Near Real-Time Motion Planning and Simulation of Cranes in Construction: Framework and System Architecture

Homam AlBahnassi[1] and Amin Hammad, M.ASCE[2]

Abstract: Motion planning of cranes is an important issue in construction projects, where rapid and accurate planning directly affects the safety and productivity of operations. The work presented in this paper is directed toward developing a framework for near real-time motion planning of cranes that satisfies safety requirements and efficiently considers the dynamic properties of construction sites. This framework is not aiming to full automation but rather for providing assistance to crane operators to replan safe paths in near real time. The proposed motion planning framework is designed in a way that makes it possible to be generalized over different types of equipment, and has the ability of visualizing and simulating motion planning results in near real-time. The framework is applied to develop a specialized motion planning system for construction equipment, Intelligent Construction Equipment Planner (ICE-Planner). This system is integrated into three-dimensional software to define, solve, and visualize motion planning in near real time. **DOI: 10.1061/(ASCE)CP.1943-5487.0000123.** *© 2012 American Society of Civil Engineers.*

CE Database subject headings: Motion; Simulation; Cranes; Construction management; Safety; Productivity.

Author keywords: Motion planning; Simulation; Cranes; Framework; Construction safety.

Introduction

A study about construction equipment 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). In this study, cranes have the most frequent fatalities associated with construction industry (17%), followed by excavators (15%), tractors (15%), loaders (9%), and pavers (7%) (NIOSH 2009).

Crane operation depends on not only the experience of the operator, but also sufficient and appropriate support in real time. For example, there are many blind spots for an operator, and most of the time the operator is 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 improve safety of crane operations and provide more awareness on site, a real-time replanning support is necessary to predict and avoid collisions based on the updated environment information.

Recent research aimed to enhance safety and productivity by proposing computational methods and simulation tools. Kang and Miranda (2009) developed a kinematic and dynamic model to generate an efficient and smooth simulation of cranes. This research is extended by proposing numerical methods to generate collision-free paths for erection activities (Kang et al. 2009). However, this research simplified the problem of multicrane planning by using a decoupled approach. Using this approach, the planner determines the paths of each individual crane independently and then tries to resolve possible conflicts by controlling their velocities. This approach is efficient but it is incomplete; i.e., it may fail to find a solution even if there is one, and it does not work in real time for dynamic environments.

Hornaday et al. (1993) proposed a system for computer-aided planning for heavy lifts. The proposed system suggested the utilization of planning algorithms for fulfilling the requirements of the system, but it did not develop a suitable algorithm or investigate available path-planning algorithms. Other researches focused on developing systems to simulate and visualize construction processes (Kamat and Martinez 2001), or to analyze and avoid collisions between equipment (Zhang et al. 2007). In the research of Chi et al. (2007), a prototype system was introduced for simulating and visualizing crane manipulation and cooperation. A dual-crane scenario was used to exemplify crane cooperation. This prototype did not consider obstacles in the environment and only focused on visualizing cranes' operations.

Cranimation and LiftPlanner are software systems developed for selecting suitable cranes and calculating the outrigger forces for mobile cranes, the distribution of ground pressures for crawling cranes, and the minimum and maximum radius ranges. These software systems are able to produce drawings to plan and document critical lifts. However, they focus on the engineering constraints of the crane and provide detailed selection and configuration of the crane. They require the users to manually plan the path for moving the object considering obstacles in the three-dimensional (3D) environment.

The research presented in this paper is part of a research program at Concordia University aiming to integrate multiagent systems, wireless communication, and field data capturing technologies to provide more awareness of dynamic construction site conditions, a safer and more efficient work site, and a more reliable decision support based on good communications (Zhang et al. 2009). The following visionary scenario is given to explain the assumptions of this research program and to put the work of the

[1] M.A.Sc. Student, Dept. of Building, Civil and Environmental Engineering, Concordia Univ., 1455 de Maisonneuve Blvd. W., Montreal PQ, Canada H3G 1M8. E-mail: hbahnassi@inframez.com

[2] Professor, Concordia Institute for Information Systems Engineering, Concordia Univ., 1455 de Maisonneuve Blvd. W., Montreal PQ, Canada H3G 1M8 (corresponding author). E-mail: hammad@ciise.concordia.ca

Note. This manuscript was submitted on May 18, 2010; approved on March 29, 2011; published online on December 15, 2011. Discussion period open until June 1, 2012; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 26, No. 1, January 1, 2012. ©ASCE, ISSN 0887-3801/2012/1-54–63/$25.00.

present paper in context. A construction site is monitored with a precise tracking technology such as ultra-wideband sensors (Ghavami et al. 2004). The workers, equipment, and components on the site are tagged with multiple tags that allow the tracking system to identify their poses in near real time with adequate accuracy so that these poses can be used to detect potential collisions. The paths of cranes and other types of equipment are planned in advance considering all the static obstacles on the construction site. During the execution of the motion plan, upon detecting a potential collision between a crane and a dynamic obstacle (e.g., another piece of equipment working in the same area that was not considered in the initial plan), the path of the crane can be recalculated in near real time.

To realize the previously discussed scenario and to fulfill crane operational tasks efficiently and safely in a complex environment with known and unknown obstacles, a framework is proposed in this paper for near real-time motion planning of cranes that satisfies safety requirements and efficiently considers the dynamic properties of construction sites. This framework is not aiming to full automation but rather for providing assistance to crane operators to replan safe paths in near real time. The objectives of this paper are (1) to provide the overall structure of the framework for motion planning of cranes; (2) to investigate safety requirements in the proposed framework; (3) to investigate the dynamic properties of multiequipment construction sites; and (4) to develop a prototype system that can be used to test the framework.

Framework for Near Real-Time Motion Planning

In this section, a framework is developed to integrate and simulate near real-time motion planning for cranes. The framework considers the dynamic nature of construction sites and engineering constraints and adds several safety aspects. Several methods are proposed to develop efficient motion planning for cranes and construction equipment in general. The general structure of the framework is shown in Fig. 1.

The solver component in this framework is the core unit for computing feasible motion paths (planning), and modifying existing ones based on environment updates (replanning). This component is based on algorithms that utilize continuous collision detection queries to search the configuration space (*C*-space) for optimized feasible paths. The *C*-space of a crane is the space of all possible configurations of the crane; and a configuration is simply a point in this abstract space. As a result, the dimensionality of the *C*-space is affected by the number of degrees-of-freedom of the crane (Choset et al. 2005). The input data to the framework includes construction, equipment definition (kinematic properties, engineering constraints, and geometrical representation) and environment definition (static and dynamic obstacles). It should be noted that the kinematic structure of a crane depends on the type of the crane (e.g