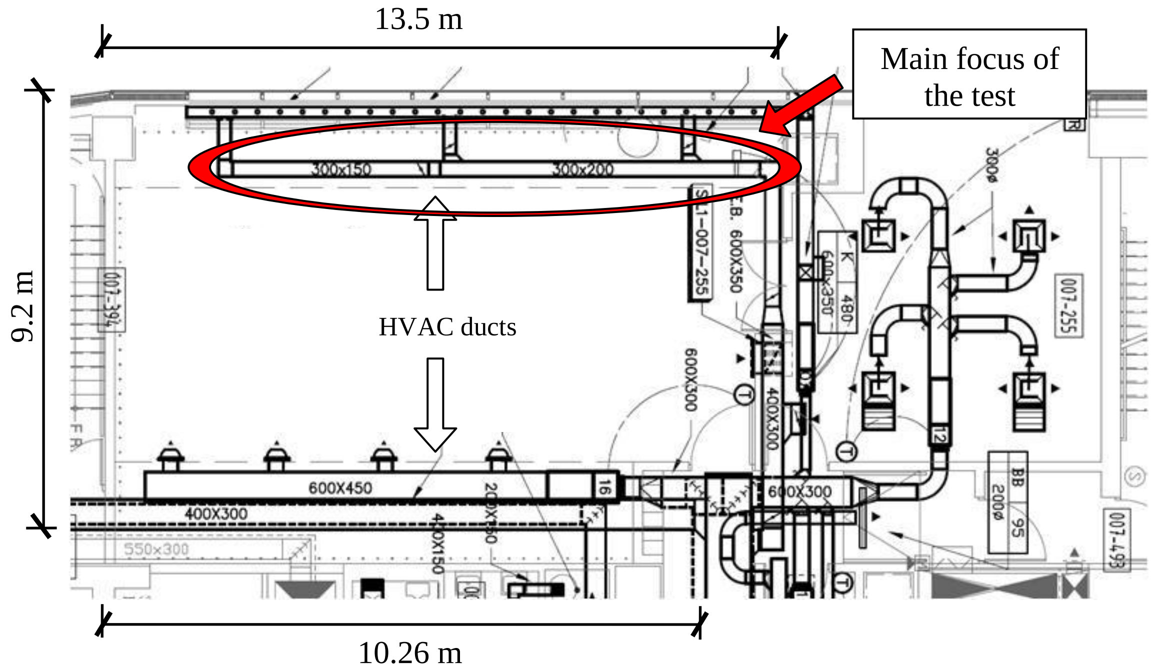
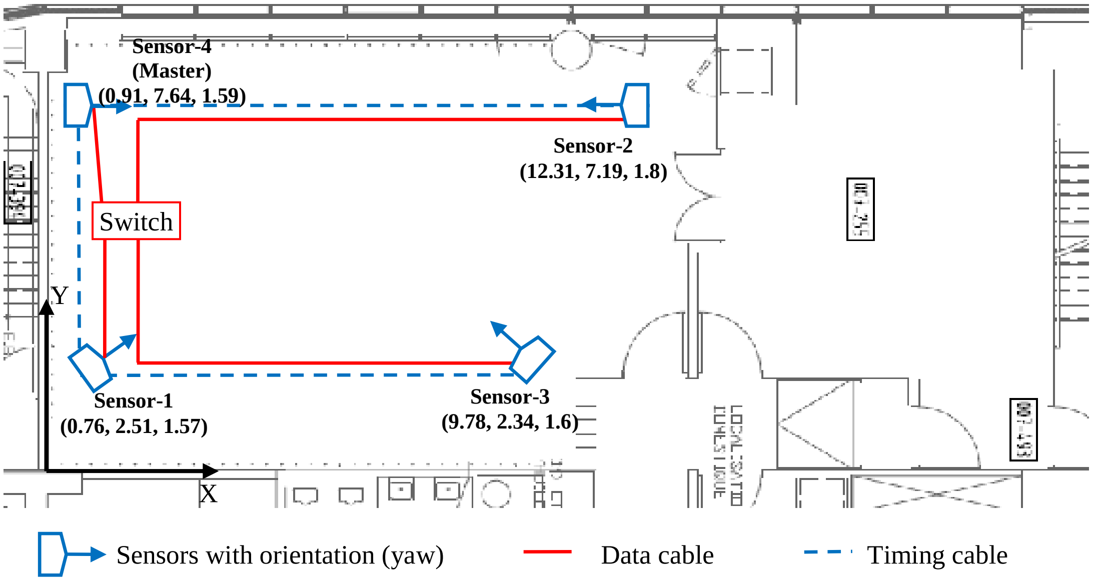


Figure 5-12: Dimensions of the monitored area

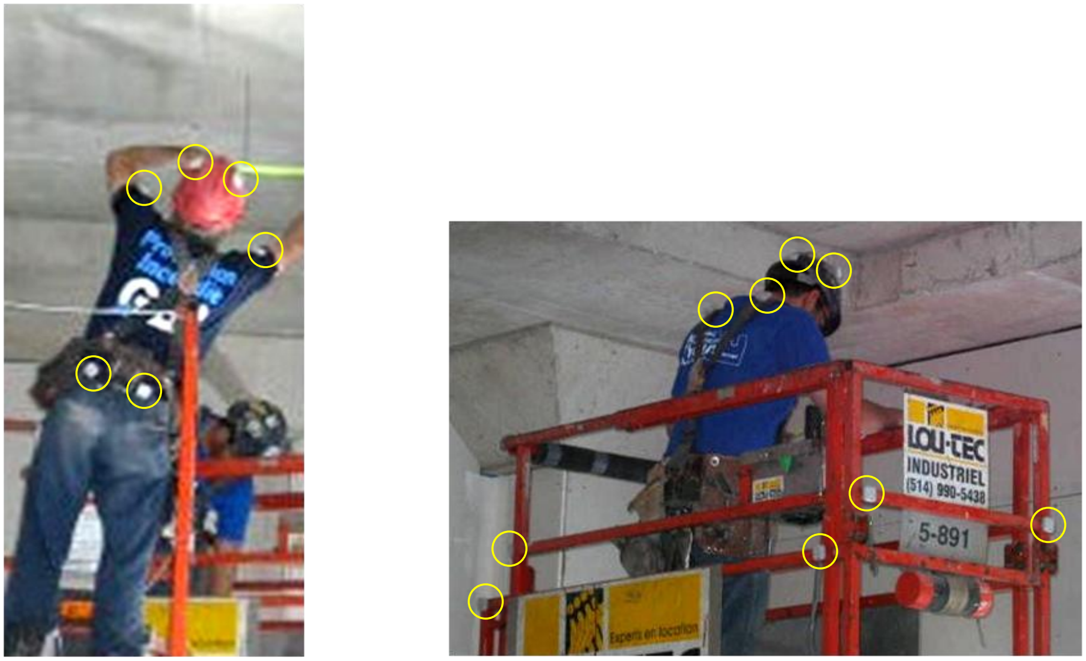


(Coordinates are in meters)

Figure 5-13: Indoor setting of sensor cell

Two workers (Worker-1 and Worker-2) and a scissor lift were monitored with six, four, and eight tags, respectively. Figure 5-14 shows the locations of the tags (within the circles). For a certain period of time, the two workers were using two scissor lifts, and they were working on different tasks at two work zones very near each other (Figure 5-15). They had to move carefully to avoid collision.

Location data were collected during two periods of 38 min (from 11:50 to 12:28) and 57 min (from 12:40 to 13:37). In addition, a video was taken to record the workers' movements, which helps in analyzing the data collected from the UWB system. For example, Worker-1 moved along a narrow work zone while doing the measurements and the marking on the floor, as shown in Figure 5-16 with pictures extracted from the video. At time T_1 Worker-1 was reviewing plans, at T_2 he was on the scissor lift and was working near the ceiling, at T_3 he was measuring the distance to the wall, and at T_4 he started moving to another room.



(a) Worker-1

(b) Worker-2 and scissor lift

Figure 5-14: Tags attached to workers and scissor lift

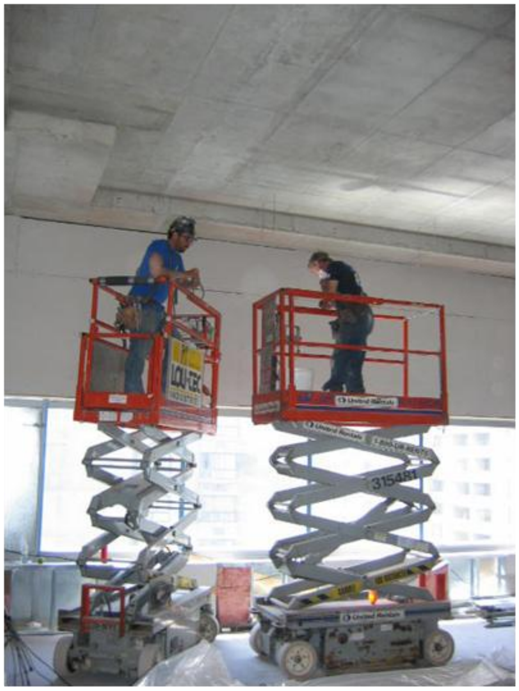


Figure 5-15: Two scissor lifts working in the same area

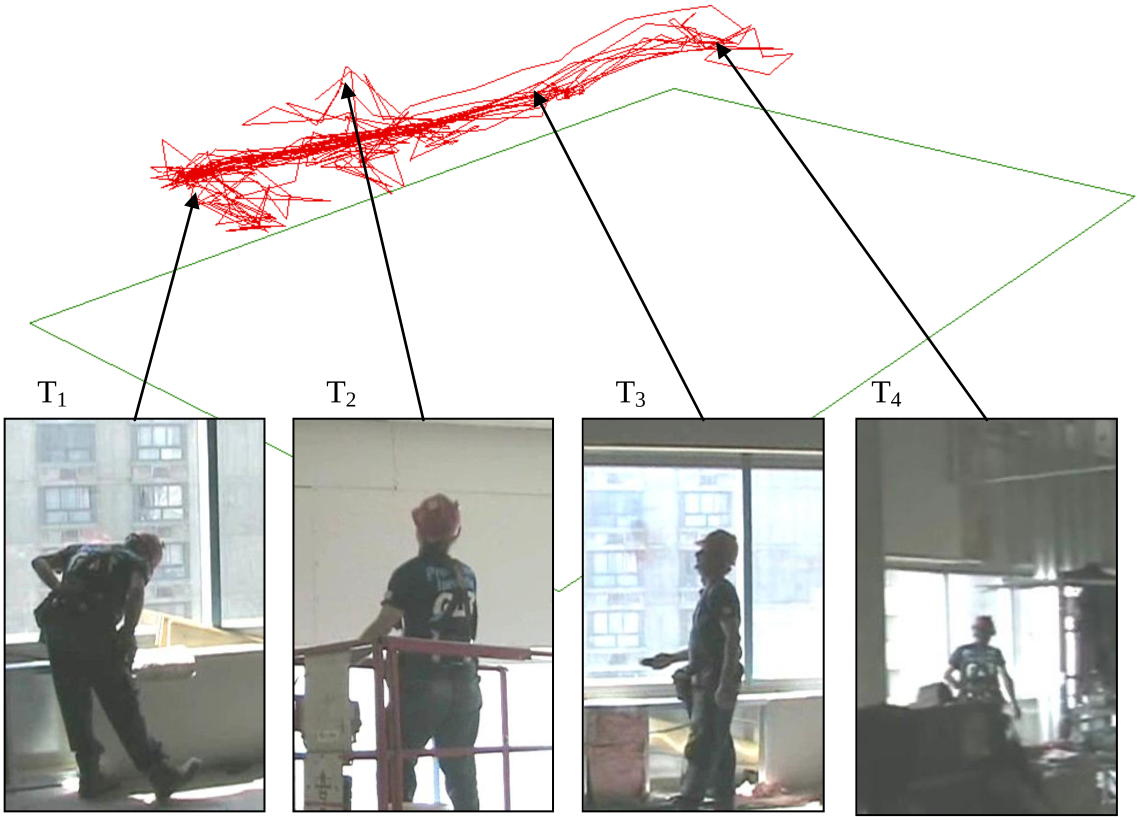


Figure 5-16: Trace of worker-1

Since safety is the main concern of the present research, the collected data were analyzed to check whether the operation regulations were followed properly. For example, to prevent a tip-over, it is not allowed to travel to a task location with the scissor lift in an elevated position (SSPC, 2009). With the location data of the tags attached to the scissor lift, it was possible to identify the height of the scissor lift and its velocity at any time. Figure 5-17 shows the traces of two tags attached to two corners of the scissor lift during 15 min. From this figure, it can be seen that there were two main work zones in which the scissor lift was working, and, at a certain time, it moved from one work zone to the other. The traces between the two work zones show the movement of the equipment. The distance between the two traces is approximately equal to the width of the scissor lift.

The average speed was 0.27 m/s and the average height of these tags was about 1.46 m during the movement, which means the scissor lift was not in an elevated position while moving.

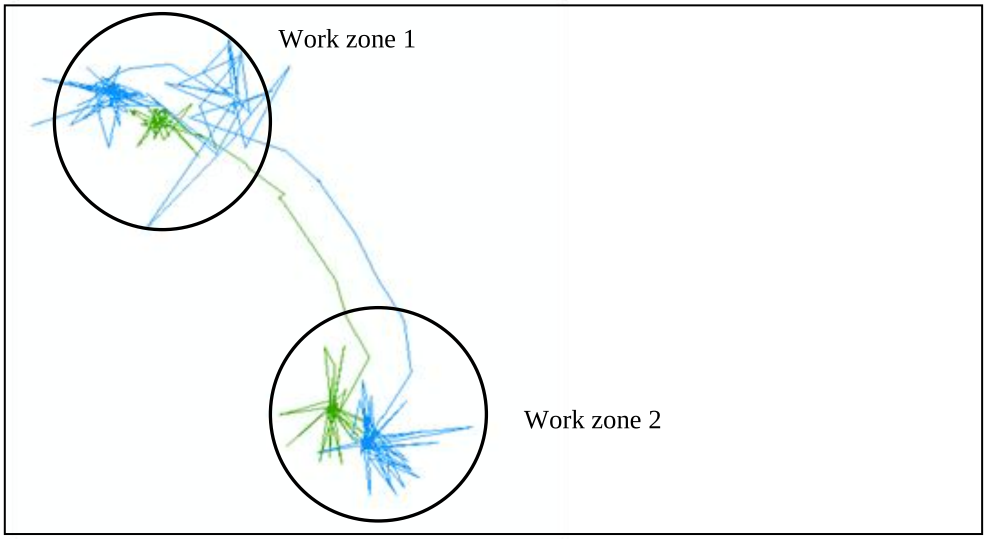
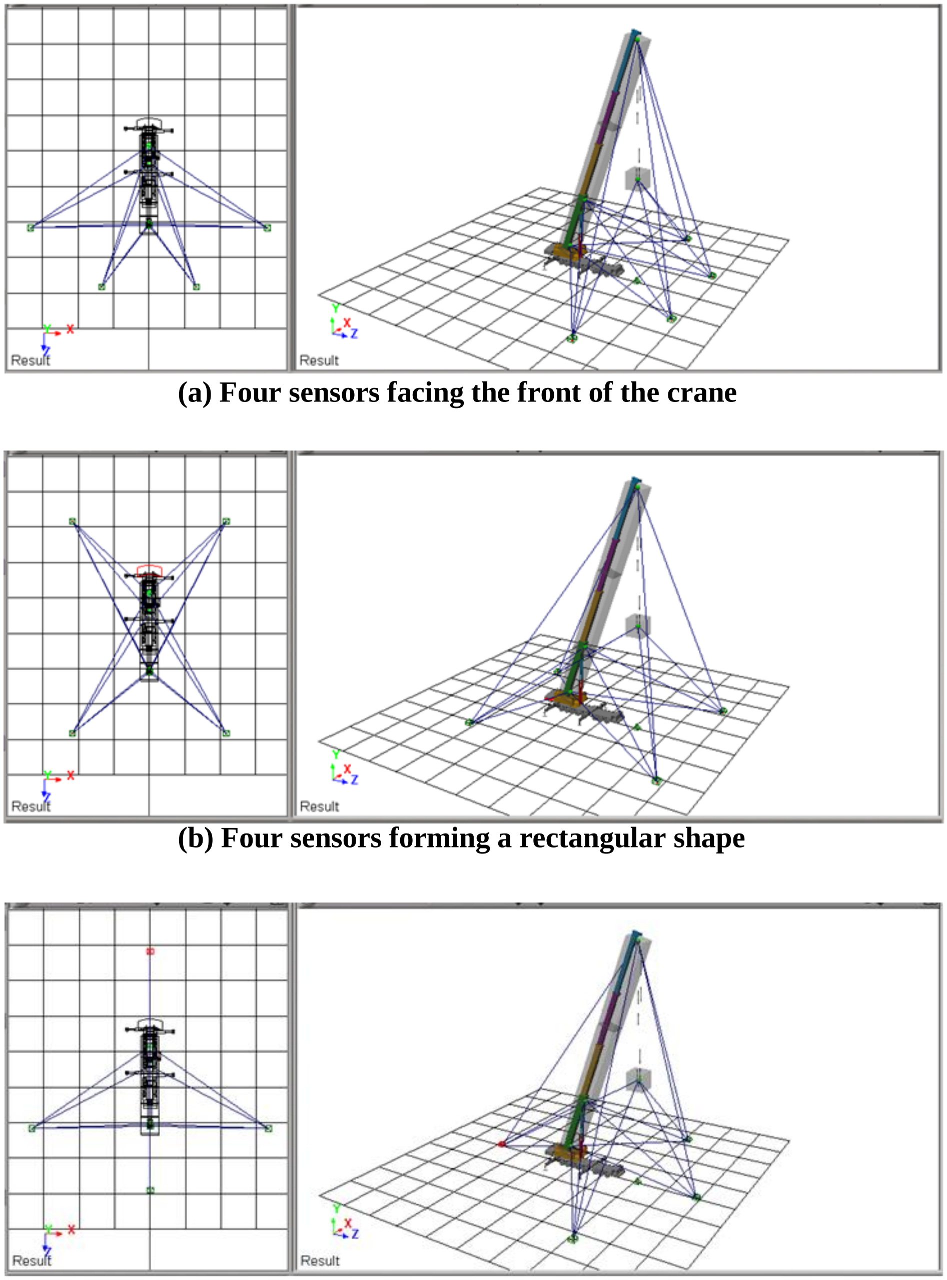


Figure 5-17: Traces showing two tags attached to the scissor lift

5.4.3 OUTDOOR TESTS

The objectives of the outdoor tests were: (1) to investigate the different aspects of the proposed methods for setting the UWB system as discussed in Section 5.2, mainly the setting of multiple tags on a crane and the selection of the number of tags and their update rates to satisfy the requirements of accuracy, visibility, scalability and real-time tracking; and (2) to calculate the crane boom poses in near real time according to the data processing approach discussed in Section 5.3. Before testing in an outdoor environment, different scenarios for studying the visibility conditions were designed for locating the sensors and tags on site using Autodesk Softimage. Four sensors were installed around the crane to maximize the probability that more tags on the boom can get line of sight to at least two sensors. The following three cases were considered: (1) Four sensors facing the front of the crane; (2) Four sensors forming a rectangular shape around the crane; and (3) One sensor facing the back of the crane, and three sensors facing its front, as shown in Figure 5-18. The field of view of each sensor was visualized; therefore, different configuration scenarios were simulated and tested to produce more reliable results.



(a) Four sensors facing the front of the crane

(b) Four sensors forming a rectangular shape

(c) One sensor facing the back of the crane and three sensors facing its front

Figure 5-18: Setting of sensor locations

First Outdoor Test

The first outdoor test was done on the yard of a crane company on December 12, 2008, with the temperature at - 4 ℃. Figure 5-19 shows the location of the sensors following the design in Figure 5-18(c). Tags were attached to the boom at eight locations and to the hook at two locations. Figure 5-20 shows some of these tags.

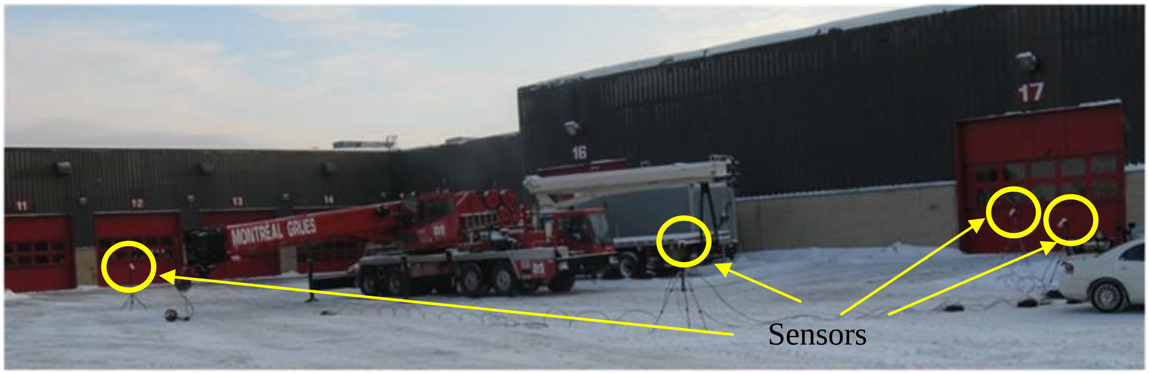


Figure 5-19: Locations of sensors

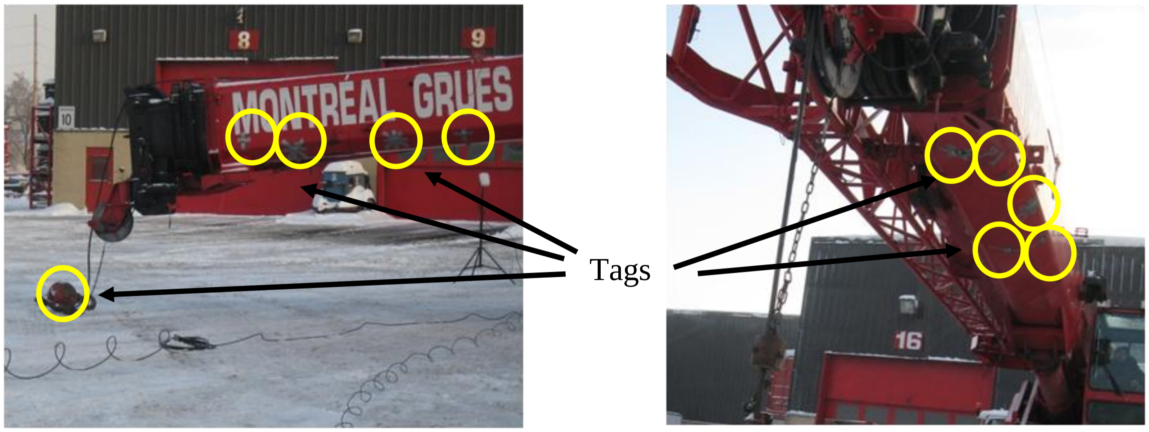


Figure 5-20: Tags attached to the boom and the hook

The result of the test was not satisfactory due to calibration problems. Furthermore, because of the cold weather, it was difficult to fix tags to the boom using the adhesive pads provided by Ubisense. Although this first outdoor test was not successful, many lessons were learned, such as the need for quick and secure magnetic mounts for the tags.

Second Outdoor Test

The second test was done on the yard of the same crane company on December 4, 2009, using a TMS300 crane (GUAY, 2010). The UWB system was upgraded from the low update rate (40 Hz) to a high update rate (160 Hz) to better fit near real-time safety requirements. The test was designed in detail, including the sensors' locations, tags' locations, cables' connections, system calibration, data filtering, and task description. Furthermore, several indoor tests were done to test the stability of the UWB system, the influence of the magnetic mounts of the tags, etc.

Setting of sensors

In this outdoor test, where the focus is on the crane, only four sensors at the ground elevation are deployed by adjusting the pitch angle to capture the boom movement, while satisfying the inequalities described in Subsection 5.2.2. Before going to the site, the UWB system setting was investigated to fulfill the requirements described in Section 3.4. The antenna pattern of the sensor is $\alpha = \pm 90°$ in the azimuth and $\beta = \pm 50°$ in the elevation. The yaw angles of the sensors were adjusted to face the center of the area. The pitch angle and height of the sensors were approximately set to $\theta = 20°$ and $H_s = 1.5$ m, respectively. The maximum boom length of the crane is 110 ft (33.5 m), and the minimum and maximum angles to the ground of the boom are 10° and 80°, respectively. Based on the working range of the crane, when the boom is fully extended and reaches the highest point, the corresponding H_{max} = 120 ft (36.58 m), and L_b = 20 ft (6.10 m). According to inequality (5-1), and taking L_b = 110 ft (33.5 m) L should be greater than 36.1 m. According to inequality (5-2), to cover the maximum height of the boom tip, L

should be greater than 18.86 m. Using the virtual crane model in Softimage, it was possible to verify that the tag at the tip of the boom was within the FoV of the sensors.

At the site of the test, the actual pitch angles of the sensors were in the range of 20 ° to 26 °, and the heights of the sensors were in the range of 1.45 m to 1.67 m. Due to the limitations of the yard dimensions, which are approximately 18 m by 22.5 m, the crane is positioned in a way to make *L* approximately 21 m. However, this does not satisfy inequality (5-1). Therefore, the operator of the crane was informed to limit the extension of the boom when lifting the object. Figure 5-21 shows the setting of the sensor cell for this test with the timing and data cables and the locations and yaw orientations of the four sensors. Figure 5-22 shows part of the site with the crane and one sensor (Sensor-2). A car was positioned as an obstacle on the moving path of the crane.

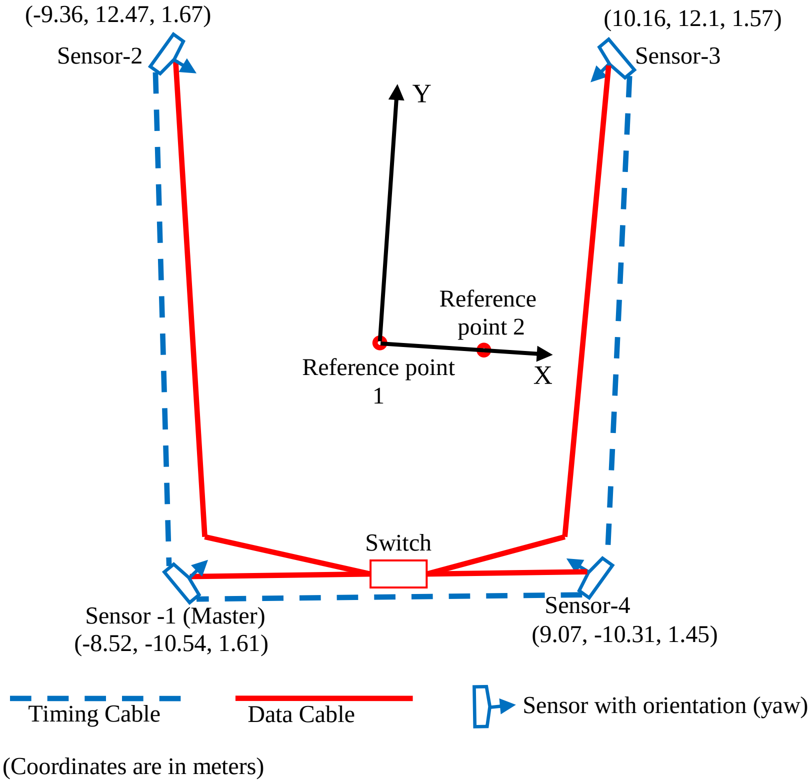


Figure 5-21: Outdoor setting of sensor cell

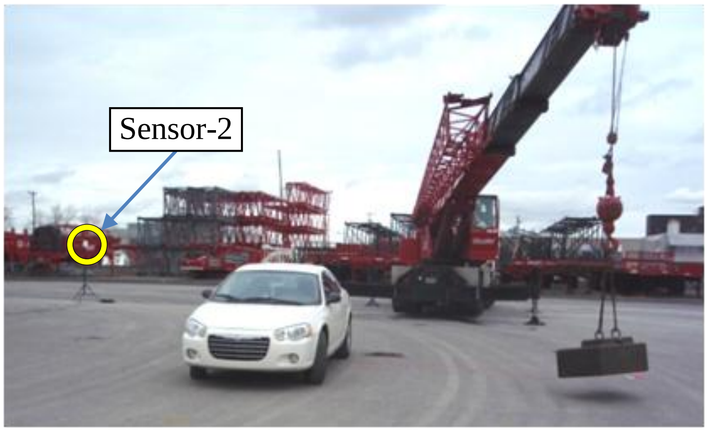
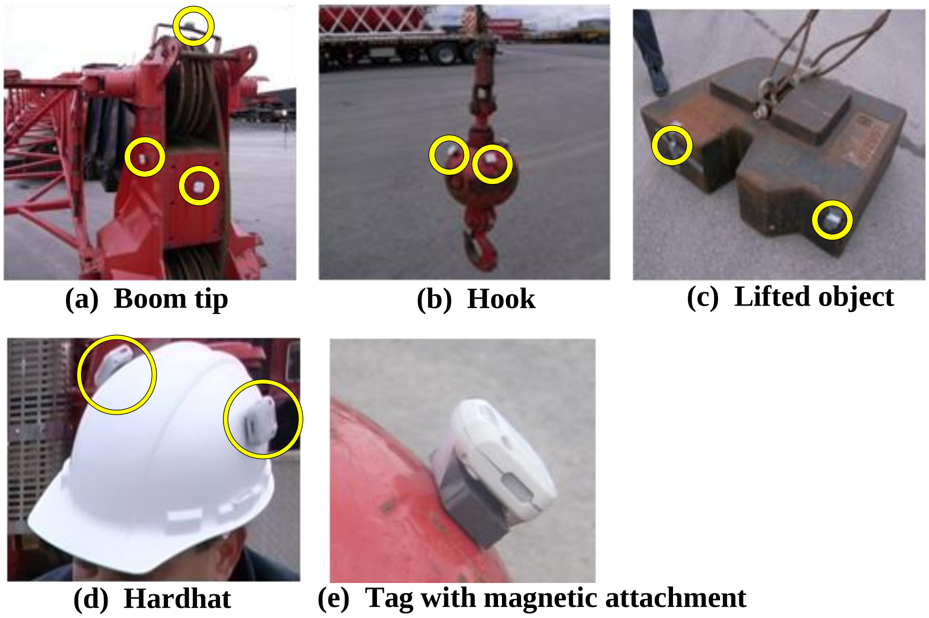


Figure 5-22: Crane with obstacle and one sensor

Setting of tags

Twenty-two tags were attached to the crane’s body, with three sets of tags (12 tags) attached to the boom, as shown in Figure 5-1. Other tags were attached to the outriggers, operator cab, hook, and lifted object. Moreover, four tags were attached to the hardhats of two workers (two tags on each hardhat) to track their movements on site. Figure 5-23 shows the tags attached to different objects. Figure 5-24 shows the pictures of tags Tag_1^2, Tag_2^2, and Tag_4^2 near the end of the first part of the boom.

To test the scalability of the UWB system, which has a high cell update rate of 160 Hz, we kept 52 additional tags in the same area so that the total number of tags in the cell was 74. According to the inequality (3-3) introduced in Section 3.4, the time slot interval should be set to 128, where the update rate is 1.2 Hz for each tag according to inequality (3-4). According to Table 3-2, the nominal update rate assigned by the system to each tag should be in the range of 1.2 to 2.4 Hz. By observing the collected data, it was found that the actual update rate was about 2 Hz. Therefore, in this test, the syncronization of multiple tags was based on t = 500 ms. An information data filter provided by Ubisese was used to improve accuracy with a motion model of position and Gaussian noise on position (Ubisense Manual, 2009).


(a) Boom tip (b) Hook (c) Lifted object

(d) Hardhat (e) Tag with magnetic attachment

Figure 5-23: Tags attached to different objects

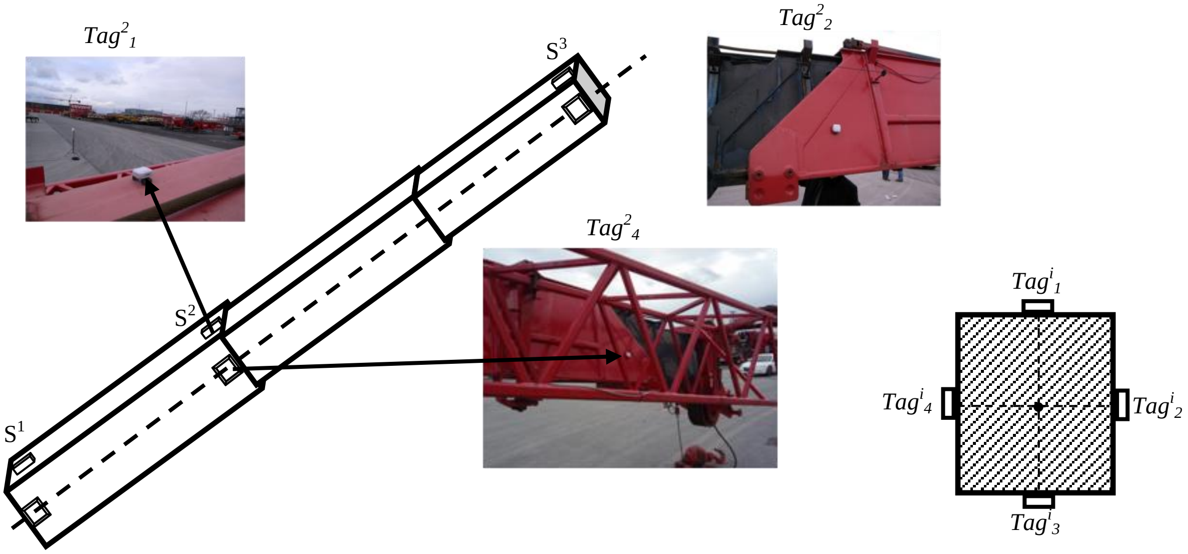
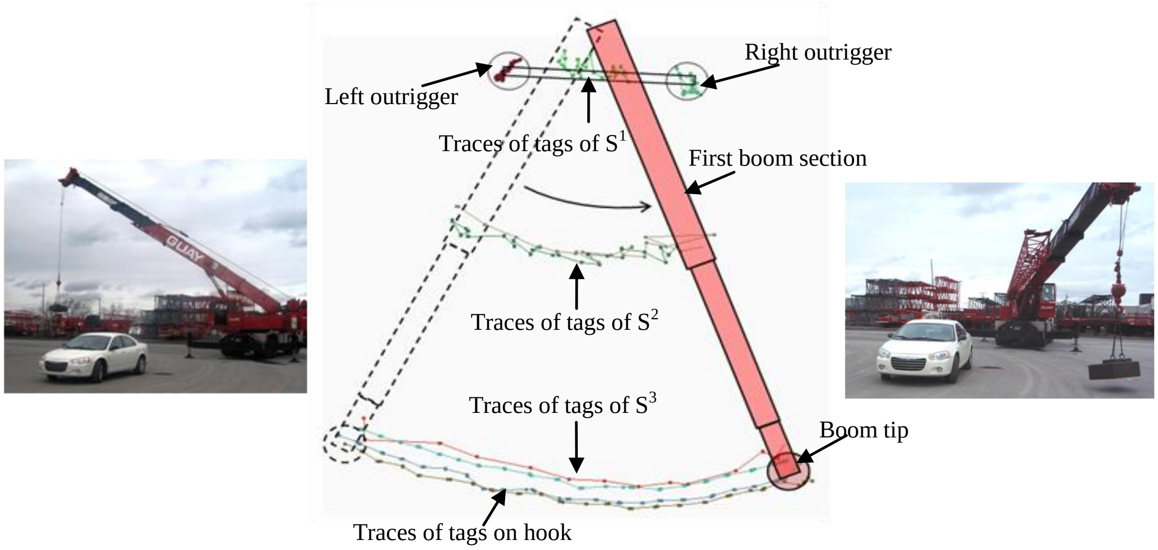


Figure 5-24: Tag position of cross section S^2 on the boom

Task description

The total duration for the outdoor test was about two hours, including the system configuration, measurement, calibration, moving the crane into the monitored area, and collecting data during the crane operation. The task given to the crane operator was to lift an object from one place to another by swinging and raising up the boom while avoiding the car on the path (Figure 5-22).



(a) Initial crane pose **(b) Crane poses based on raw data** **(c) Final crane pose**

Figure 5-25: Part of the raw data collected

During the lifting, the length of the boom and the length of the cable were fixed. A part of the raw data collected in the test is shown as traces in Figure 5-25(b). The tags shown in three cross sections are Tag_4^1, Tag_1^2, Tag_3^2, Tag_1^3 and Tag_4^3. The data analysis of this test is discussed in the next section.

5.5 DATA ANALYSIS OF THE SECOND OUTDOOR TEST

The tags attached to the top of the boom had very good visibility and better accuracy (less noisy data) compared with those attached to the bottom and the sides of the boom, which had a large number of missing points and noisy data. The raw UWB data were processed following the steps explained in Figure 5-7 in order to get the poses of the boom. However, because of the low update rate (2 Hz) and the large amount of missing data, some steps were not always applicable (e.g., averaging or extrapolation at a certain time period). Nevertheless, the redundancy provided by having multiple tags on the boom made it possible to calculate the poses of the boom based on the traces as shown in Figure 5-31.

5.5.1 VISIBILITY ANALYSIS

As mentioned before, twenty-two tags were attached to the crane and four tags were attached to the hardhats of workers. Within the recording time of 36 seconds, which was the duration of the lifting task, the tags obtained different numbers of updates, as shown in Table 5-1. The measured update rate r' is calculated by dividing the number of updates of each tag by 36 seconds. Due to missing data, some tags have lower update rates than other tags. From this table, it can be seen that tags on the upper and bottom sides of the boom had better visibility than the ones attached to the side surfaces. As shown in Figure 5-24, tags attached to the side with a truss structure (i.e., Tag_4^1, Tag_4^2, and Tag_4^3) received fewer updates compared to other tags in the same cross section. This could be explained by radio signal reflections on the truss. Tags attached to the cab also showed bad visibility because the rotation of the cab could not guarantee direct line-of-sight from two sensors. All the four tags attached to the hook had excellent visibility. Tags attached to the lift object had bad visibility that may be explained by the lack of direct line of sight. One tag attached to the left outrigger had good visibility.

Table 5-1: Tag updates

Tag Location		Tag name	Number of updates in 36 seconds	Measured update rate r' (Hz)
Boom	S^1	Tag_1^1	61	1.7
		Tag_2^1	37	1.0
		Tag_3^1	**73**	2.0
		Tag_4^1	24	0.7
	S^2	Tag_1^2	**74**	2.1
		Tag_2^2	50	1.4
		Tag_3^2	**70**	1.9
		Tag_4^2	24	0.7
	S^3	Tag_1^3	**74**	2.1
		Tag_2^3	18	0.5
		Tag_3^3	42	1.2
		Tag_4^3	20	0.6
Cab	C_1		12	0.3
	C_2		39	1.1
Hook	H_1		**74**	2.1
	H_2		**74**	2.1
	H_3		**73**	2.0
	H_4		**73**	2.0
Lift	L_1		27	0.8
	L_2		20	0.6
Outrigger	right	O_r	34	0.9
	left	O_l	**74**	2.1
Hardhat-1	$H^1{}_r$		50	1.4
	$H^1{}_l$		24	0.7
Hardhat-2	$H^2{}_r$		**72**	2.0
	$H^2{}_l$		61	1.7

5.5.2 ACCURACY ANALYSIS

The location data of two static tags on the two outriggers were analysed to reveal the accuracy of the system based on the measured coordinates of these tags. Table 5-2 shows the mean difference and the standard deviations in three directions of these data. The accuracy of the data collected for these two tags is around 25 cm. The tag on the left outrigger has more readings than the tag on the right outrigger (74 vs. 34 readings, as shown in Table 5-1), thereby contributing to the more accurate results of the left tag.

Table 5-2: Mean difference and standard deviation in three directions for static tags

Tag name	Mean difference in X (m)	Mean difference in Y (m)	Mean difference in Z (m)	Standard deviation in X (m)	Standard deviation in Y (m)	Standard deviation in Z (m)
O^l	-0.129	-0.089	-0.200	0.2084	0.1976	0.2787
O^r	-0.212	-0.085	0.248	0.1917	0.1807	0.2060

5.5.3 REMOVING ERRORS AND FILLING IN THE MISSING DATA

Example of filtering readings of tags based on heuristics

Based on the steps defined in Figure 5-7, errors have been identified and eliminated in near real-time. After identifying tag IDs on different crane components, the heuristic of the maximum expected velocity v_{max} can be set for specific tags. Based on our observation, the average velocity of tags in cross section S^2 of the boom is about 0.5 m/s. By adding the UWB system error, which is about ±30 cm in all readings, v_{max} used to filter the UWB readings in near real time for tags in S^2 is set to 1.5 m/s. Taking Tag_3^2 as an example, there is a sudden movement in the Z direction at t_{24}, as shown in Table 5-3 and Figure 5-26, and the velocity of Tag_3^2 is calculated as 4.53 m/s, which by far exceeds v_{max}. Therefore, the reading at t_{24} is rejected and replaced by a location calculated based on extrapolation according to the Δ value in each dimension (X, Y, and Z). The purpose of calculating the difference in each dimension individually is that the accuracies in these three dimensions are different, and based on our observation, the accuracy in the Z dimension is lower than those in the X and Y dimensions. The average Δ value (μ_{Δ}) and the standard deviation (σ_{Δ}) are calculated according to previous data history during the last 5 seconds. Only points with a Δ in any of the X, Y, Z dimensions that is out of the range of [μ_{Δ} - 2σ_{Δ}, μ_{Δ} + 2σ_{Δ}] are corrected in those specific dimensions using extrapolation from two previous points. This range contains 95.44% of the data assuming that the differences follow a normal distribution (Allen, 2006). As shown in Table 5-3, the Δ values in the Y and Z dimensions are out of range at t_{24}, where Δ_y= 0.21 m, Δ_z= 2.13 m, and out of the ranges of [-0.073 m, 0.059 m] and [-0.184 m, 0.220 m], respectively, where $\mu_{\Delta y}$= -0.007 m, $\mu_{\Delta z}$= 0.018 m, $\sigma_{\Delta y}$= 0.033 m, $\sigma_{\Delta z}$= 0.101 m (these values are at t_{23}). Extrapolation is done based on the location data at t_{22} and t_{23} for those two dimensions (Y and Z). It should be noticed that the information filter used for all the tags in the Ubisense system always predicts location data based on previous readings; therefore, the data collected for the next time periods (from t_{25} to t_{31}) are all affected by the prediction based on errors, and they have to be recalculated by extrapolation similar to the point at t_{24} to avoid exceeding v_{max}. This extrapolation results in creating new data as shown in the highlighted part in Table 5-3. The results are shown in Figure 5-26, where the raw data and the processed data are plotted. The big jump in the Z dimension is eliminated.

Table 5-3: Data processed for Tag_3^2 in real time

time	x (m)	y (m)	z (m)	v (m/s)	Δ			Average Δ (μ_Δ)			Std. deviation (σ_Δ)			time	x_n (m)	y_n (m)	z_n (m)	v (m/s)	Δ			Average Δ (μ_Δ)			Std. deviation (σ_Δ)		
					Δx	Δy	Δz	Δx	Δy	Δz	Δx	Δy	Δz						Δx	Δy	Δz	Δx	Δy	Δz	Δx	Δy	Δz
t_{23}	0.400	5.43	3.750	0.047	0.020	0.000	0.010	0.082	-0.01	0.018	0.170	0.033	0.101	t_{23}	0.400	5.430	3.750	0.047	0.020	0.000	0.010	0.153	-0.05	0.063	0.250	0.033	0.101
t_{24}	0.700	5.64	5.880	4.531	0.300	0.210	2.130	0.085	-0.02	0.035	0.180	0.080	0.669	t_{24}	0.700	5.430	3.760	0.629	0.300	0.000	0.010	0.153	-0.02	0.035	0.180	0.033	0.100
t_{25}	0.710	5.64	5.870	0.030	0.010	0.000	-0.010	0.115	0.002	0.248	0.183	0.079	0.673	t_{25}	0.710	5.430	3.770	0.030	0.010	0.000	0.010	0.115	-0.02	0.036	0.183	0.034	0.010
t_{26}	1.040	5.22	5.630	1.228	0.330	-0.42	-0.240	0.104	0.005	0.215	0.193	0.156	0.685	t_{26}	1.040	5.430	3.780	0.694	0.330	0.000	0.010	0.104	-0.02	0.005	0.193	0.034	0.008
t_{27}	1.060	5.23	5.600	0.078	0.020	0.010	-0.030	0.135	-0.04	0.192	0.193	0.157	0.686	t_{27}	1.060	5.430	3.790	0.047	0.020	0.000	0.010	0.135	-0.02	0.007	0.193	0.034	0.006
t_{28}	1.280	5.38	5.150	1.096	0.220	0.150	-0.450	0.135	-0.03	0.190	0.192	0.167	0.714	t_{28}	1.280	5.430	3.800	0.462	0.220	0.000	0.010	0.135	-0.01	0.009	0.192	0.034	0.006
t_{29}	1.290	5.38	5.140	0.030	0.010	0.000	-0.010	0.153	-0.02	0.144	0.194	0.167	0.714	t_{29}	1.290	5.380	3.810	0.109	0.010	-0.05	0.010	0.153	-0.01	0.009	0.194	0.036	0.005
t_{30}	1.320	5.37	5.100	0.107	0.030	-0.01	-0.040	0.149	-0.02	0.143	0.132	0.164	0.716	t_{30}	1.320	5.370	3.820	0.070	0.030	-0.01	0.010	0.149	-0.02	0.010	0.132	0.016	0.003
t_{31}	1.690	5.19	4.460	1.595	0.370	-0.18	-0.640	0.096	-0.01	0.137	0.153	0.173	0.757	t_{31}	1.690	5.360	3.830	0.776	0.370	-0.01	0.010	0.096	-0.01	0.009	0.153	0.016	0.000
t_{32}	1.690	5.19	4.450	0.021	0.000	0.000	-0.010	0.133	-0.02	0.073	0.154	0.173	0.757	t_{32}	1.690	5.190	4.450	1.348	0.000	-0.17	0.620	0.133	-0.01	0.010	0.154	0.054	0.193
t_{33}	1.710	5.18	4.430	0.063	0.020	-0.01	-0.020	0.131	-0.02	0.071	0.154	0.173	0.757	t_{33}	1.710	5.180	4.430	0.063	0.020	-0.01	-0.02	0.131	-0.02	0.071	0.154	0.053	0.194

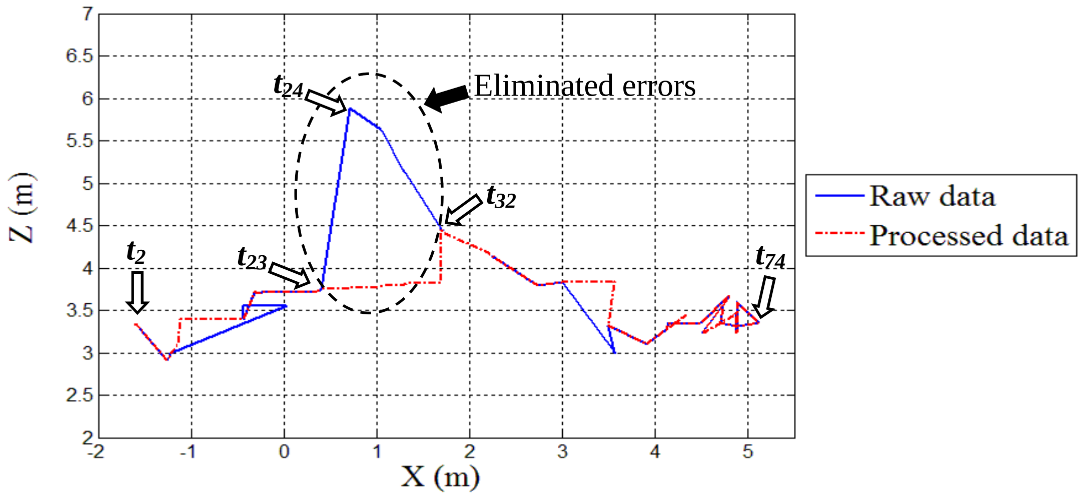


Figure 5-26: Comparison of traces of Tag_3^2 in X-Z plane before and after correction

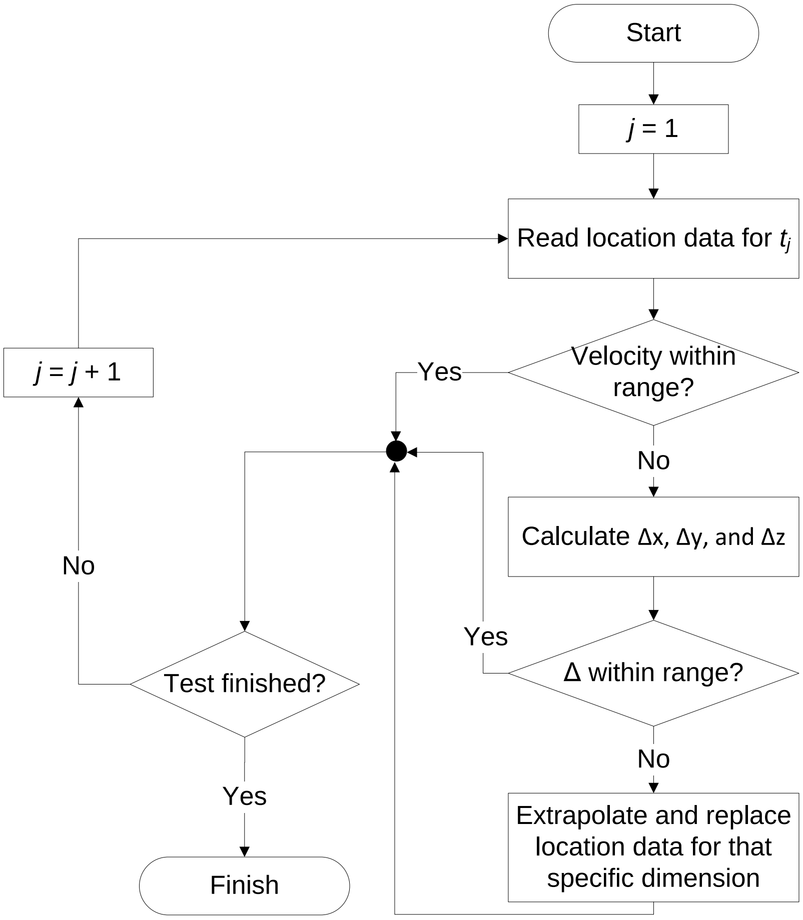


Figure 5-27: Flowchart of near real-time data processing for single tags

It should be clarified that by chance the movement of Tag_3^2 during the period between t_{24} and t_{32} is almost parallel to the X axis and to the X-Y plane, as can be seen in Figure 5-31; therefore, after correction, the Δ_y and Δ_z values are close to 0. A flowchart is shown in Figure 5-27 to summarize the near real-time data processing for single tags.

Example of calculating missing data based on geometric constraints

The same procedure is applied to Tag_1^2 as shown in Figure 5-28. However, in some cases, missing data occur more than two consecutive times because of radio interference, for example, between t_{41} and t_{57}, as shown in Figure 5-28. In these cases, repeating extrapolation according to the history of the tag itself may increase the error, which could be detected by checking geometric constraints. As described in Step 5 in Section 5.3, multiple tags are used to filter errors and fill in the missing data based on geometric constraints of the object. The distance between Tag_1^2 and Tag_3^2 in each time period t is calculated to check if it is within the range of [$D_{ii'}$ - 2ε, $D_{ii'}$ + $2\ \varepsilon$], where $D_{ii'}$ is 1.6 m and ε is 30 cm, resulting in a range of [1.0 m, 2.2 m]. This step has been applied starting from t_{42}, where extrapolation is applied four times in a row to fill in the missing data of Tag_1^2. However, at t_{46}, the distance between Tag_1^2 and Tag_3^2 is 2.44 m, which is out of range. Therefore, the location of Tag_1^2 calculated according to extrapolation is not acceptable. In this case, according to Step 6, the data of Tag_1^3 and Tag_1^1 are used to calculate the missing data of Tag_1^2 between t_{46} and t_{57} based on the known distances from Tag_1^1 to Tag_1^2 and from Tag_1^2 to Tag_1^3 (3.9 m and 8.4 m, respectively), as shown in Figure 5-31. Figure 5-28 shows the extrapolation based on the history of Tag_1^2 from t_{42} to

t_{44} and the interpolation based on geometry according to the other two tags (Tag_1^3 and Tag_1^1) from t_{46} to t_{56}. From t_{57} the system is able to capture the data for Tag_1^2 again.

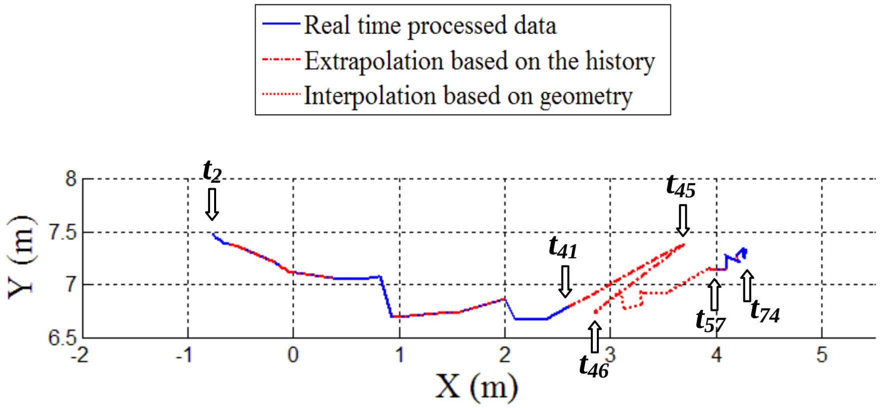


Figure 5-28: Trace of Tag_1^2 based on extrapolation of its history and interpolation of other two tags

Other observations based on data processing

The data of all tags are assumed to be almost synchronized (step 4 in Figure 5-7). However, it should be noticed that in extrapolation based on geometry using two tags on the boom, the small time gaps between different tags can cause problems when the update rate of tags is not high enough. For example, in this test, for tags attached to the upper side of the boom, which are Tag_1^1, Tag_1^2 and Tag_1^3, in each time period *t*, based on the automatic scheduling of the Ubisense system, the data of Tag_1^3, Tag_1^1, and Tag_1^2 were captured in that order with fixed time difference of 119 ms and 74 ms, respectively.

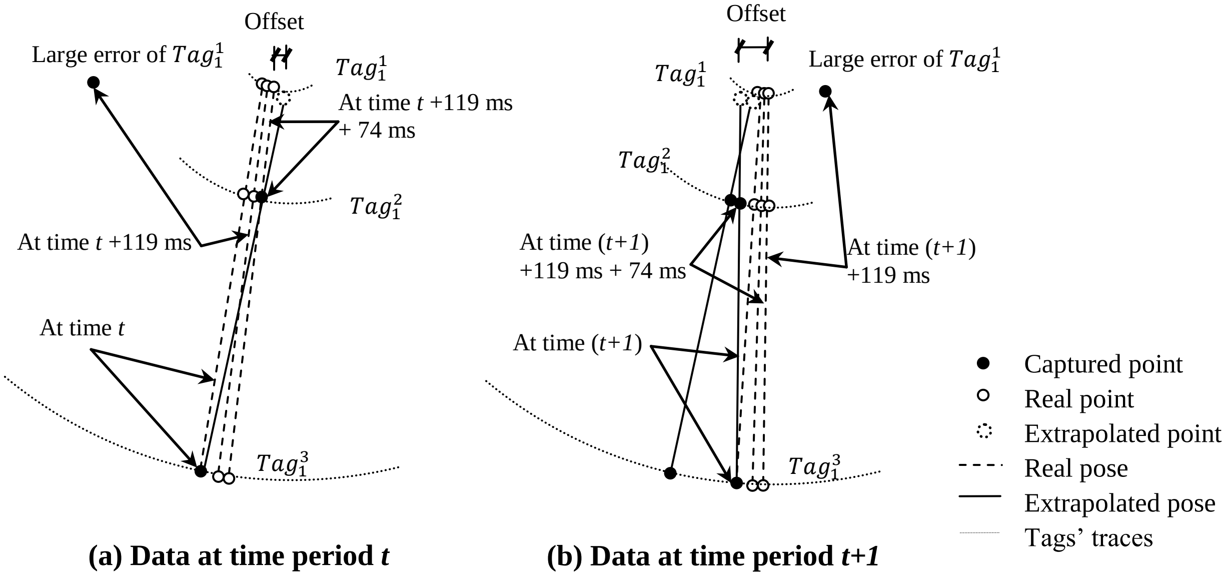
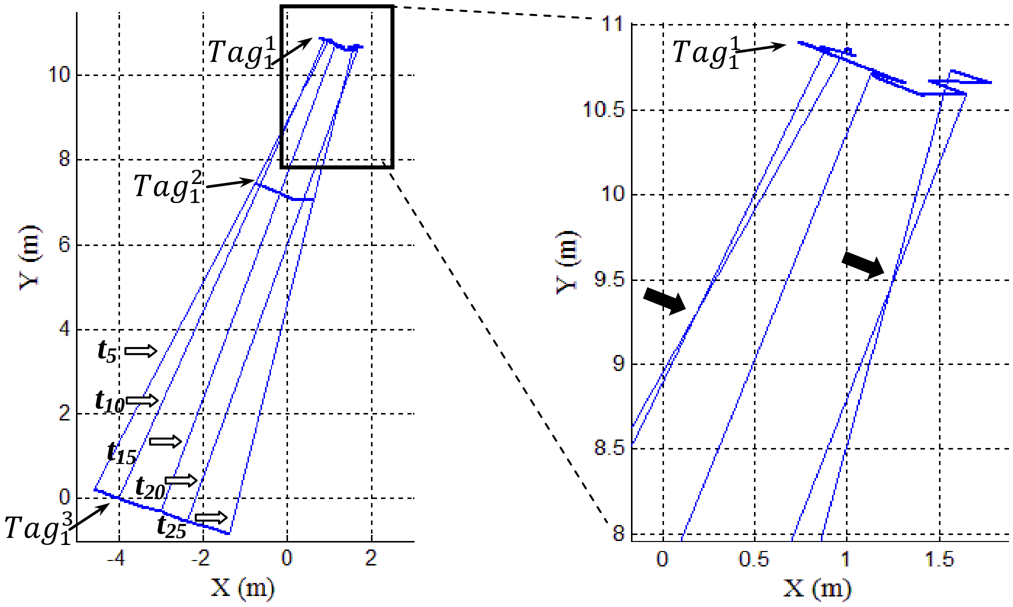


Figure 5-29: Conceptual figure of extrapolation errors



(a) **(b)**

Figure 5-30: Data processed in real time showing the traces of three tags at different time

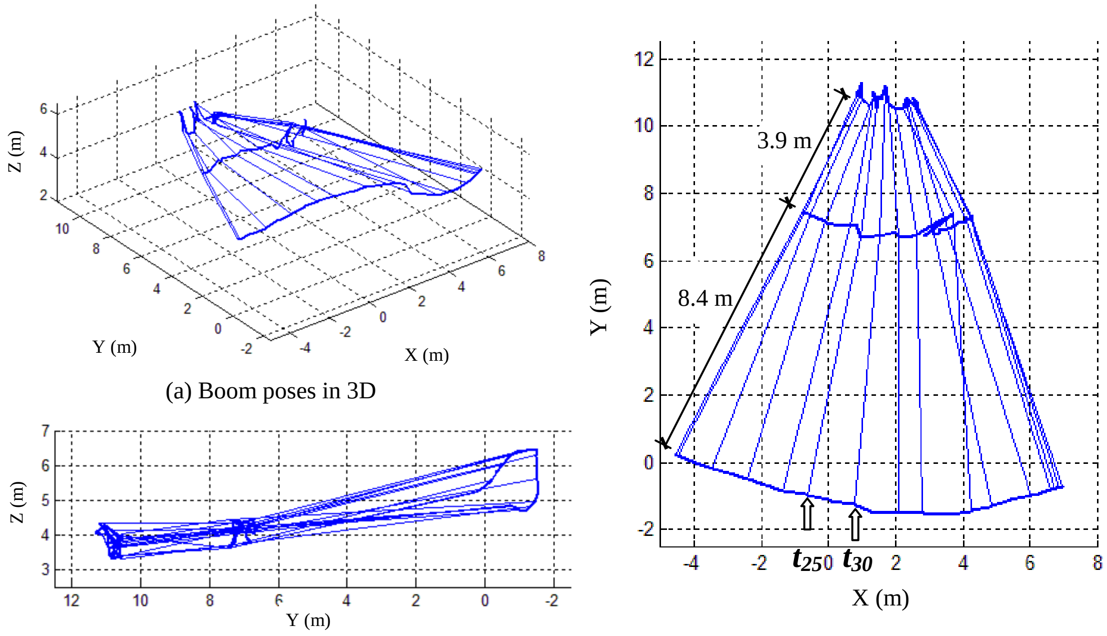
As shown in Figure 5-29(a), a point with a large error was captured for Tag_1^1 at time t+119 ms; therefore, extrapolation based on Tag_1^2 and Tag_1^3 is applied to calculate the position of Tag_1^1. The black circles are the location data captured by the system, whereas the solid white circles are the real locations of the tags at specific times, and the dotted circles are the ones calculated according to extrapolation. This figure also shows the traces of Tag_1^1, Tag_1^2 and Tag_1^3 and the boom poses based on extrapolation as explained above. Notice that we are ignoring the accuracy errors for Tag_1^3 and Tag_1^2 in Figure 5-29(a) and only for Tag_1^3 in Figure 5-29(b). However, because of the small time gap and the relatively big distance between these three tags (around 12.3 m between Tag_1^1 and Tag_1^3, as shown in Figure 5-31) during the lifting task, a big offset of the location of Tag_1^1 is expected when applying extrapolation. Moreover, due to the static information filter of the Ubisense system used in this test (with Gaussian noise on position), small movements of a tag are ignored when predicting the next location of the tag. This filtering results in a cluster of almost overlapping points. Using these data for extrapolation may cause a backward movement of Tag_1^1, as shown in Figure 5-29(b).

As an illustration of this problem, the trace for Tag_1^1 is shown in Figure 5-30, which gives the data processed in real time. Figure 5-30(b) focuses on the zigzag shape of the trace and the crossing of the boom poses at times t_5 and t_{10}, and at times t_{20} and t_{25}. Based on this observation, the continuous extrapolation for Tag_1^1 based on the other two tags may increase errors.

5.5.4 CALCULATING THE POSES OF THE BOOM

As described in Step 7 in Section 5.3, averaging the data of multiple tags in the same cross section should be applied to get the center points of these sections, thereby defining the axis of the boom. A bounding shape (e.g., a cylinder) to cover these three points at each time period can be created with a suitable buffer according to the cross section dimensions of the boom.

This method assumes that the quality of the data of each tag is equal; however, based on the actual collected data, the method of calculating the poses of the boom should be adapted so as to preserve the data of high quality. Based on our observation, tags on the top side of the boom have better quality; therefore, the traces of these tags (Tag_1^i) are used to create the poses of the boom. As shown in Figure 5-31, the three traces show the poses of the boom at different times.



(a) Boom poses in 3D

(b) Boom poses in Y-Z plane

(c) Boom poses in X-Y plane

Figure 5-31: Boom poses at different time periods

5.5.5 CALCULATING THE POSES OF THE BOOM USING POST PROCESSING

Although near real-time data processing is applied to remove errors and fill in missing data, in order to investigate the quality of the UWB data, post processing has been applied to take advantage of the whole data set where interpolation can be used to produce data with better quality. Figure 5-32 shows an example of correcting the errors of Tag_1^1, which has more errors because it is attached at the base of the boom where there are more radio reflection problems. In the X-Y plane, the trace of Tag_1^1 should follow a curve according to the movement of the boom; however, the raw data have relatively large errors. An improvement can be seen after removing errors by applying interpolation based on history and extrapolation based on Tag_1^2 and Tag_1^3. In addition, the trace of Tag_1^2 is post processed to remove the jaggedness due to the missing data. Although this post processing of Tag_1^2 is unnecessary because the pose of the boom can be directly drawn according to Tag_1^1 and Tag_1^3, the purpose of applying post processing is to compare the poses based on near real-time processing and post processing. Figure 5-33 shows the traces and the boom poses after post processing.

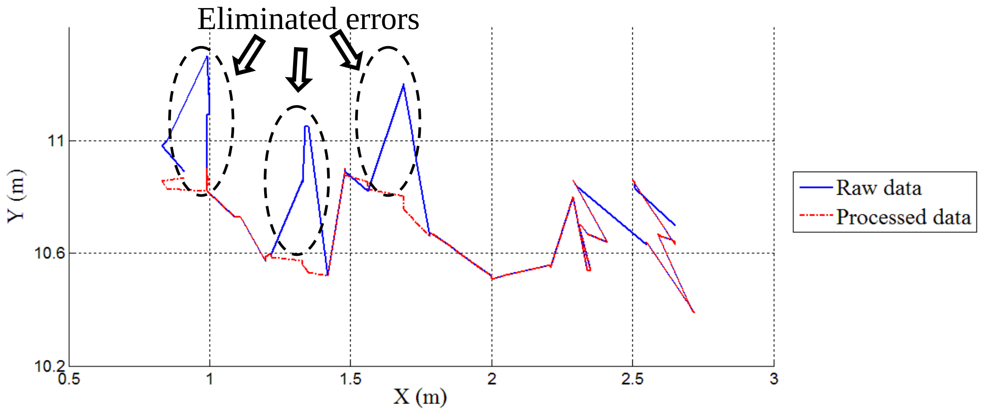
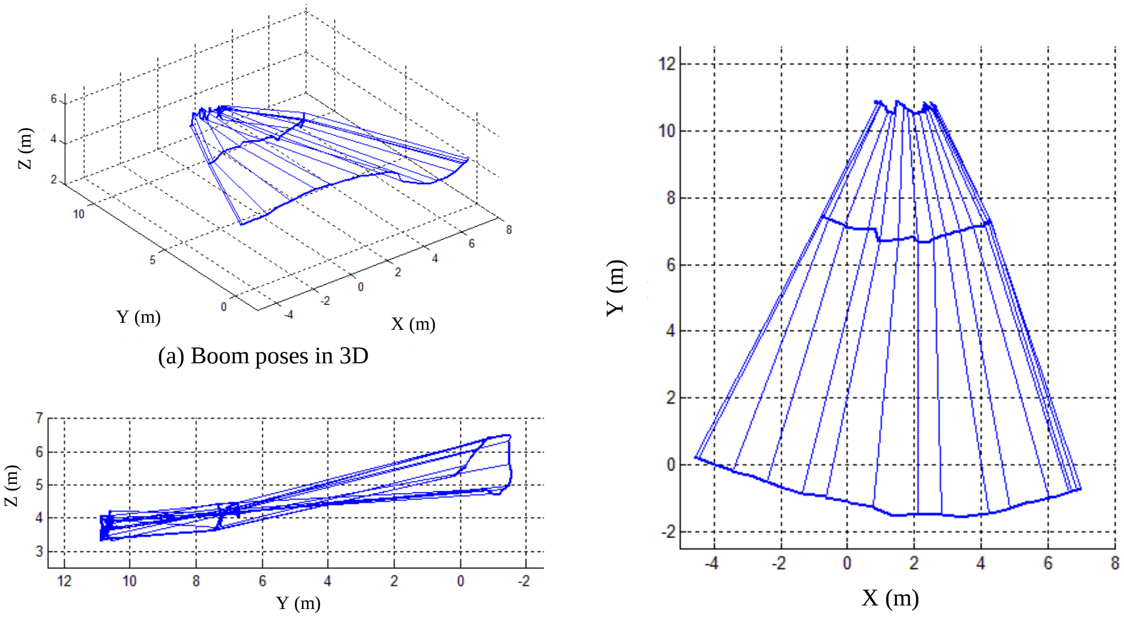


Figure 5-32: Comparison of traces of $\mathbf{Tag_1^1}$ in X-Y plane before and after correction using post processing



(a) Boom poses in 3D

(b) Boom poses in Y-Z plane

(c) Boom poses in X-Y plane

Figure 5-33: Boom poses at different time periods after post processing

5.6 SUMMARY AND CONCLUSIONS

The present chapter has discussed the use of UWB RTLS in construction sites to calculate crane boom poses in near real time. System settings for satisfying the requirements defined in Section 3.3 have been discussed in detail. Indoor and outdoor tests were undertaken to gain experience and to evaluate system performance. Data collected from an outdoor crane test were analyzed in detail to investigate how to improve the UWB system's usability when applied in construction, especially to improve crane safety.

Our observations from the UWB tests are the following: (1) The number of tags in the monitored area should be kept smaller than the maximum number given in inequality (3-4); otherwise, not only more sensors, but also more cells should be used to achieve a better update rate by dividing the monitored area into smaller areas sensed by different groups of sensors. (2) Regarding the visibility, tags should be attached to the upper and bottom sides of the boom to obtain a better visibility and better data quality. Attaching one tag to the hook is enough. More tags should be attached to the lift object to avoid obstruction of radio signals, and it is better to attach the tags to the top surface of the lift object. Tags should be attached to the top of the operator’s cab to achieve better visibility. (3) Better visibility and an adequate number of tags result in better accuracy. Furthermore, appropriate filtering improves accuracy as can be seen from the results of the indoor test and the second outdoor test.

The results of the tests showed a good potential for the use of UWB RTLS on construction sites to reduce safety risks. However, some limitations exist. For example, near real-time data processing has a limitation when applying linear interpolation and extrapolation based on two points only, which may not fit the accuracy requirement. Future improvement can be achieved by using curve fitting or other methods while taking more points into account. The filter embedded in the system is not easy to control, and a Kalman filter combined with geometric constraints (Arras et al., 2003) is a better solution.

To conclude, our contributions related to using UWB RTLS in construction are the following: (1) The setting method of the UWB system to satisfy the requirements of accuracy, visibility, scalability and real-time by deciding the locations and orientations of sensors, and the number and locations of tags; (2) Defining a method of location data processing to improve data quality by filtering errors based on heuristic rules, by filling in missing data based on historic data, and by applying geometric constraints; and (3) Extensive testing of the UWB system in the context of crane safety where it was demonstrated that the poses of the crane can be calculated in near real time based on the proposed data processing method.

CHAPTER 6 PROTOTYPE SYSTEM DEVELOPMENT AND INTEGRATED TESTING

6.1 INTRODUCTION

Although the implementation of a fully integrated system is beyond the scope of the present research, a prototype system is developed to test the proposed approach. The prototype system integrates the motion planning and re-planning algorithms with the UWB system and some basic functions of the agent system. Autodesk Softimage is used to take advantage of its 3D visualization and animation capabilities. Motion Strategy Library (2003), which includes variations of RRT algorithms, is used as a base library for developing an integrated motion planning solution in Softimage. Scaled crane models with attached tags are controlled wirelessly by executing the generated motion plans. Three integrated tests are undertaken to investigate the applicability and the accuracy of the prototype system.

6.2 SELECTION OF DEVELOPMENT TOOLS

Several development tools are reviewed and compared based on their capability, compatibility and extensibility. Table 6-1 shows a summary of the functionalities provided by some software tools that can be used in the present research.

MATLAB (Mathworks, 2010) is a powerful interactive environment that enables users to perform computationally intensive tasks. MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, and interfacing with programs in other languages. Several toolboxes are available, such as the RRT toolbox for motion planning algorithms (Paul and Clifton, 2008), the VR toolbox for virtual reality, and the V-Collide toolbox for collision detection. However, the source codes of these tools have been written by different people for different purposes. Integrating these separate pieces into a well-organized system is time consuming. Therefore, MATLAB has been used in the present research as the tool for the preliminary data analysis discussed in Chapter 5.

Table 6-1: Summary of the functionalities of software tools

Software Functionality	MATLAB	Autodesk Softimage	Ubisense Software
Visualization	VR toolbox	Available	Available
Motion planning	RRT toolbox	MSL API	Not available
Collision detection	V-Collide source code	Available	Available
Agent development	Robotics Toolbox	Behavior tool	Not available
Tracking	Not available	Not available	Available

Autodesk Softimage (2010) integrates modeling, animation, simulation, compositing, and rendering into a single, seamless environment. The modeling tools are designed for creating and editing seamless animated models that can be customized by programming. Therefore, Softimage has been selected as the main development tool in this research to take advantage of its modeling and visualization functions. Other software tools are linked with Softimage using its plug-in mechanism. Figure 6-1 shows a 3D simulation environment for multiple cranes developed in Softimage for this research.

Since the RRT algorithm family has been selected in the proposed approach, we have explored some tools for its implementation within Softimage. Motion Strategy Library (MSL) (2003) Application Programming Interface (API) was selected to be integrated into Softimage because it has several variations of RRT including the basic RRT and

RRT-Connect-Connect and a built-in collision detection algorithm (Proximity Query Package (PQP)).

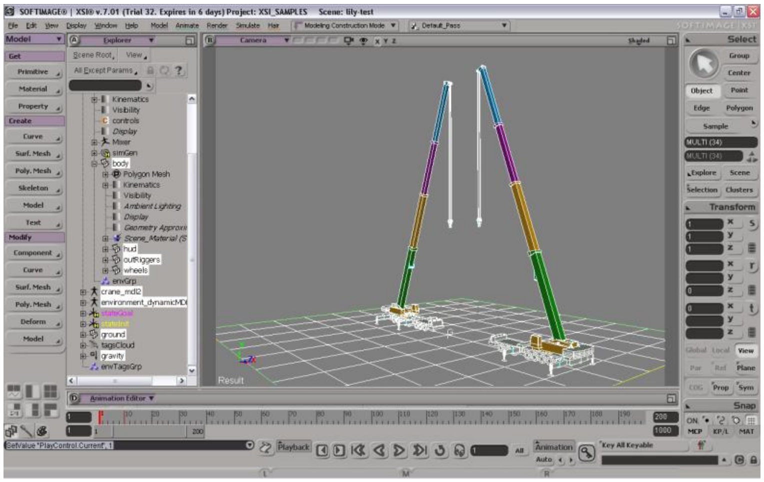


Figure 6-1: 3D simulation environment in Softimage

Ubisense software (2010) has specific functionalities for UWB tracking, visualization, and collision detection. The Ubisense API provides functions for capturing and processing the UWB data, which can be integrated with other software tools.

6.3 PROTOTYPE SYSTEM DESIGN

Three modules have been designed in the prototype system. They are the following:

- ***Visualization module*****:** includes the visualization of the scene and the display of the motions of cranes.
- ***Problem solving module*****:** includes the path planning and re-planning algorithms and the coordination strategies by assigning priorities.
- ***UWB data capturing module*****:** includes the processing of the near real-time data captured from the sensors and the updated environment model.

Figure 6-2 shows the partially integrated prototype system design where it is assumed that two cranes with attached multiple tags are in operation near each other in a construction site. The location data of these two cranes are collected by the UWB sensors and sent to the Ubisense server, which is connected with Softimage using a plug-in, as is discussed in Subsection 6.4.1. A virtual environment scene is created in Softimage to simulate the actual site with all the obstacles and the two cranes. Motion plans are generated for these two cranes by the *Crane Agents* and translated into actions that can be sent to the crane operator using an intuitive Graphic User Interface (GUI). However, the design of the GUI is beyond the scope of this research. The same actions can be sent directly to the cranes using an autonomous control that is similar to the auto pilot mode of an airplane where the operator can intervene and take charge of operating the crane if necessary. Once the two cranes start executing their tasks, their actual locations are captured by using the UWB system and are sent to Softimage in near real time to update the virtual crane location in the scene. Each movement in the actual environment is reflected in the virtual environment, and collision detection is applied for the next movement for each crane. In the current prototype system, it is assumed that the only change in the actual environment is the movement of the booms of the cranes. However, other changes can be captured in the future using laser scanners or other technologies. Once a potential collision is detected by the two *Crane Agents*, the *Coordinator Agent* gives the priority to one of them based on the priority patterns discussed in Section 3.5 to continue its planned path, whereas the other crane has to re-plan its path to avoid collision. The communication channels are open between the *Crane Agents* and the *Coordinator Agent* only, enabling the reporting of potential collisions by the *Crane*

Agents and deciding the priorities by the *Coordinator Agent*. This prototype system can be further extended in the future in a distributed manner using a specialized agent environment to enhance the multi-agent communication and negotiation functions.

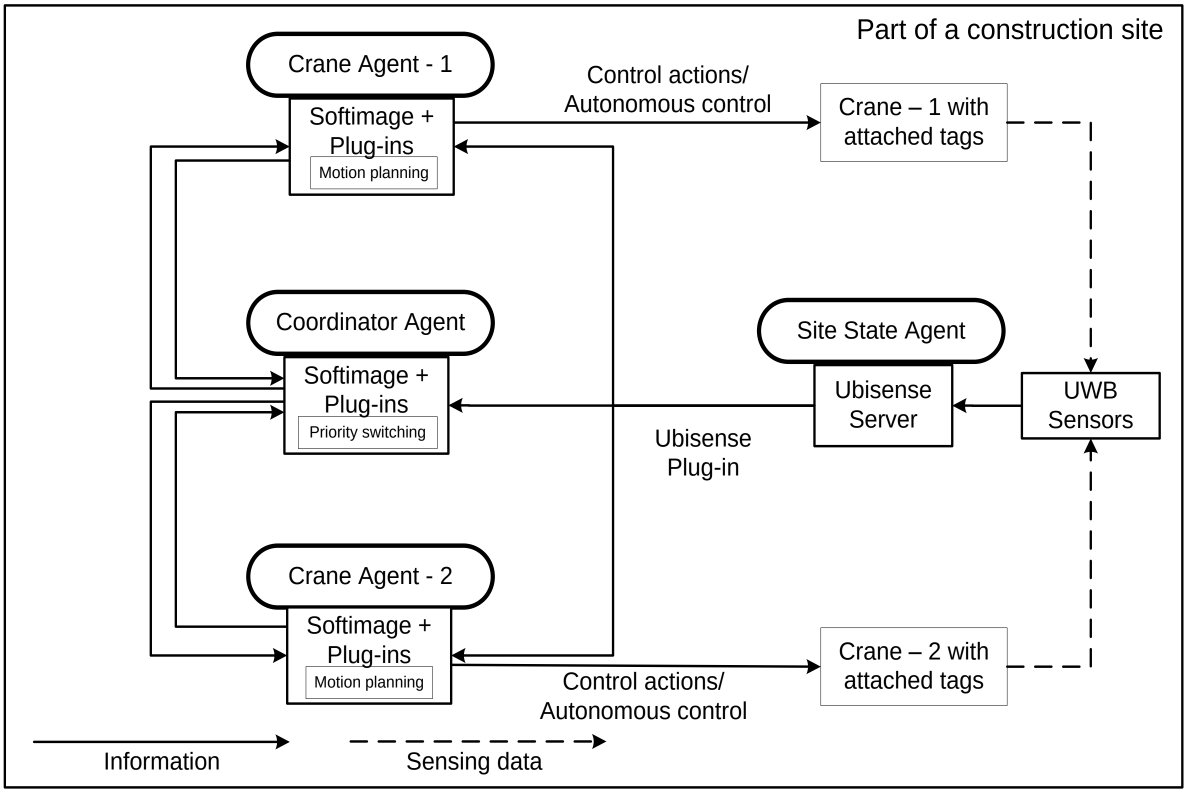


Figure 6-2: Integrated prototype system design

6.4 PROTOTYPE SYSTEM DEVELOPMENT

6.4.1 SOFTWARE COMPONENTS

Visualization module

All software components are integrated into Softimage using its Software Development Kit (SDK) and its C++ API to ensure a seamless integration that takes full advantage of its 3D capabilities. The 3D environment is created in scenes, including static and dynamic objects. The simulation of the crane movements is done by defining key frames with configuration parameters resulting from the crane motion plan. The position of each component of the crane in the work space for each key frame is calculated. Scene environment updating is done by reading information from the UWB system and by linking with specific frames in the scene. Two levels of detail are used for the crane for which an accurate and detailed model is used for visualization, and the bounding boxes are used for collision detection to reduce computational time. As mentioned before, the boom poses can be calculated in near real time and a buffer is added to the boom for safety purpose taking into consideration the accuracy of the tracking system. The buffered bounding boxes are used for collision detection in the scene.

Problem solving module

MSL (Motion Strategy Library, 2003) is used for solving planning queries in the system. This library provides a wide range of randomized motion planning algorithms including RRTs. This library was modified and extended according to the proposed motion planning algorithms to fit with the system. Modifications were added to update the code to be compatible with the Softimage API. New classes were added for interacting with the data in Softimage. This interaction is required in order to read the motion planning problem directly from the Softimage scene, which includes the kinematic properties and geometrical representation of the cranes in addition to the static and dynamic obstacles. Additional extensions to the library were made to develop the new motion planning and re-planning algorithms discussed in Chapter 4. Along with MSL, the Proximity Query Package (PQP) (PQP, 1999) was used for performing collision detection queries on obstacles found in the environment. Each simulation step is defined as one frame in Softimage, and at every simulation step, the geometry of the scene is accessed to detect collisions. It is assumed that each second of the simulation is composed of 32 frames. This assumption provides enough accuracy for the PQP library to detect dynamic obstacles in construction sites.

UWB data capturing module

Ubisense software is used as the platform of the near real-time location system. A plug-in of Ubisense is developed to transfer data into Softimage. This allows Softimage to read near real-time location data from the UWB system and to show the traces of the tags that are attached to the physical cranes for updating the location of the virtual crane in the virtual scene. Due to different coordinate systems defined in Ubisense and Softimage, the following coordinate mapping is performed by the plug-in:

- Softimage X = Ubisense -X
- Softimage Y = Ubisense Z
- Softimage Z = Ubisense Y

Data transferred to Softimage can be recorded by using two methods. The first method is to record the coordinates of selected tags during the recording time period and to show the traces of the tags, which are linked with specific frames in the scene. This information can be saved in the scene and replayed later. The second method is to record the same data and store them in a database so that the data can be easily retrieved and analyzed. MySQL (2010) is used for storing the data from Ubisense in near real time. MySQL is a relational database management system. The data from Ubisense can be saved in this database and imported back from the database to be viewed in Softimage at a later time. The database contains the time when the data are collected, tag ID, X, Y, Z coordinates, and the frame in which the data are displayed in Softimage when imported from the database.

Once installed, the plug-in creates a *UbisenseConnector* menu in Softimage. From this menu, all the features of this plug-in can be controlled and customized according to the needs of the user. Figure 6-3 shows the interface of the plug-in in Softimage.

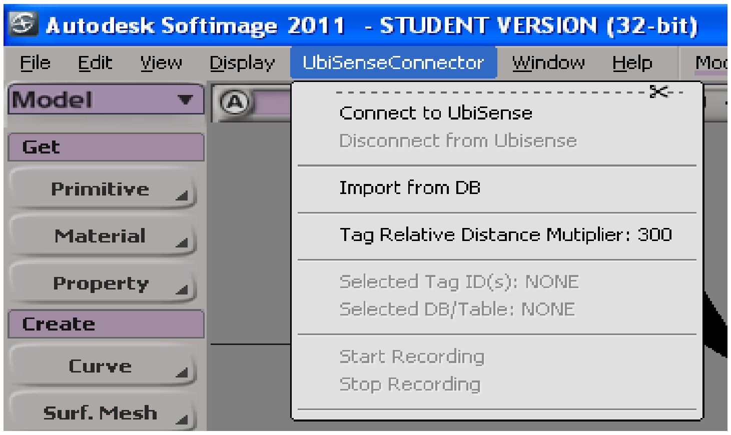


Figure 6-3: Interface of Ubisense plug-in

The functions of this plug-in are as follows: (1) connecting to Ubisense, (2) selecting tags to be rendered in the scene by choosing the tag IDs from a tag list, (3) creating/selecting a database table if the data need to be saved in the database, (4) linking the selected tags with the virtual tags in Softimage, (5) recording data and displaying the traces, and (6) importing data from the MySQL database when needed.

6.4.2 HARDWARE COMPONENTS

The decision to carry out the integrated tests in the laboratory was taken because carrying out the same tests in a full-scale real construction site (or even simulated site) has the following limitations:

(1) The cost of renting cranes with operators is high. The setting of each test requires several hours mainly for the installation and calibration of the UWB system. Furthermore, because of the complex nature of the system, several unpredictable

problems may occur during the setting of the test. This complexity makes the outdoor test more difficult especially in the case of bad weather. It should be noted that it is possible to protect the UWB sensors and tags from harsh weather and site conditions using special casing.

(2) The lab testing environment can be fully controlled and can be repeated as many times as necessary.

As a consequence of the reasons given above, two radio-controlled (RC), scaled (1:18) hydraulic crane models were used in the integrated tests (Hobby Engine, 2010). Each crane has six motors that allow the movement of the body of the crane (drive forward/backward, turn right/left), of the boom (swing right/left, turn up/down, extend/retract), and of the hook (move up/down). Table 6-2 shows the DoFs, ranges, and speeds of the boom movements. A crane can be manually controlled using a remote control with different buttons and joysticks that allow the movement of one DoF at a time.

The radio frequency between the remote controllers and the receivers of the cranes is 27 M.Hz. All the scaled cranes within the range of the remote controller (at about 10 m) receive and execute the same commands because the remote controller does not specify a specific target crane for a specific command. To control multiple cranes in the same area by computer, one remote controller was interfaced with a microcontroller (Phidgets, 2010) connected to the computer with a USB cable. An encoding scheme was implemented allowing for sending commands from the computer equivalent to pushing buttons on the remote control. Furthermore, the receiver circuits of the scaled cranes were modified to react only to commands sent to that specific crane. As a result, it became possible to send a series of commands from the computer that controlled each crane separately using software developed in C++ based on the API library of the microcontroller.

Table 6-2: DoFs, ranges and speeds of boom movements

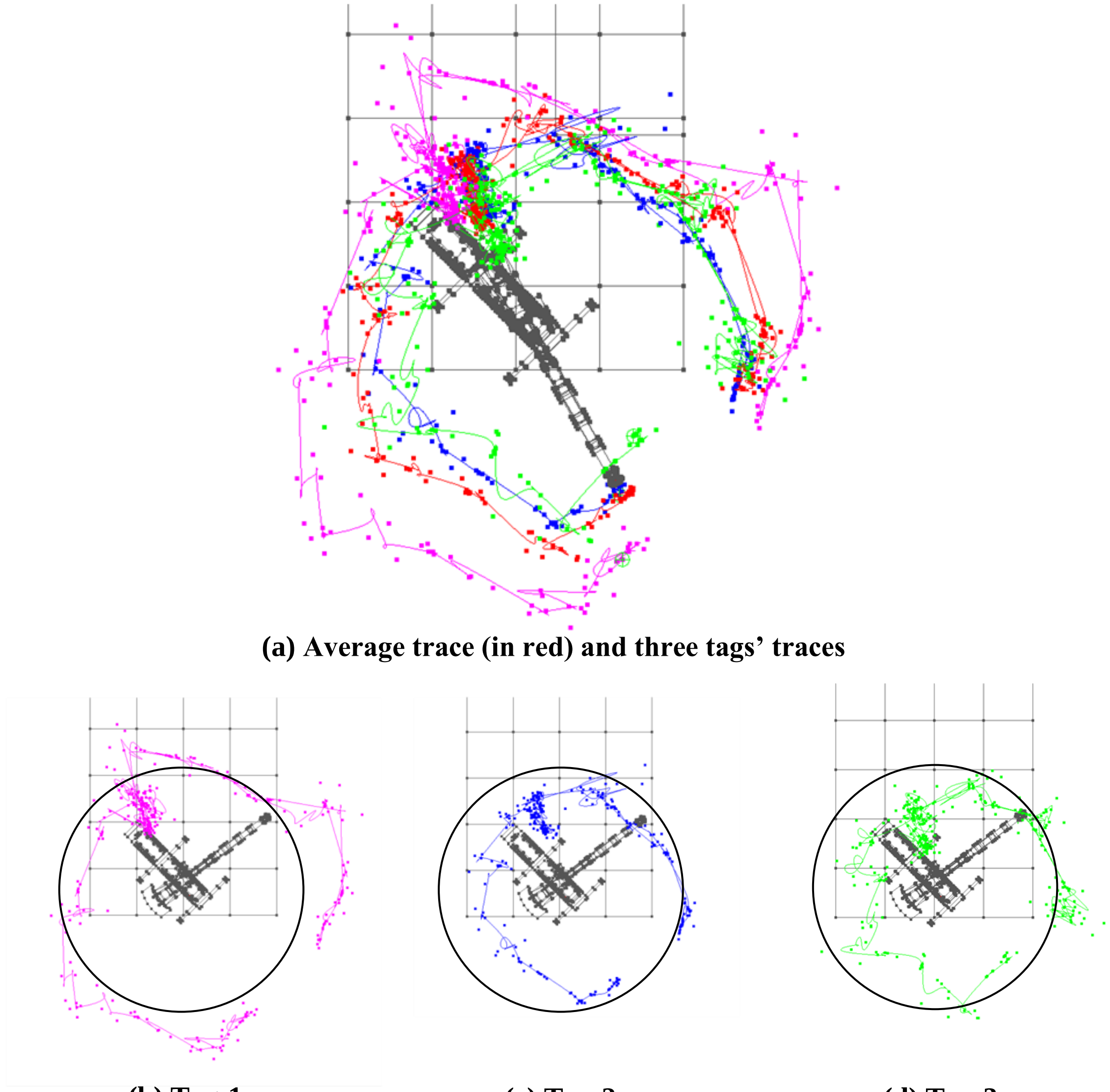
DoF	Range	Unit	Speed
Swing counter/clockwise	[-170, +170]	degree	13.3 °/second
Raise up/down	[0, 40]	degree	8 °/second
Extend/retract	[33, 68.5]	cm	3.94 cm/second

6.5 INTEGRATED TESTING

This partial integrated testing was based on the laboratory prototype system. In Section 4.7, the path planning/re-planning algorithms were tested in the virtual environment. In Section 5.4, the feasibility of using the UWB system for tracking cranes was examined in several tests. These tests were essential to verify and to validate the methods used in the system proposed in Figure 3-1. However, in order to validate the proposed methods at the integrated system level, three laboratory tests were carried out using the RC scaled cranes. The same virtual models of cranes used in the case studies discussed in Section 4.7 were used in the tests after adjusting their scale and locations in the virtual model so that the bases of the booms and their axes match the scaled RC cranes. Compared with the near real-time data processing explained in Chapter 5 where the poses of the boom are calculated according to the multiple tags at different sections, the lab tests were simplified to focus only on the tags attached to the boom tip. The location of the base of the boom is fixed in the virtual scene; consequently, the movement of the boom of the virtual crane was defined by the tags attached to the boom tip. Three tags were attached to the boom tip to improve accuracy and visibility, and the average location of these tags was used to guide the movement of the boom in the virtual scene. Virtual tags were created in Softimage to show the locations of the tags attached to the crane. Furthermore, geometric constraints were used to control the movement of the virtual crane based on the location of the tags attached to the tip of the crane's boom.

Test 1 – Integration of tracking and real-time visualization

The purpose of this test was to verify the performance of the UWB system when integrated with Softimage, so that the monitored movement of the actual crane could be visualized in the virtual environment. The actual crane was moved by raising and extending the boom, then swinging the boom clockwise by 170 ° followed by swinging it counter clockwise by 340 °; consequently, the boom moved on the surface of a cone and its tip moved in a horizontal circle. Three tags (Tag-1, Tag-2, and Tag-3) were used to improve visibility and accuracy, as shown in Figure 6-6(a). About two hundred readings were collected for each tag for the swing motion. The traces of the tags are shown in Figures 6-4(b), (c), and (d), respectively. The circle is the real path of the boom tip. As can be seen in the figures, it was found that most of the points of the trace of Tag-1 were outside the circle whereas most of the points of the traces of Tag-2 and Tag-3 were inside the circle, matching the real locations' relationships between these tags and the boom tip. Location data averaged from these three tags were used to update the position of the virtual boom because the base of the crane was fixed during the test. Figure 6-4(a) shows the average trace in red together with the three tags' traces. Figure 6-5 shows the average trace in top and 3D views.


(a) Average trace (in red) and three tags' traces

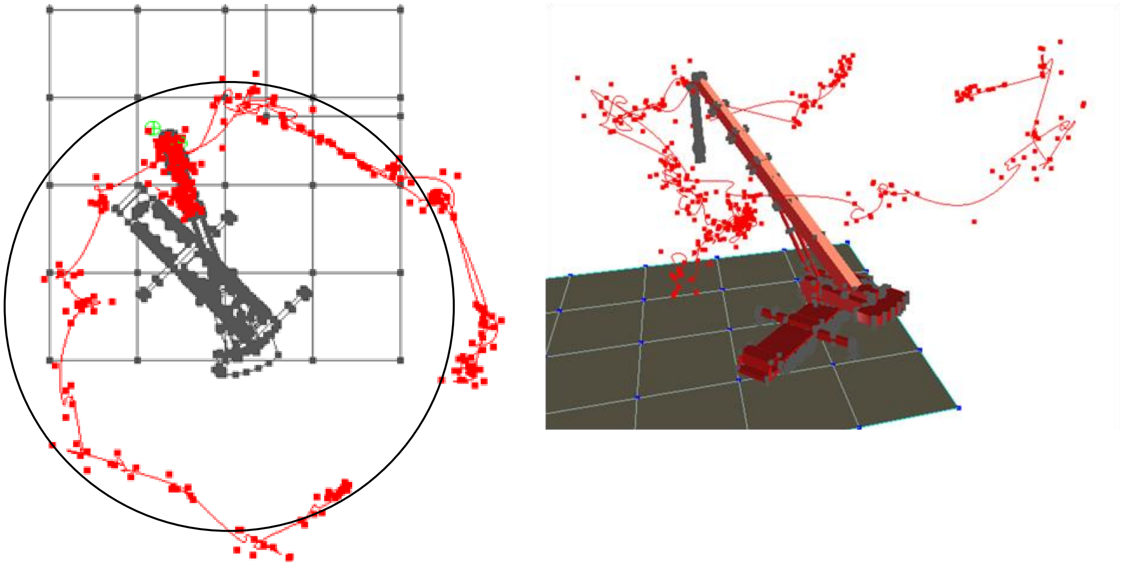
(b) Tag-1

(c) Tag-2

(d) Tag-3

Figure 6-4: Traces of three tags attached to the boom tip

The accuracy of the data is evaluated by measuring the difference between the radius of the real circle path r and the measured radius r' based on the collected data in 2D (X and Y). Adjustments are made according to the tags' locations relative to the boom tip. The accuracy in the Z dimension is evaluated separately. Table 6-3 shows the accuracy analysis results. It can be seen from this table that the averaged location data based on three tags can be used for updating the boom location with a good accuracy.



(a) Top view **(b) 3D view**

Figure 6-5: Boom tip trace based on averaging the locations of three tags

Table 6-3: Accuracy analysis

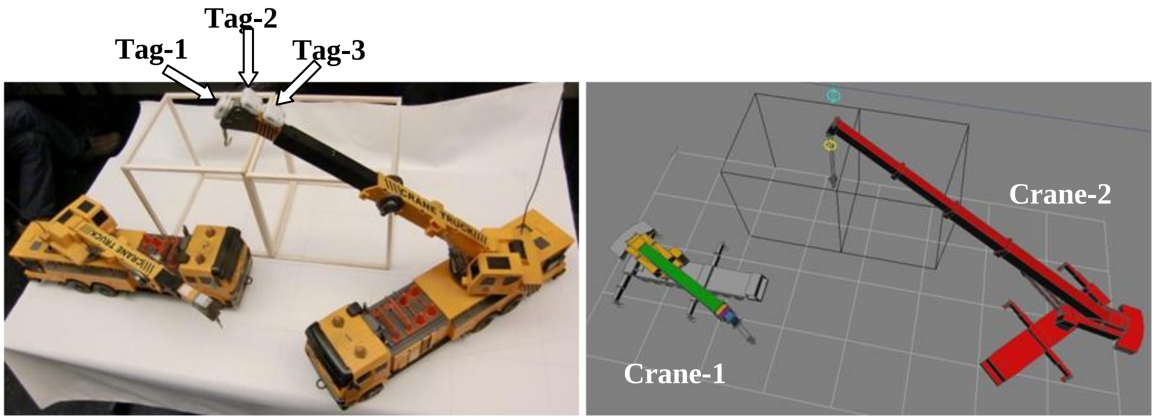
Tag	*Mean radius difference* (cm)	*Standard deviation of radius difference* (cm)	*Mean difference in Z direction* (cm)	*Standard deviation of difference in Z direction* (cm)
Tag-1	3.63	10.04	4.00	9.22
Tag-2	-6.36	10.83	5.18	10.87
Tag-3	-6.72	11.50	6.27	10.87
Ave.	-5.69	10.08	3.31	6.07

Test 2 – Integrated test of tracking and motion re-planning

The purpose of this test is to evaluate the integration of the UWB tracking and the motion re-planning. Figure 6-6(a) shows a picture of the two scaled crane models, with UWB tags attached to the tip of the boom of Crane-2 and with a simple frame structure representing static obstacles. Figure 6-6(b) shows the virtual models representing the cranes and the frame structure. It is assumed that Crane-2 has a higher priority than

Crane-1 based on the safety, task, cost, or time factors related to the tasks that are executed, as explained in Section 3.5. The location data of the UWB tags attached to Crane-2 were used to update its pose in the virtual model, which was used in the motion re-planning of Crane-1 following the methods discussed in Chapter 4.

The scaled model Crane-2 was controlled by using the remote controller to swing the boom in a way that blocks the movement of Crane-1. In the virtual scene, the movements of the boom of Crane-2 followed the physical scaled crane and a potential collision was detected by the agent of Crane-1. Then, motion re-planning was triggered and Crane-1 followed the new path to avoid potential collision with Crane-2. The test successfully demonstrated the applicability of the proposed methods for tracking and motion re-planning at the level of the integrated system.



(a) Scaled cranes (b) Virtual cranes

Figure 6-6: The scaled cranes and their virtual models

Test 3 – Integrated test of motion planning and agents

The purpose of this test was to demonstrate the role of the *Coordinator Agent* in managing the priorities of two cranes operating in the same area as discussed in Subsection 3.5.3. In this test, multiple lifting tasks were executed by each crane. These multiple tasks collectively represented a macro task in the project schedule. In the present test, it was assumed that the two cranes were erecting different elements of a steel structure as shown in Figure 6-7. The tasks executed by Crane-2 consisted of lifting the columns and beams of one part of the structure (the green part). The tasks executed by Crane-1 consisted of lifting the columns and beams of another part of the structure (the magenta part). The pyramid shape represents the picking area of the steel elements. Inverse kinematics is used to define the initial configurations and the goal configurations of the cranes based on the tasks. The motion planning algorithm proposed in Section 4.6 was used to find collision-free and time-efficient paths.

In this test, two scenarios were simulated. In the first scenario, it was assumed that the macro task of Crane-2 was on the critical path of the project; consequently, the *Coordinator Agent* gave the priority to Crane-2 to guarantee the project was not delayed. If a potential collision was detected, the high-priority crane (Crane-2) was considered as an obstacle for the low-priority crane (Crane-1) when Crane-1 re-planned its path. In the second scenario, it was assumed that both cranes had equal priority; consequently, the *Coordinator Agent* decided to alternate the priorities between the two cranes each time a collision was detected. Both scenarios were successfully tested, demonstrating the feasibility of changing priorities by the *Coordinator Agent*.

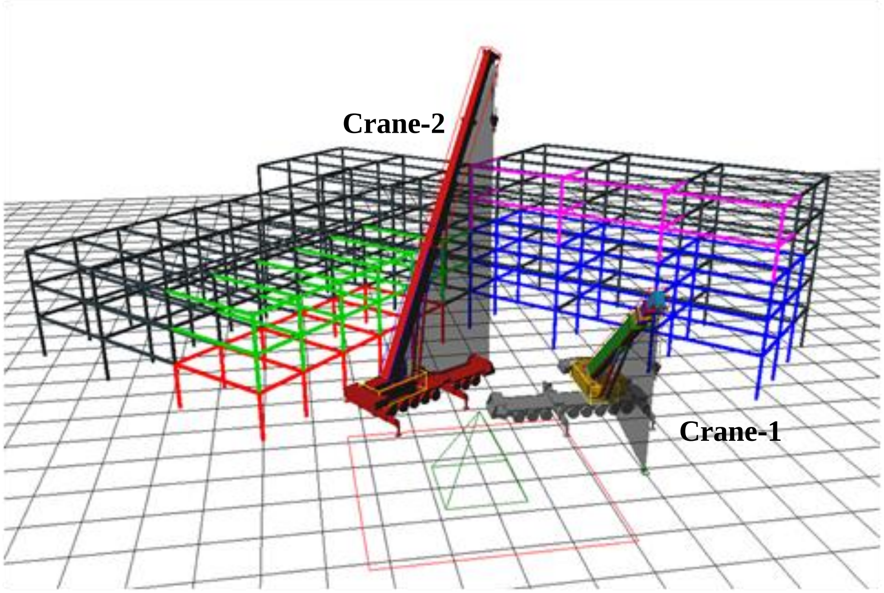


Figure 6-7: Two cranes operating in the same area

6.6 SUMMARY AND CONCLUSIONS

The present chapter describes the development of a prototype system and the integrated testing of the proposed approach. The selection of the development tools was based on an investigation of the availability, the integration possibility, and the functionalities of several software systems. Softimage was selected as the main development tool for the present research to take advantage of its modeling and visualization functions. Other software tools were linked with Softimage using its plug-in mechanism, including Motion Strategy Library (MSL) API and Ubisense API. Three modules were designed for the prototype system including visualization, problem solving, and UWB data capturing. The software components and the hardware components of the prototype system have been discussed in detail. Several partially integrated tests have been carried out by using the RC scaled cranes to demonstrate and to validate the proposed methods at the integrated system level.

CHAPTER 7 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

7.1 SUMMARY

Limited research has focused on providing real-time motion re-planning for cranes while taking into account dynamic environment changes. The present research has proposed an innovative approach to integrate UWB tracking technology and advanced motion planning and re-planning algorithms for crane operations. The resulting integrated agent-based system can improve the safety of crane operations by providing better understanding and near real-time monitoring on construction sites.

7.2 CONCLUSIONS

The conclusions of this research are as the following:

(1) The framework of an agent-based system has been proposed for supporting crane operations to ensure safety. It provides a platform to integrate different modules for tracking, problem solving, communication, and visualization. This framework has several agents that supporting the crane operation, including *Crane Agents*, *Coordinator Agent*, and *Site State Agent*. The main characteristics of the agent-based system have been described as follows: (1) A hybrid approach has been used in the system to gain the flexibility of distributing motion planning to each *Crane Agent* based on the priorities decided by the *Coordinator Agent*; (2) Priority patterns have been defined to decide which agent should re-plan the equipment's path to avoid

potential collisions; and (3) In order to guide the crane operators, motion plans can be translated into actions based on the crane's configurations. Communication between agents enhances environment awareness and improves efficiency by distributed decision-making.

(2) A modified motion planning algorithm RRT-Con-Con-Mod has been proposed for crane operation. The main charastistics of this algorithm are the following: (1) It is a dual-tree RRT algorithm, which generates two trees from the initial configuration and the goal configuration; (2) A cost function is used to evaluate the quality of the path by taking into account the smoothness of the path and the time taken to execute the path; and (3) Engineering constraints are considered to generate safe paths for the cranes and to avoid tip over due to overloading. By using an anytime algorithm, improvement of the path smoothness and reduction in execution time has been obtained. An average improvement of 11.51% better smoothness has been obtained compared with the paths found by using RRT-Con-Con. The calculation time of RRT-Con-Con-Mod is much less than that of the RRT algorithm. The best path found using the anytime algorithm shows an improvement in smoothness and execution time of 21.59% and 16.86%, respectively. The cost of the path is consequently reduced by 18.32%.

(3) A new dynamic motion re-planning algorithm DRRT-Con-Con-Mod has been proposed to efficiently repair the path when the environment is updated. Compared with the DRRT algorithm proposed by Furguson (2006), the proposed dynamic algorithm maintains a focus on the path instead of the entire tree by regenerating a partial path to replace the part that is not collision-free due to new obstacles. The

advantage of the proposed algorithm is that the time for trimming the entire tree is eliminated and an immediate collision-free movement is ensured for the crane, thereby reducing safety risks. Regenerating the trees and finding a feasible partial plan are carried out quickly thanks to the greediness of the dual-tree structure and *Connect-Mod* function. A reduction in re-planning time by a factor of four has been achieved by using the proposed dynamic algorithm. Furthermore, the DRRT-Con-Con-Mod algorithm is always able to find a new path during re-planning.

(4) The requirements of using real-time data collection in construction to improve crane safety using UWB technology have been analyzed. The requirements include accuracy, visibility, scalability and real-time requirements, tag form and function requirements, power, and networking requirements. The setting method of the system has been discussed in detail to satisfy these requirements by deciding the locations and orientations of sensors, and the number and locations of tags. Heuristic rules have been proposed to balance these requirements by clarifying the relationship between the number of tags, the update rate, and the velocity of objects.

(5) A method of location data processing has been proposed to improve data quality by filtering errors using heuristic rules, by filling in missing data based on historic data, and by applying geometric constraints. Extensive testing of the UWB system in the context of crane safety has been carried out where it has been demonstrated that the poses of the crane could be calculated in near real time based on the proposed data processing method.

(6) A prototype system has been developed to integrate the motion planning and re-planning algorithms with the UWB system and to test the proposed approach.

Autodesk Softimage has been used to take advantage of its 3D visualization and animation capabilities. Motion Strategy Library API and Ubisense API have been linked with Softimage using its plug-in mechanism. Three modules have been designed for the prototype system including visualization, problem solving, and UWB data capturing. The software components and the hardware components of the prototype system have been discussed in detail. Several partially integrated tests have been carried out by using the RC scaled cranes to demonstrate and validate the proposed methods at the integrated system level. The simulation of motion planning has enhanced the understanding of the tasks and identified the potential bottle-necks or conflicts in advance, while the near real-time data visualization has greatly improved the monitoring quality and ensured safety. Assisted by the intelligent agents, solving conflicts has been faster by comparison with the conventional methods.

The results of crane motion planning and re-planning, and the near real-time data processing of the location data discussed in the present research have been presented to engineers and experts from crane companies and a construction safety organiztion (CSST). These engineers and experts have provided us with a positive evaluation for applications of the present research in pratice.

7.3 LIMITATIONS AND FUTURE WORK

Although the objectives of the present research have been successfully achieved, the following improvements and extensions have been considered for future research:

(1) The proposed framework has not been fully implemented. The roles and the relationships of agents are defined in the present research; however, the framework

needs to be refined by investigating the details of agent communication and negotiation, and by constructing the agent system using suitable development tools. We have explored some tools, such as Jadex (2008), which is a Java-based, FIPA (Foundation for Intelligent Physical Agents) (2010) compliant agent environment. It allows the development of goal-oriented agents. The main concepts of Jadex are beliefs, goals and plans, which are defined by the programmer and prescribe the behaviour of the agents. Other tools for agent system design can be investigated for easy integration with Softimage, such as tools written in C++. Furthermore, a user-friendly GUI needs to be developed to support the crane operator without disturbing his/her concentration on the tasks. Rules from safety codes need to be included in the knowledge base of the agents to make the system more applicable.

(2) The paths generated by using the proposed motion planning algorithms are not optimal. Further path improvement in the motion planning algorithms can be considered. One option is that instead of choosing only the nearest node on the tree to be connected to the sampled node, the *k* nearest nodes on the tree can be selected and compared based on heuristics, as proposed by Urmson and Simmons (2003). Another improvement could be achieved by using the joint configuration-time state space to model the trajectories of dynamic objects explicitly and to take them into account during planning. Collision avoidance can be carried out by considering the future poses of the dynamic objects in addition to pose detection. Time-optimal or near time-optimal approaches can also be used for computing paths through state-time space.

(3) Linear interpolation and extrapolation are used for data processing in the present research. Further improvement of data processing could be carried out using curve fitting or other methods based on historical data. In addition, Kalman filtering combined with geometric constraints (Arras et al., 2003) could be investigated in the future to improve the accuracy of UWB data.

(4) Building Information Modeling (BIM) could be used to automatically update the construction environment through time and to generate incremental path planning queries for all of the project phases. This updating can be done by allocating each crane to the group of construction elements that it will handle sequentially. Furthermore, by arranging the erection sequence of the elements, risks of collision can be reduced. More research is required in this area to arrange the BIM data and to import it into the proposed system. Additionally, multi-task planning using BIM could be considered to enhance the performance of sequential motion planning queries by caching previous generated RRTs and updating them based on site updates.

7.4 IMPACT ON RESEARCH AND PRACTICE

The proposed approach is expected to have an impact on the construction industry by improving safety and eliminating delays caused by unforeseen spatial problems on the construction site, thereby improving productivity. The intelligence of the multi-agent system can be extended from the re-planning of equipment motions to a more advanced concept, which we call Smart Construction Site (SCS) (Zhang, et al., 2009). Figure 7-1 shows the proposed roadmap towards the SCS based on agent technology, field data capturing technologies, wireless communication, and path re-planning. This roadmap can be considered as an extension of the following concepts and emerging topics used in the Information and Communication Technologies (ICT) Roadmap for Construction (2003), which was proposed by the ROADCON project, focusing on the following new and emerging ICTs: (1) Adaptive and self-configuring systems (early warning/situation tracking), (2) Collaborative virtual teams (smart self-controlling teams, collaborative modeling and visualization), (3) Digital site (site team management tools), and (4) Smart building (long term & real time data). The following paragraph explains the proposed roadmap starting from available technologies that are already in use and from types of take-up technology (the bottom part of the roadmap).

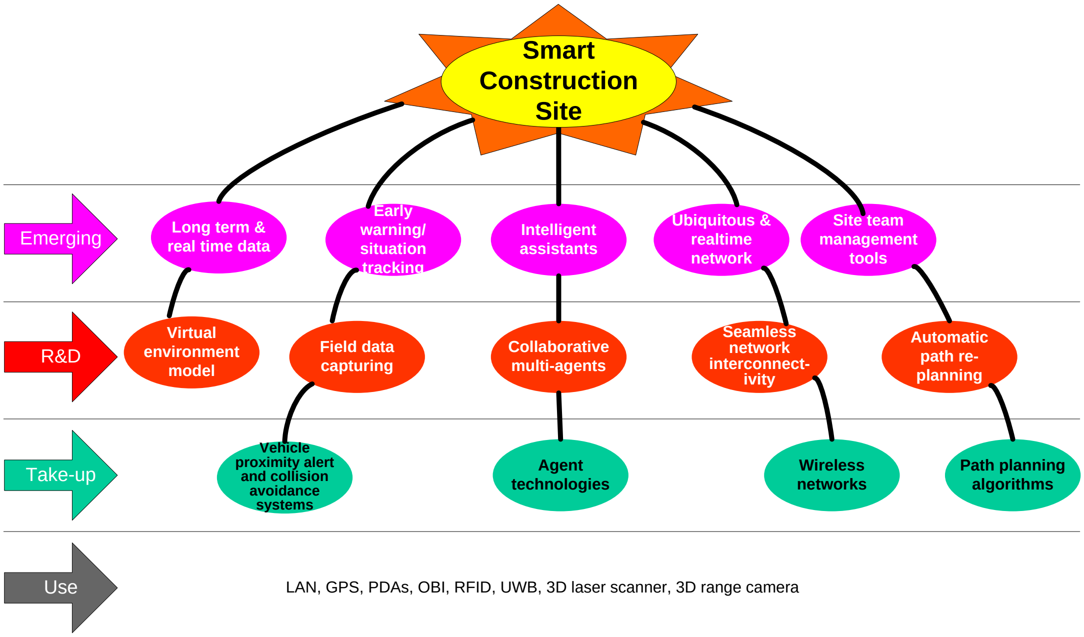

Smart Construction Site
Emerging
Long term & real time data
Early warning/ situation tracking
Intelligent assistants
Ubiquitous & realtime network
Site team management tools
R&D
Virtual environment model
Field data capturing
Collaborative multi-agents
Seamless network interconnect-ivity
Automatic path re-planning
Take-up
Vehicle proximity alert and collision avoidance systems
Agent technologies
Wireless networks
Path planning algorithms
Use
LAN, GPS, PDAs, OBI, RFID, UWB, 3D laser scanner, 3D range camera

Figure 7-1: Roadmap of Smart Construction Site

As discussed in Chapter 2, in the current state of construction projects, GPS is used to monitor the location of equipment. OBI systems are available for heavy construction equipment. Crane path planning software is used in some projects; however, during the execution stage, the tasks of cranes are usually carried out by using a trial-and-error process, based on feedback provided by the operator's own vision and assessment, by using hand signals of a director at the work zone, or by using radio communication. RFID has been proposed to track materials and tools. People communicate with each other using mobile phones or radio terminals. Furthermore, several types of technology are ready for take up, such as vehicle proximity alert and collision avoidance systems, types of agent technology, wireless networks, and path planning algorithms. A virtual environment can be created for simulation and training purposes. Based on these types of available technology, R&D is being undertaken: (1) To capture field data in near real-time and support early warning/situation tracking; (2) To develop collaborative multi-agent systems to provide intelligent assistants; (3) To create a seamless network interconnectivity for collaborative multi-equipment taking advantage of wireless communication; and (4) To develop automatic path re-planning algorithms as an efficient tool for site team management. By integrating all the emerging topics in the roadmap, a vision of SCS can be established where every worker, operator, and staff would have intelligent support from agents encapsulating knowledge and decision-making strategies. Environment information would be obtained and updated by using 3D scanners, range cameras or sensors attached to moving objects on site. Path planning and re-planning of equipment would be carried out automatically to help the operators fulfill their tasks safely and efficiently.

The benefits of a SCS are the following: (1) Safety assurance: Each moving object on site can be monitored and tracked with a precise location, and a warning system can be developed to warn the workers and operators when a potential accident is detected; (2) Productivity control: the tracking records can be used to analyse the workers' and equipment performance and estimate their productivity; (3) Quality control: more awareness of the site situation by tracking different equipment can help the staff make better decisions; and (4) The work process can be easily understood by visualizing the paths of equipment.

An ideal construction site can be described as follows: the construction site is fully modelled and updated in near real-time; construction equipment is fully monitored and controlled with different types of sensors; near real-time location systems are used to track workers, vehicles, materials, etc.; estimation of productivity for equipment and crews is automatically carried out according to sensed data; distributed information is fully integrated and analysed; and intelligent support is provided by near real-time problem solving, such as path re-planning and conflict resolution.

REFERENCES

Abdul-Latif, O., Shepherd, P., and Pennock, S. (2007). TDOA/AOA data fusion for enhancing positioning in an ultra wideband system, *Proceedings of Signal Processing and Communications*, ICSPC, IEEE, November 2007, Dubai, United Arab Emirates.

Al-Hussein, M. (1999). *An integrated information system for crane selection and utilization*, Ph.D. Thesis, Dept. of Building Civil & Environmental Engineering, Concordia University.

AlBahnassi, H. (2009). Real-time motion planning and simulation of cranes in construction, Master thesis of the Department of Building, Civil and Environmental Engineering at Concordia University.

AlBahnassi, H., and Hammad, A. (2010). Real-time motion planning and simulation of cranes in construction: framework, Submitted to *Journal of Computing in Civil Engineering*, ASCE.

Ali, M.S.A.D., Babu, N.R., and Varghese, K. (2005). Collision free path planning of cooperative crane manipulators using genetic algorithm, *Journal of Computing in Civil Engineering*, ASCE, Vol. 19, No. 2, pp. 182-193.

Allen, T.T. (2006). Introduction to Engineering Statistics and Six Sigma: Statistical Quality Control and Design of Experiments and Systems, Springer, London.

Alshibani, M., and Moselhi, O. (2007). Tracking and forecasting performance of earthmoving operation using GPS, *Proceedings of Construction Management and Economics 25th Anniversary Conference*, July 2007, Reading, UK.

Angeles, J. (1997). *Fundamentals of robotic mechanical systems: theory, methods, and algorithm*, Springer-Verlag, New York, N.Y., USA.

ArcGIS 3D Analyst (2010). <www.esri.com/software/arcgis/extensions/3danalyst/index.html> (June, 25, 2010).

Arizono, N., Shiota, K., Uemura, Y., Sasada, T., Morikawa, Y., and Onoda, M. (1993). Crane operation support system, Res. Dev., Vol. 43, No. 1, pp. 47-50.

Arras, K., Castellanos, J.A., Schilt, M., and Siegwart, R. (2003). Feature-based multi-hypothesis localization and tracking using geometric constraints, *Robotics and Autonomous Systems*, Vol. 44, pp. 41-53.

Autodesk Softimage (2010). <usa.autodesk.com/adsk/servlet/pc/index?id=13571168&siteID=123112> (June 25, 2010).

Ban, Y. (2002). Unmanned construction system: present status and challenges, *Proceedings of the 19th International Symposium on Automation and Robotics in Construction*, September 2002, Gaithersburg, Maryland, U.S., pp. 241-246.

Bandi, S., and Thalmann, D. (1997). A configuration space approach for efficient animation of human figures, *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, NAM, June 1997, Puerto Rico, pp. 38-45.

Barraquand, J., Kavraki, L., Latombe, J.C., Li, T.Y., Motwani, R., and Raghavan, P. (1997). A random sampling scheme for path planning, *International Journal of Robotics Research*, Vol. 16, No. 6, pp. 759-774.

Beavers, J.E., Moore, J.R., Rinehart, R., and Schriver, W.R. (2006). Crane-related fatalities in the construction industry, *Journal of Construction Engineering and Management*, ASCE, Vol. 132, No. 9, pp. 901-910.

Berg, J., Ferguson, D., and Kuffner, J. (2006). Anytime path planning and replanning in dynamic environments, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, May 2006, Orlando, Florida, U.S., pp. 2366-2371.

Bilek, J., and Hartmann, D. (2003). Development of an agent-based workbench supporting collaborative structural design, *Proceedings of the 20th CIB W78 Conference on IT in Construction*, April 2003, Waiheke Island, New Zealand.

Bouvel, D., Froumentin, M., and Garcia, G. (2001). A real-time localization system for compactors, *Automation in Construction*, Vol. 10, pp. 417-428.

Brandt, D. (2006). Comparison of A* and RRT-Connect motion planning techniques for self-reconfiguration planning, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, Beijing, China.

Bruce, J.R. (2004). Robot Motion Planning. Computer Science Department Carnegie Mellon University, <www4.cs.umanitoba.ca/~jacky/Robotics/Papers/Bruce-ERRTMotionPlanningSlides.pdf> (June 25, 2010).

Bruce, J.R., and Veloso, M.M. (2006). Safe multirobot navigation within dynamics constraints, *Proceedings of the IEEE*, *Special Issue on Multi-Robot Systems*, Vol. 94, No. 7, pp. 1398-1411.

Bunin, S.G., and Valikov, D.P. (2006). Widening UWB system coverage zone by repeaters-regenerators, *Proceedings of the Ultrawideband and Ultrashort Impulse Signals*, September 2006, Sevastopol, Ukraine, pp. 251-252.

Canadarm (2009). The Canadian Crane, <www.nasa.gov/audience/foreducators/k-4/features/F_Canadian_Crane.html> (June 25, 2010)

Carbonari, A., Naticchia, B., Giretti, A., and De Grassi, M. (2009). A proactive system for real-time safety management in construction sites, *Proceedings of the 26th International Symposium on Automation and robotics in Construction*, June 2009, Austin, Texas, U.S.

Caterpillar (2010). Mining equipment from Caterpillar, <www.cat.com/cda/components/fullArticle?m=38462&x=7&id=1067282> (June 25, 2010).

Chae, S., and Yoshida, T. (2008). A study of safety management using working area information on construction site, *Proceedings of the 25th International Symposium on Automation and Robotics in Construction*, June 2008, Vilnius, Lithuania, pp. 292-299.

Chang, C., Chung, M., and Lee, B. (1994). Collision avoidance of two general robot manipulators by minimum delay time, *IEEE Trans. on Systems, Man & Cybernetics*, Vol. 24, No. 3, pp. 517-522.

Cheng, P., and LaValle, M. (2002). Resolution complete rapidly-exploring random trees, *Proceedings of International Conference on Robotics & Automation*, May 2002, Washington, D.C., U.S. Vol. 1, pp. 267-272.

Chi, S., and Caldas, C.H. (2009). Development of an automated safety assessment framework for construction activities, *Proceedings of the 26th International Symposium on Automation and Robotics in Construction*, ISARC, June 2009, Austin, Texas, U.S., pp. 38-46.

Cho, Y.K., Youn, J.H., and Martinez, D. (2010). Error modeling for an untethered ultra-wideband system for construction indoor asset tracking, *Automation in Construction*, Vol. 19, pp. 43-54.

Choset, H., and Burdick, J. (2000). Sensor based exploration: the hierarchical generalized Voronoi graph, *International Journal of Robotics Research*, Vol. 19, No. 2, pp. 96-125.

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and Thrun, S. (2005), *Principles of Robot Motion – Theory, Algorithms, and Implementations*, The MIT Press, Cambridge.

Clark, C., Rock, S., and Latombe, J. (2003). Motion planning for multirobot systems using dynamic robot networks, *Proceedings of the IEEE International Conference on Robotics and Automation*, September 2003, Taiwan, pp. 4222-4227.

Clifton, M., Paul, G., Kwok, N., Liu, D., and Wang, D. (2008). Evaluating performance of multiple RRTs, *Proceedings of the International Conference on Mechtronic and Embedded Systems and Applications*, MESA, IEEE/ASME, October 2008, Beijing, China, pp. 564-569.

Craig, J. (2004). *Introduction to Robotics Mechanics and Control Third Edition*, Prentice Hall, New Jersey.

Crane Accident Statistics (2010). <www.craneaccidents.com/stats.htm> (June 25, 2010).

Crane-Related Occupational Fatalities (2006). <www.bls.gov/iif/oshwc/osh/os/osh_crane_2006.pdf > (June 25, 2010).

Crane Safety on Construction Sites (1998). ASCE Manuals and Reports on Engineering Practice No. 93, American Society of Civil Engineers.

Cranimax (2010). <www.cranimax.com> (June 25, 2010).

CSST (2010). Commission de la santé et de la sécurité du travail du Québec, <www.csst.qc.ca/portail/fr/> (June 25, 2010).

Davis, B.R., and Sutton, S.C. (2003). A guide to crane safety, N.C. Department of Labor, Division of Occupational Safety and Health.

Durfee, E. (1999). Distributed problem solving and planning, *Multiagent Systems – A Modern Aapproach to Distributed Artificial Intelligence*, Gerhard Weiss (ed.). MIT Press.

Ferber, J. (1999). *Multi-Agent System, an Introduction to Distributed Artificial Intelligence*, Addison-Wesley, Pearson Education Limited.

Ferguson, D. (2006). *Single Agent and Multi-agent Path Planning in Unknown and Dynamic Environments*, Ph. D. dissertation of the Robotics Institute Carnegie Mellon University, Pittsburgh, Pennsylvania.

Ferguson, D., Kalra, N., and Stentz, A. (2006). Replanning with RRTs, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, May 2006, Orlando, Florida, U.S., pp. 1243-1248.

FIPA (2010). The Foundation for Intelligent Physical Agents, </www.fipa.org/> (June 25, 2010).

Fontana, R.J. (2007). Ultra wideband technology – obstacles and opportunities, *Plenary Talk, European Microwave Conference,* Munich, Germany.

Fullerton, C. E., Allread, B. S., and Teizer, J. (2009). Pro-active-real-time personnel warning system, *Proceedings of the Construction Research Congress,* April 2009, Seattle, Washington, U.S.

Furlani, K.M., Latimer, IV D.T., Gilsinn, D.E., and Lytle, A.M. (2002). Prototype implementation of an automated structural steel tracking system, *Proceedings of the 19th International Symposium on Automation and Robotics in Construction*, September 2002, Maryland, U.S., pp. 467-473.

Ghavami, M., Michael, L.B., and Kohno, R. (2004). *Ultra Wideband Signals and Systems in Communication Engineering,* Wiley.

Gireesh, K.T., and Vijayan, V.P. (2007). A multi-agent optimal path planning approach to robotics environment, *Proceedings of International Conference on Computational Intelligence and Multimedia Application*, IEEE, December 2007, Vol. 1, No. 13-15, pp. 400-404.

Giretti, A., Carbonari, A, Naticchia, B., and Grassi, M.D. (2009). Design and first development of an automated real-time safety management system for construction sites, *Journal of Civil Engineering and Management*, ASCE, Vol. 15, No. 4, pp. 325-336.

Gordon, C., and Akinci, B. (2005). Technology and process assessment of using LADAR and embedded sensing for construction quality control, *Proceedings of Construction Research Congress 2005: Broadening Perspectives*, April 2005, San Diego, U.S.

Groove Crane. (2008). *TMS870/TTS870 Product Guide*, Manitowoc Crane Group.

GUAY (2010). TMS300, <gruesguay.com/en/equipements/grueTeleCamion.html> (June 25, 2010).

Hammad, A., Wang, H., and Mudur, S.P. (2009). Distributed augmented reality for visualizing collaborative construction tasks, *Journal of Computing in Civil Engineering*, Vol. 23, No. 6, pp. 418-427.

Hammad, A., Zhang, C., AlBahnassi, H., and Rodriguez, S. (2010). Path Re-planning of Cranes Using Real-Time Location System, submitted to *Journal of Automation in Construction* (accepted with modification).

Hammad, A., Zhang, C., Al-Hussein, M., and Cardinal, G. (2007). Equipment workspace analysis in infrastructure projects, *Canadian Journal of Civil Engineering*, Vol. 34, No. 10, pp. 1247-1256.

Heavy Equipment Safety (2005). Commonwealth of Virginia Workers' Compensation Program, <www.covwc.com/lcarticles/archives/000092.php> (June 25, 2010).

Hirschmann (2008). Electronic control systems, <hus.hirschmann.com/English/Electronic_Control_Systems/index.phtml > (June 25, 2010).

Hobby Engine (2010). <www.hobbyengine.com.hk/catalog/index.php > (June 25, 2010).

Hwang, K., Ju, M., and Chen, Y. (2003). Speed alternation strategy for multijoint robots in co-working environment, *IEEE Trans. on Industrial Electronics*, Vol. 50, No. 2, pp. 385-393.

Hwang, Y.K., and Ahuja, N. (1992). Gross motion planning: a survey, *ACM Computing Surveys*, Vol. 24, No. 3, 219-291.

ICT Roadmap for Construction. (2003). <cic.vtt.fi/projects/roadcon/docs/roadcon_d52.pdf> (June 25, 2010).

Jadex (2008). <vsis-www.informatik.uni-hamburg.de/projects/jadex/features.php> (June 25, 2010).

Kalra, N., Ferguson, D., and Stentz, A. (2005). Hoplites: A market-based framework for planned tight coordination in multirobot teams, *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005, Barcelona, Spain, pp. 1170-1177.

Kamat, V.R., and Martinez, J.C. (2001). Visual simulated construction operations in 3D, *Journal of Computing in Civil Engineering*, ASCE, Vol. 15, No. 4, pp. 329-337.

Kamezaki, M, Iwata, H., and Sugano, S. (2009). Work state identification using primitive static states – implementation to demolition work in double-front work machines, *Proceedings of the 26th International Symposium on Automation and Robotics in Construction*, ISARC, June 2009, Austin, Texas, U.S., pp. 278-287.

Kang, S. (2005). *Computer planning and simulation of construction erection processes using single or multiple cranes*, Ph.D. Dissertation of the Department of Civil and Environmental Engineering, Stanford University.

Kang, S., and Miranda, E. (2006). Planning and visualization for automated robotic crane erection processes in construction, *Automation in Construction*, Vol. 15, pp. 398-414.

Karaman, S., and Frazzoli, E. (2010). Incremental sampling-based algorithms for optimal motion planning, *Proceedings of the 2010 Robotics: Science and Systems Conference*, June, 2010, Zaragoza, Spain, <www.roboticsproceedings.org/rss06/p34.pdf> (August 9, 2010).

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-98.

Kim, K., and Paulson, Jr.B.C. (2003). Agent-based compensatory negotiation methodology to facilitate distributed coordination of project schedule changes, *Journal of Computing in Civil Engineering*, ASCE, Vol. 17, No. 1, pp. 10-18.

Komtrax (2010) <www.komatsuamerica.com/?p=komtrax&pageid=172> (June 25, 2010).

Kuffner, J.J., and LaValle, S.M. (2000). RRT-connect: an efficient approach to single-query path planning, *Proceedings of IEEE International Conference on Robotics and Autonomous Systems*, pp. 995-1001.

Latombe, J.C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, USA.

LaValle, S.M. (1998). Rapidly-exploring random trees: A new tool for path planning, Computer Science Dept., Iowa State University. <msl.cs.uiuc.edu/~lavalle/papers/Lav98c.pdf> (June 25, 2010).

LaValle, S.M. (2006). *Planning Algorithm*, Cambridge University Press, New York.

LaValle, S.M., and Kuffner, J.J. (1999). Randomized kinodynamic planning, *Proceedings of Robotics and Automation*, May 1999, Detroit, Michigan, U.S., Vol. 1, pp. 473-479.

LaValle, S.M., and Kuffner, J.J. (2000). Rapidly-exploring random trees: progress and prospects, *Proceedings of the Fourth International Workshop on Algorithmic Foundations of Robotics*, March 2000, Dartmouth, Hanover, NH.

LaValle, S.M., and Kuffner, J.J. (2001). Rapidly-exploring random trees: Progress and prospects, *Algorithmic and Computational Robotics: New Directions*, A K Peters, Wellesley, MA.

Lee, J., and Bernold, L.E. (2008). Ubiquitous agent-based communication in construction, *Journal of Computing in Civil Engineering*, ASCE, Vol. 22, No. 1, pp. 31-39.

LiftPlanner (2010). <www.liftplanner.net> (June 25, 2010).

Lozano-Pérez, T., and Wesley, M. (1979). An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, Vol. 22, No. 10, pp. 560-570.

Lytle, A.M., Katz, I., and Saidi, K.S. (2005). Performance evaluation of a high-frame rate 3D range sensor for construction applications, *Proceedings of the 22nd International Symposium on Automation and Robotics in Construction*, September 2005, Ferrara, Italy.

Lytle, A.M., Saidi, K.S., Bostelman, R.V., Stone, W.C., and Scott, N.A. (2004). Adapting a teleoperated device for autonomous control using three-dimensional positioning sensors: experiences with the NIST RoboCrane, *Automation in Construction*, Vol. 13, pp. 101-118.

Lytle, A.M., Saidi, K.S., and Stone, W.C. (2002). Development of a robotic structural steel placement system, *Proceedings of the 19th International Symposium on Automation and Robotics in Construction*, September 2002, Gaithersburg, Maryland, U.S., pp. 263-268.

Machinedesign (2004). Onboard vehicle testing gets an upgrade, <machinedesign.com/ContentItem/62380/Onboardvehicletestinggetsanupgrade.aspx> (June 25, 2010).

Mailhot, G., and Busuio, S. (2006). Application of long range 3D laser scanning for remote sensing and monitoring of complex bridge structure, *Proceedings of the 7th International Conference of Short & Medium Span Bridges*, August 2006, Montreal, Canada.

Manitowoc (2010). Manitowoc products – GMK6300. <http://www.manitowoccranes .com/MCG_GRO/Products/UK/GMK6300.asp> (June 25, 2010).

Marsh L., Calbert G., Tu J., Gossink D., and Kwok, H. (2005). Multi-agent UAV path planning, *Defence Science and Technology Organisation Scientific Publication Online*.

Mathworks (2010). <www.mathworks.com/ > (June 25, 2010).

MESA Imaging. (2010). <www.mesa-imaging.ch/prodviews.php> (June 25, 2010).

Moon construction (2010). <www.constructionequip.net/2009/02/27/construction-is-not-limited-to-earth/#more-429> (June 25, 2010).

Moore, M., and Wilhelms, J. (1988). Collision detection and response for computer animation, *Computer Graphics*, Vol. 22, No. 4, pp. 289-298.

Motion Strategy Library (2003). <msl.cs.uiuc.edu/msl/> (June 25, 2010).

Multispectral Solution. (2010). <www.multispectral.com/products/sapphire.htm> (June 25, 2010).

Munoz, D., Lara, F.B., Vargas, C., and Enriquez-Caldera, R. (2009). *Position location techniques and applications*, Academic Press, Amsterdam; Boston.

Muthukrishnan, K., and Hazas, M. (2009). Position estimation from UWB pseudorange and angle-of-arrival: a comparison of non-linear regression and kalman filtering, *Proceedings of the 4th International Symposium on Location and Context Awareness*, May 2009, Tokyo, Japan, pp. 222-239.

MySQL (2010). <www.mysql.com/> (June 25, 2010).

Navon, R., Goldschmidt, E., and Shpatnisky, Y. (2004). A concept proving prototype of automated earthmoving control, *Automation in Construction*, Vol. 13, No. 2, pp. 225-239.

NASA (2010). Lunar robotic manipulator passes early test, <www.nasa.gov/exploration/home/lsms.html> (June 25, 2010).

Ni, J., Arndt, D., Ngo, P., Phan, C., Dekome, K., and Dusl, J. (2010). “Ultra-wideband time-difference-ofarrival high resolution 3D proximity tracking system.” Proc., the Position Location and Navigation Symposium (PLANS), Indian Wells, CA, USA, 37-43.

NIOSH (2010). The National Institute for Occupational Safety and Health, Goal 1: Reduce the major risks associated with traumatic injuries and fatalities in construction, <www.cdc.gov/niosh/nas/construction> (June 25, 2010).

NIST (2010). National Institute of Standards and Technology, <www.nist.gov> (June 25, 2010).

O’Donnell, P., and Lozano-Pérez, T. (1989). Deadlock-free and collision-free coordination of two robot manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 1989, Scottsdale, AZ, U.S., Vol. 1, pp. 484-489.

Orbit Communications. (2008). Vehicle proximity alert and collision avoidance system from orbit communications, <www.ferret.com.au/c/Orbit-Communications/Vehicle-Proximity-Alert-and-Collision-Avoidance-System-from-Orbit-Communications-n778105> (June 25, 2010).

Paul, G., and Clifton, M. (2008). Basic implementation of the RRT search algorithm, <www.mathworks.com/matlabcentral/fx_files/21443/1/content/RRT20080916/html/RRT.html> (June 25, 2010).

Peyret, F., Jurasz, J., Carrel, A., Zekri, E., and Gorham, B. (2000). The computer integrated road construction project, *Automation in Construction*, Vol. 9, pp. 447-461.

Phidgets (2010). < www.phidgets.com/ > (June 25, 2010).

Photogrammetry (2007). <www.photogrammetry.com> (June 25, 2010).

Potain (2008). <www.potain.com/products/prdRange_home.cfm?objectID=000D8E06-72C2-1ABF-B59480F03BE70000&selectedHeadingID=EQUIPEMENT> (June 25, 2010).

Price, M., Kenney, J., Eastman, R.D., and Hong, T. (2007). Training and optimization of operating parameters for flash LADAR cameras, *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, April 2007, Roma, Italy, pp. 3408-3413.

Proximity Query Package (PQP) (1999). <www.cs.unc.edu/~geom/SSV/> (June 25, 2010).

RC-Truck-Club (2010). <www.rc-truck.com> (June 25, 2010).

Reddy, H.R., and Varghese, K. (2002). Automated path planning for mobile crane lifts, *Computer-aided Civil and Infrastructure Engineering*, Vol. 17, pp. 439-448.

Reif, J. (1979). Complexity of the mover's problem and generalizations, *Proceeding of 20th IEEE Symposium on Foundations of Computer Science*, pp. 421-427.

Ren, Z., and Anumba, C. (2002). Learning in multi-agent systems: a case study of construction claims negotiation, *Advanced Engineering Information*, Vol. 16, No. 4, pp. 265-275.

Riaz, Z., Edwards, D.J., and Thorpe, A. (2006). SightSafety: A hybrid information and communication technology system for reducing vehicle/pedestrian collisions, *Automation in Construction*, Vol. 15, pp. 719-728.

Ritchie, D. (2004). Crane simulators as training tools, *CraneWorks*, Vol. 16.

Russell, S., and Norvig, P. (2003). *Artificial Intelligence*, *A Modern Approach*, second edition, Prentice Hall.

Saidi, K., and Lytle, A. (2008). NIST Research in Crane Automation: 2007 Overview, *the 87th Annual Meeting*, Transportation Research Board.

Shapiro, H.I., Shapiro, J.P., and Shapiro, L.K. (2000). *Cranes and Derricks*, 3rd Ed., McGraw-Hill, New York.

Simlog (2010). <www.simlog.ca> (June 25, 2010).

Sivakumar, PL., Varghese, K., and Babu, N.R. (2003). Automated path planning of cooperative crane lifts using heuristic search, *Journal of Computing in Civil Engineering*, ASCE, Vol. 17, No. 3, pp. 197-207.

Spong, M.W., Lewis, F.L., and Abdallah, C.T. (1992). *Robot Control – Dynamics, Motion Planning, and Analysis*, IEEE Press, IEEE Control Systems Society.

SSPC (2009). The Society for Protective Coatings. *Safety Tips for Aerial Lifts*. <www.sspc.org/regnews/regnewsother/aerialsafety.html> (June 25, 2010).

Stentz, A. (1995). The focused D* algorithm for real-time replanning, *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995, Montreal, Canada.

Sud, A., Andersen, E., Curtis, S., Lin, M., and Manocha, D. (2007). Real-time path planning for virtual agents in dynamic environment, *Proceedings of IEEE Virtual Reality Conference*, March 2007,Charlotte, U.S., pp. 91-98.

Tavakoli, Y., Javadi, H.H.S., Adabi, S. (2008). A cellular automata based algorithm for path planning in multi-agent systems with a common goal, *International Journal of Computer Science and Network Security*, Vol. 8, No. 7, pp. 119-123.

Teizer, J. (2006). RAPIDS Laborotary, <rapids.ce.gatech.edu/pe_research.htm> (June 25, 2010).

Teizer, J., Bosche, F., Caldas, C.H., and Hass, C.T. (2006). Real-time spatial detection and tracking of resources in a construction environment, *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, June 2006, Montreal, Canada, pp. 494-502.

Teizer, J., and Castro-Lacouture, D. (2007). Combined ultra-wideband positioning and range imaging sensing for productivity and safety monitoring in building construction, *Proceedings of the 2007 ASCE International Workshop on Computing in Civil Engineering*, ASCE, July 2007, Pittsburgh, Pennsylvania, U.S., pp. 681-688.

Teizer, J., Lao, D., and Sofer, M. (2007). Rapid automated monitoring of construction site activities using ultra-wideband, *Proceedings of the 24th International Symposium on Automation & Robotics in Construction*, ISARC, September 2007, Kochi, Kerala, India, pp. 23-28.

Teizer, J., Liapi, K.A., Caldas, C.H., and Haas, C.T. (2005). Experiments in real-time spatial data acquisition for obstacle detection, *Construction Research Congress*, April 2005, San Diego, CA, U.S.

Topcon (2008). <www.topconpositioning.com/news-events/single/item/topcon-debuts-new-2-d-excavator-grade-control-system> (June 25, 2010).

Trimble GCS900 (2010). Trimble GCS900 grade control system. <www.trimble.com/GCS900.shtml> (June 25, 2010).

Trimble GX (2010). Trimble GX 3D scanner for spatial imaging-accurate terrestrial positioning data for the geospatial industry. <www.trimble.com/trimblegx.shtml> (June 25, 2010).

Tserng, H.P., Ran, B., and Russell, J. S. (2000). Interactive path planning for multi-equipment landfill operations, *Automation in Construction*, Vol. 10, pp. 155-168.

Ubisense Manual (2009). Ubisense Location Engine Configuration User Manual.

Ubisense (2010). <www.ubisense.net/en> (June 25, 2010).

Urmson, C., and Simmons, R. (2003). Approaches for heuristically biasing RRT growth, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* IROS, October 2003, Las Vegas, Nevada, U.S., pp. 1178-1183.

Varghese, K., Dharwadkar, P., Wolfhope, J., and O'Connor, J.T. (1997). A heavy lift planning system for crane lifts, *Microcomputers in Civil Engineering*, Vol. 12, No. 1, pp. 31-42.

Ward, A.M.R., and Webster, P.M. (2009). Location Device and System using UWB and Non-UWB Signals. *Patent US 7,636,062 B2.* Date of patent: Dec. 22, 2009.

Wing, R. (2006). RFID application in construction and facilities management, *Journal of Information Technology in Construction,* Vol. 11, pp. 711-721.

WorkSafeBC (2010). <worksafebc.com> (June 25, 2010).

Yang, L., and Hammad, A. (2007). Ad-hoc wireless networking for supporting communication and onsite data collection, *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, July 2007, Pittsburgh, U.S., pp. 854-861.

Zaki, A.R., and Mailhot, G. (2003). Deck reconstruction of Jacques Cartier Bridge using precast prestressed high performance concrete panels, *PCI Journal*, Vol. 48, No. 5, pp. 20-33.

Zhang, C., Hammad, A., and AlBahnassi, H. (2009). Collaborative Multi-Agent Systems for Construction Equipment Based on Real-Time Field Data Capturing, ITcon Vol. 14, *Special Issues on Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and Scenario Planning*, pp. 204-228, <www.itcon.org/2009/16> (June 25, 2010).

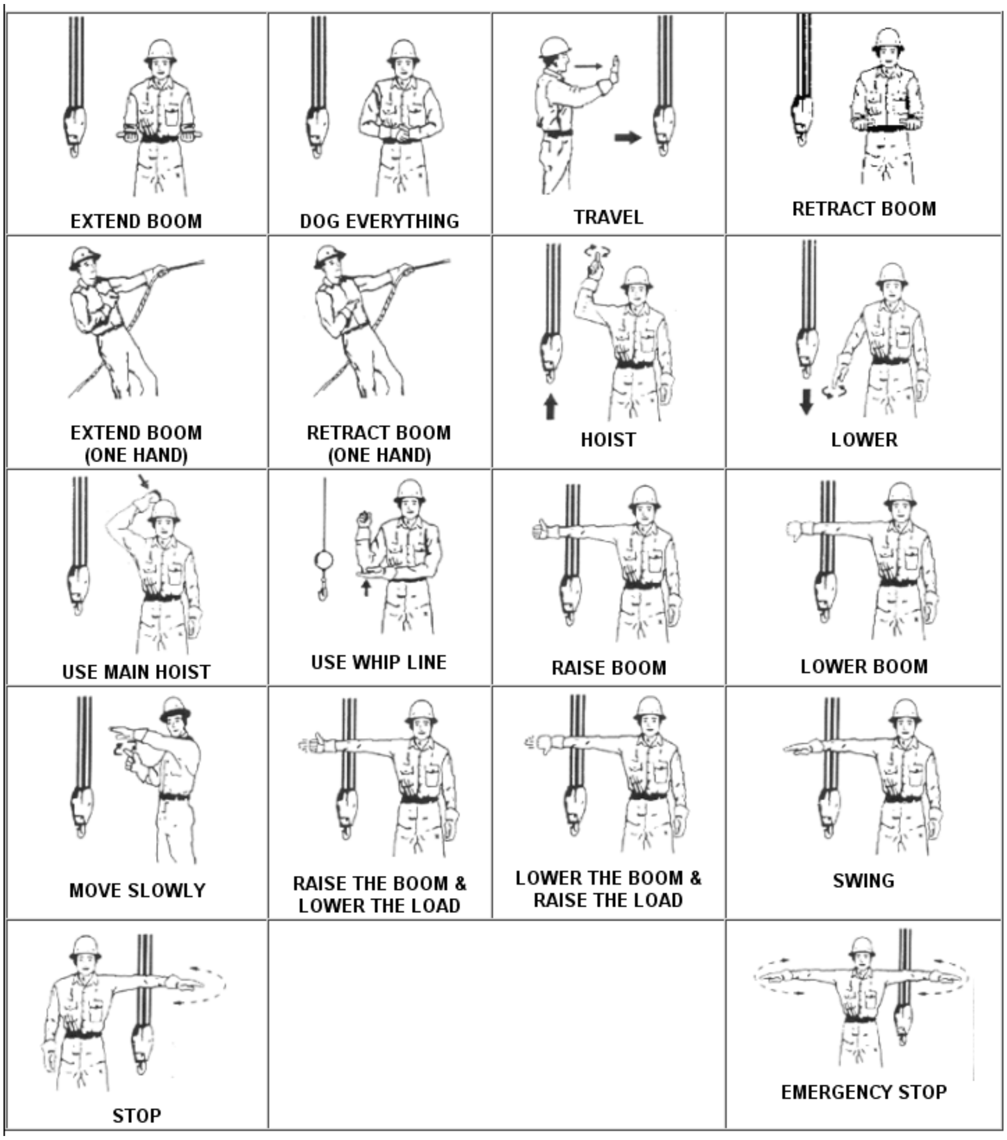
Zhang, C., Hammad, A., Zayed, T.M., Wainer, G., and Pang, H. (2007). Cell-based representation and analysis of spatial resources in construction simulation, *Journal of Automation in Construction*, Vol. 16, No. 4, pp. 436-448.

Zheng, Y. (1989). Kinematics and dynamics of two industrial robots in assembly, *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1989, Scottdale, A.Z., U.S., Vol. 3, pp. 1360-1365.

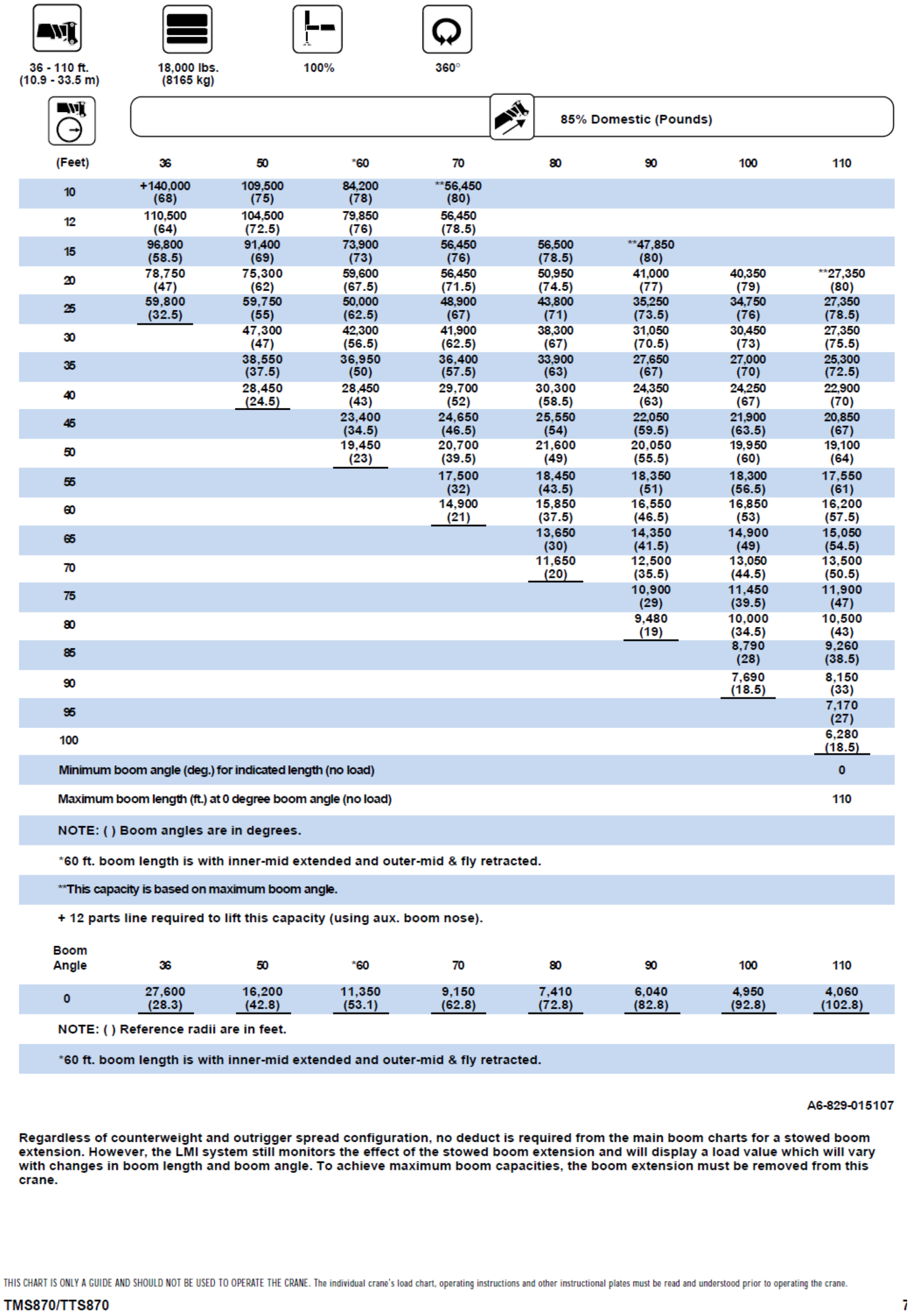
Zhou, S., and Zhang, S. (2007). Co-simulation on automatic pouring of truck-mounted concrete boom pump, *Proceedings of the IEEE International Conference on Automation and Logistics*, August 2007, Jinan, China, pp. 928-932.

Zilberstein, S., and Russell, S. (1995). *Approximate Reasoning using Anytime Algorithms, Imprecise and Approximate Computation*, Kluwer Academic Publishers.

APPENDIX A – STANDARD HAND SIGNALS FOR CRANE OPERATION



APPENDIX B – EXAMPLE OF CRANE LOAD CHART



36 - 110 ft.
(10.9 - 33.5 m)

18,000 lbs.
(8165 kg)

100%

360°

85% Domestic (Pounds)

(Feet)	36	50	*60	70	80	90	100	110
10	+140,000 (68)	109,500 (75)	84,200 (78)	**56,450 (80)				
12	110,500 (64)	104,500 (72.5)	79,850 (76)	56,450 (78.5)				
15	96,800 (58.5)	91,400 (69)	73,900 (73)	56,450 (76)	56,500 (78.5)	**47,850 (80)		
20	78,750 (47)	75,300 (62)	59,600 (67.5)	56,450 (71.5)	50,950 (74.5)	41,000 (77)	40,350 (79)	**27,350 (80)
25	59,800 (32.5)	59,750 (55)	50,000 (62.5)	48,900 (67)	43,800 (71)	35,250 (73.5)	34,750 (76)	27,350 (78.5)
30		47,300 (47)	42,300 (56.5)	41,900 (62.5)	38,300 (67)	31,050 (70.5)	30,450 (73)	27,350 (75.5)
35		38,550 (37.5)	36,950 (50)	36,400 (57.5)	33,900 (63)	27,650 (67)	27,000 (70)	25,300 (72.5)
40		28,450 (24.5)	28,450 (43)	29,700 (52)	30,300 (58.5)	24,350 (63)	24,250 (67)	22,900 (70)
45			23,400 (34.5)	24,650 (46.5)	25,550 (54)	22,050 (59.5)	21,900 (63.5)	20,850 (67)
50			19,450 (23)	20,700 (39.5)	21,600 (49)	20,050 (55.5)	19,950 (60)	19,100 (64)
55				17,500 (32)	18,450 (43.5)	18,350 (51)	18,300 (56.5)	17,550 (61)
60				14,900 (21)	15,850 (37.5)	16,550 (46.5)	16,850 (53)	16,200 (57.5)
65					13,650 (30)	14,350 (41.5)	14,900 (49)	15,050 (54.5)
70					11,650 (20)	12,500 (35.5)	13,050 (44.5)	13,500 (50.5)
75						10,900 (29)	11,450 (39.5)	11,900 (47)
80						9,480 (19)	10,000 (34.5)	10,500 (43)
85							8,790 (28)	9,260 (38.5)
90							7,690 (18.5)	8,150 (33)
95								7,170 (27)
100								6,280 (18.5)
Minimum boom angle (deg.) for indicated length (no load)								0
Maximum boom length (ft.) at 0 degree boom angle (no load)								110

NOTE: ( ) Boom angles are in degrees.

*60 ft. boom length is with inner-mid extended and outer-mid & fly retracted.

**This capacity is based on maximum boom angle.

+ 12 parts line required to lift this capacity (using aux. boom nose).

Boom Angle	36	50	*60	70	80	90	100	110
0	27,600 (28.3)	16,200 (42.8)	11,350 (53.1)	9,150 (62.8)	7,410 (72.8)	6,040 (82.8)	4,950 (92.8)	4,060 (102.8)

NOTE: ( ) Reference radii are in feet.

*60 ft. boom length is with inner-mid extended and outer-mid & fly retracted.

A6-829-015107

Regardless of counterweight and outrigger spread configuration, no deduct is required from the main boom charts for a stowed boom extension. However, the LMI system still monitors the effect of the stowed boom extension and will display a load value which will vary with changes in boom length and boom angle. To achieve maximum boom capacities, the boom extension must be removed from this crane.

THIS CHART IS ONLY A GUIDE AND SHOULD NOT BE USED TO OPERATE THE CRANE. The individual crane's load chart, operating instructions and other instructional plates must be read and understood prior to operating the crane.

TMS870/TTS870 7

APPENDIX C – UBISENSE SERIES 7000 SENSOR SPECIFICATIONS (UBISENSE, 2010)

Size and Weight	
Dimensions	20cm x 13cm x 6cm (8” x 5” x 2.5”)
Weight	650g (23 oz)
Operating Conditions	
Temperature	0 ℃ to 60 ℃ (32 ℉ to 140 ℉)
Humidity	0 to 95%, non-condensing
Enclosure	IP30
Location Performance	
Operating Range	Up to 160m (520ft) in open field conditions
Achievable Accuracy	Better than 30cm (12”) in 3D
Radio Frequencies	
Ultra-wideband	6GHz – 8GHz
Telemetry channel	2.4GHz
Certifications	
FCC Part 15 (FCC ID SEASENSOR20)	
EU CE	
Power Supply	
Power-over-Ethernet IEEE 802.3af compatible 12V DC @ 10W (optional)	
Mounting Options	
Adjusting mounting bracket (supplied)	
Ubisense Part Codes	
UBISENSOR7000, UBISENSPS (optional 12V power supply)	

APPENDIX D – UBISENSE SERIES 7000 COMPACT TAG SPECIFICATIONS

Size and Weight	
Dimensions	38mm x 39mm x 16.5mm (1.50” x 1.53” x 0.65”)
Weight	25g (0.88 oz)
Operating Conditions	
Standard	-20 ℃ to 60 ℃ (-4 ℉ to 140 ℉)
Extended	-30 ℃ to 70 ℃ (-22 ℉ to 158 ℉)
Humidity	0 to 95% non-condensing
Enclosure	
Standard	IP63
Extended	IP65
Location Performance	
Update Rate	0.00225Hz to 33.75Hz (can be varied dynamically under software control)
Peripherals	
LED (application controllable)	
Push button (application controllable)	
Motion detector	
Radio Frequencies	
Ultra-wideband	6GHz – 8GHz
Telemetry channel	2.4GHz
Certifications	
FCC Part 15 (FCC ID SEATAG22, SEATAG22HH)	
EU CE	
Power Supply	
3V coin cell (CR2477)	
Mounting Options	
Industrial adhesive pas (supplied)	
Industrial velcro	
Magnetic mountings	
Screw mountings	
Ubisense Part Codes	
Standard	UBITAG70022
Extended	UBITAG7022 X-65

APPENDIX E – UWB SYSTEM INSTALLATION PROCEDURES

NEEDED EQUIPMENTS

Sensors, Mounting equipments, Cables (4 regular network cables and 3 thick timing cable), Computer, Software: (Ubisense package, DHCP server, solver),
Switch and power: (PoE switch plus its power cables, power generator for outdoor usage)
Level

LAYOUT DESIGN STEPS

1. Conceptual connectivity design (daisy chain, star, extended start)
2. Decide where to put the sensors in the yard
3. Decide where to put one tag for calibrating the sensors
4. Decide two points that are easy to measure on the yard
5. Decide how to run the cables and protect them
6. Decide where to put the switch
7. Draw the connectivity map
8. Decide the reference point (0, 0, 0)
9. Fill out table (Calibration tag ID info, Sensors: x, y, z and mac address)

SITE PREPARATION

1. Fix the sensors on the mounting device on the designated place
2. Put the switch on its designated place and attach to the power
3. Run the network cables from switch to sensors and fix the cables
4. Run the timing cables based on the connectivity map (most often between the sensors)

MEASUREMENTS

1. Measure the monitored area: W, L, H
2. Measure x, y, z of the sensors (one by one or using the solver)
    a. Decide on two points and measure the x, y of them, (preferably set the (0,0) on the corner of two walls in the area)
    b. Measure the distance from the points to the sensors
    c. Enter them in the solver
    d. Get the x, y, z of the sensors

BASIC CONFIGURATION

1. Restart the PC (switch is powered, sensors are not connected, PC is connected to switch)
2. Attach the computer to the switch
3. Start DHCP server
4. Open platform control

5. Make sure the services are running (no prefix, not in standalone mode)
6. Open location engine
7. Open log tab
8. Connect the sensor cables
9. Looking at logs to see if there are warnings

SOFTWARE CONFIGURATION

1. Open “Site Manager” and go to tab “Area”
2. Open “Notepad” and type the coordinates of the area and save as “.dat” file
3. In “Area” tab, load walls by load the .dat file
4. Go to the “Cell” tab and load the area
5. Extend the cell
6. Open location engine and load area and cell
7. Drag the sensors to the area
8. Select the master and check: “master”, “timing source” and “disable sleep”
9. Check the RF power of the cell (must be 255)
10. Check the LEDs (should be solid green)
11. Enter x, y, z of the sensors
12. Put the tag on the calibration point
13. Do dual calibration for all sensors

ASSIGN TAGS TO OBJECTS

1. Open “Site Manager” and go to tab “Objects”
2. Click “Objects” on the menu, and select “New”
3. Create new object and type
4. Open “Location Engine Config” and go to tab “Owners”
5. Click “Ownership” on the menu, and select “New” and assign tags to objects

SET UPDATE TIME SLOTS

1. Open “Location Engine Config”
2. Go to tab “tags”
3. Double click on lines, and select the Slower QoS, the Faster QoS, and the Threshold.

USING LOGGER TO RECORD DATA

1. Run UbisenseLogger
2. Click “Record/Playback file…”
3. Create a new file with “.txt” under a specified folder
4. Select objects need to be monitored
5. Go to tab “Record” and click on the red button
6. Captured events will be shown in the frame

APPENDIX F – UWB CALIBRATION WORK SHEET

Date:		**Dec. 04, 2009**	**Time:**	**9:30 am to 13:30 pm**	
Sensor name		**Sensor-1**	**Sensor-2**	**Sensor-3**	**Sensor-4**
MAC address		**00:11:CE:00:1C:3F**	**00:11:CE:00:1C:41**	**00:11:CE:01:1C:45**	**00:11:CE:01:1C:61**
Sensor position	**X**	**-8.52**	**-9.36**	**10.16**	**9.07**
	Y	**-10.54**	**12.47**	**12.1**	**-10.31**
	Z	**1.61**	**1.67**	**1.57**	**1.45**
Ref. tag name		**020-000-059-089**			
Ref. tag position	**X**	**0**			
	Y	**0**			
	Z	**2.46**			
Angel calibrated	**Yaw**	**45.8939**	**-52.1697**	**-135.879**	**134.174**
	Pitch	**22.5174**	**20.5125**	**26.6233**	**22.5969**
	Roll	**0**	**0**	**0**	**0**
Recording file name (*.txt & *.xcm)		**test.txt; gridtest.txt; taskwithfilter.txt; secondtest.txt; test.xcm; secondtest.xcm; end.xcm**			

APPENDIX G – SAMPLE OF .XCM FILE

.xcm file recording the monitored event

```
<events>
  <value>
    <instance_>0</instance_>
    <tag_>
      <id_>335559499</id_>
    </tag_>
    <timeslot_>13565</timeslot_>
    <location_>
      <flags_>0</flags_>
      <x_>-8.00000000e+000</x_>
      <y_>-8.00000000e+000</y_>
      <z_>1.00000000e+000</z_>
      <gdop_>-8.00000000e+000</gdop_>
      <error_>-8.00000000e+000</error_>
    </location_>
    <sensors_>
      <value>
        <sensor_>
          <mac_>0:11:ce:0:1c:41</mac_>
        </sensor_>
        <position_>
          <x_>3.99000001e+000</x_>
          <y_>2.09999993e-001</y_>
          <z_>2.76999998e+000</z_>
          <yaw_>2.24040937e+000</yaw_>
          <pitch_>-5.15919328e-001</pitch_>
        </position_>
        <offsets_>
          <zero_offset_>1.46800000e+003</zero_offset_>
          <cable_offset_>0.00000000e+000</cable_offset_>
        </offsets_>
        <flags_>65</flags_>
        <radar_>
          <azimuth_>-5.51666796e-001</azimuth_>
          <elevation_>1.88908026e-001</elevation_>
          <ppo_>1.49800000e+003</ppo_>
          <event1_>6.28000000e+002</event1_>
          <event2_>1.37800000e+003</event2_>
          <code_>0.00000000e+000</code_>
          <raw_>1.07599492e+004</raw_>
```

APPENDIX H – SAMPLE OF LOGGER FILE

Text file resulting from the Logger application

```
-,C-20,10/03/2009 18:53:41:449,1.82,3.93,1.13
-,C-20,10/03/2009 18:53:41:557,1.82,3.93,1.13
-,C-20,10/03/2009 18:53:41:666,1.82,3.93,1.14
-,C-20,10/03/2009 18:53:41:774,1.82,3.93,1.14
-,C-20,10/03/2009 18:53:41:882,1.81,3.93,1.15
-,C-20,10/03/2009 18:53:41:990,1.81,3.93,1.15
-,C-20,10/03/2009 18:53:42:098,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:206,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:314,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:422,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:530,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:638,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:746,1.81,3.92,1.15
-,C-20,10/03/2009 18:53:42:854,1.81,3.92,1.14
-,C-20,10/03/2009 18:53:42:963,1.81,3.92,1.14
-,C-20,10/03/2009 18:53:43:071,1.81,3.92,1.14
-,C-20,10/03/2009 18:53:43:179,1.81,3.92,1.14
-,C-20,10/03/2009 18:53:43:287,1.82,3.92,1.14
-,C-20,10/03/2009 18:53:43:395,1.81,3.92,1.13
-,C-20,10/03/2009 18:53:43:503,1.81,3.92,1.13
-,C-20,10/03/2009 18:53:43:611,1.81,3.92,1.13
-,C-20,10/03/2009 18:53:43:719,1.81,3.92,1.13
-,C-20,10/03/2009 18:53:43:827,1.81,3.92,1.13
-,C-20,10/03/2009 18:53:43:935,1.81,3.91,1.12
-,C-20,10/03/2009 18:53:44:043,1.80,3.91,1.12
-,C-20,10/03/2009 18:53:44:152,1.80,3.91,1.12
-,C-20,10/03/2009 18:53:44:260,1.80,3.91,1.13
-,C-20,10/03/2009 18:53:44:368,1.80,3.91,1.14
-,C-20,10/03/2009 18:53:44:476,1.80,3.91,1.14
-,C-20,10/03/2009 18:53:44:584,1.81,3.91,1.15
-,C-20,10/03/2009 18:53:44:692,1.81,3.91,1.15
-,C-20,10/03/2009 18:53:44:800,1.81,3.91,1.16
-,C-20,10/03/2009 18:53:44:908,1.81,3.91,1.17
-,C-20,10/03/2009 18:53:45:016,1.81,3.91,1.18
-,C-20,10/03/2009 18:53:45:124,1.81,3.91,1.18
-,C-20,10/03/2009 18:53:45:232,1.81,3.92,1.19
-,C-20,10/03/2009 18:53:45:340,1.81,3.92,1.20
-,C-20,10/03/2009 18:53:45:449,1.81,3.92,1.20
-,C-20,10/03/2009 18:53:45:557,1.81,3.92,1.20
-,C-20,10/03/2009 18:53:45:665,1.80,3.91,1.20
-,C-20,10/03/2009 18:53:45:773,1.80,3.91,1.20
-,C-20,10/03/2009 18:53:45:881,1.79,3.91,1.20
-,C-20,10/03/2009 18:53:45:989,1.79,3.91,1.19
-,C-20,10/03/2009 18:53:46:097,1.79,3.91,1.19
-,C-20,10/03/2009 18:53:46:205,1.79,3.91,1.18
-,C-20,10/03/2009 18:53:46:313,1.78,3.91,1.18
-,C-20,10/03/2009 18:53:46:421,1.78,3.91,1.18
-,C-20,10/03/2009 18:53:46:529,1.78,3.92,1.18
-,C-20,10/03/2009 18:53:46:638,1.79,3.92,1.19
```

APPENDIX I – PUBLICATIONS

a. Articles in refereed journals

(1) **Zhang, C.,** and Hammad, A. Improving Motion Planning and Re-planning of Cranes Considering Safety and Efficiency, submitted to *Advanced Engineering Informatics*.

(2) **Zhang,** C., Hammad, A., Rodriguez, S. Improving Crane Safety using UWB Real-time Location System: Data Collection and Analysis, submitted to *Journal of Construction Engineering and Management*, ASCE.

(3) Hammad, A., **Zhang, C.**, AlBahnassi, H, and Rodriguez, S. Path Re-planning of Cranes Using Real-Time Location System, submitted to *Journal of Automation in Construction* (accepted with modification).

(4) **Zhang, C.**, Hammad, A., and AlBahnassi, H. (2009) Collaborative Multi-Agent Systems for Construction Equipment Based on Real-Time Field Data Capturing, ITcon Vol. 14, Special Issues on *Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and Scenario Planning,* 204-228, http://www.itcon.org/2009/16.

(5) **Zhang, C.**, Zayed, T., and Hammad, A. (2008). Resource Management of Bridge Deck Rehabilitation: Jacques Cartier Bridge Case Study, *Journal of Construction Engineering and Management*, ASCE, 134(5): 311-319.

(6) **Zhang, C.**, Hammad, A., Zayed, T.M., Wainer, G., and Pang, H. (2007). Cell-based Representation and Analysis of Spatial Resources in Construction Simulation, *Journal of Automation in Construction*, Elsevier, 16(4): 436-448.

(7) Hammad, A., **Zhang, C.**, Al-Hussein, M. and Cardinal, G. (2007). Equipment Workspace Analysis in Infrastructure Projects, *Canadian Journal of Civil Engineering*, 34 (10): 1247-1256.

(8) Hammad, A., **Zhang, C.**, Hu, Y. and Mozaffari, E. (2006). Mobile Model-Based Bridge Lifecycle Management Systems, *Journal of Computer-Aided Civil and Infrastructure Engineering*, Blackwell Publishing, 21(7): 530-547.

b. Articles in refereed conference proceedings

(9) **Zhang, C.**, AlBahnassi, H., and Hammad, A. (2010). Improving Construction Safety through Real-time Motion Planning of Cranes, International Conference on Computing in Civil and Building Engineering, 2010, London, UK.

(10) Rodriguez, S., **Zhang, C.**, Motamedi, A., and Hammad, A. (2010). Location Tracking of Construction Resources Using UWB for better Productivity and Safety,

International Conference on Computing in Civil and Building Engineering, 2010, London, UK.

(11) **Zhang, C.**, Hammad, A., and AlBahnassi, H. (2009). Path Re-planning of Cranes Using Real-Time Location System, International Symposium on Automation & Robotics in Construction, ISARC, June, 2009, Austin, Texas.

(12) **Zhang, C.**, Hammad, A., and AlBahnassi, H. (2009). Collaborative Multi-Agent Systems for Supporting Construction Equipment, *The 2nd International Construction Specialty Conference*, CSCE, 2009.

(13) AlBahnassi, H., Hammad, A. and **Zhang, C.** (2009). Accurate Heavy Equipment Motion Planning Considering Local and Global Constraints, *The 2nd International Construction Specialty Conference*, CSCE, 2009.

(14) **Zhang, C.** and Hammad, A. (2007). Agent-based Simulation for Collaborative Cranes, Winter Simulation Conference, December, 2007, Washington, D.C., U.S.

(15) **Zhang, C.** and Hammad, A. (2007). Collaborative Agent-based System for Multiple Crane Operation, International Symposium on Automation & Robotics in Construction, ISARC, September, 2007, Kochi, Kerala, India.

(16) **Zhang, C.** and Hammad, A. (2007). Towards Supporting Construction Equipment Operation Using Collaborative Agent-based Systems, *ASCE Computing in Civil Engineering Conference*, July, 2007, Pittsburgh, U.S.

(17) Pang, H., **Zhang, C.**, and Hammad, A. (2007). Sensitivity Analysis of Construction Simulation Considering Site Layout Patterns, *ASCE Computing in Civil Engineering Conference*, July, 2007, Pittsburgh, U.S.

(18) Hong, P., **Zhang, C.**, and Hammad, A. (2006). Sensitivity Analysis of Construction Simulation Using CELL-DEVS and MICROCYCLONE, *The 2006 Winter Simulation Conference*, December, 2006, Monterey, U.S.

(19) **Zhang, C.**, and Hammad, A. (2006). 4D Modeling in Bridge Lifecycle Management Systems, *The 7th International Conference on Short & Media Span Bridges*, 2006, Montreal.

(20) **Zhang, C.**, Hammad, A., and Zayed, T.M. (2006). Simulation of the Re-decking of Jacques Cartier Bridge Considering Spatial Constraints, *The 7th International Conference on Short & Media Span Bridges*, 2006, Montreal.

(21) Hammad, A., Wang, H., **Zhang, C.**, and Al-Hussein, M. (2006). Visualizing Crane Selection and Operation in Virtual Environment, *The 6th International Conference on Construction Applications of Virtual Reality*, August, 2006, Orlando, Florida.

(22) **Zhang, C.**, and Hammad, A. (2006). Spatio-temporal Representation and Analysis in Infrastructure Systems, *The Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, 2006, Montreal.

(23) **Zhang, C.**, Hammad, A., Zayed, T.M., and Wainer G. (2006). Cell-based Representation and Analysis of Spatial Resources in Construction Simulation, *The 1st International Construction Specialty Conference*, CSCE, 2006, Calgary.

(24) **Zhang, C.**, Hammad, A., Zayed, T.M., and Wainer, G. (2005). Representation and Analysis of Spatial Resources in Construction Simulation, *Winter Simulation Conference*, 2005, Florida, U.S.A. 9 pages.

(25) **Zhang, C.**, and Hammad, A. (2005). Spatio-Temporal Issues in Infrastructure Lifecycle Management Systems, *In proceedings of the 1st CSCE Specialty Conference on Infrastructure Technologies*, Management and Policy, 2005, Toronto, Ontario, Canada. 10 pages.

(26) **Zhang, C.**, Mozaffari, E., El-Ammari, K. and Hammad, A. (2005). Integrated Framework for Lifecycle Infrastructure Management Systems, *International Congress of Urbistics*, 2005, Montreal, Canada. 8 pages.

(27) Mozaffari, E., Khabir, B., **Zhang, C.**, Devarakonda, P., Bauchkar, P., and Hammad, A. (2005). Interaction Models for Infrastructure Management Systems Using Virtual and Augmented Realities, *International Congress of Urbistics*, 2005, Montreal, Canada. 8 pages.

(28) Hammad, A., **Zhang, C.**, Hu, Y. and Mozaffari, E. (2004). Mobile Model-Based Bridge Lifecycle Management Systems, *In Proceedings of Conference on Construction Application of Virtual Reality*, ADETTI/ISCTE, September, 2004, Lisbon, pp. 109-120.

c. Book chapters

(29) Hammad, A., Pang, H., and **Zhang, C.** (to appear in 2010). Construction Simulation Using Cell-DEVS Modeling, Discrete-Event Modeling and Simulation: Theory and Applications, Wainer, G. and Mosterman, P. (editors), Taylor and Francis.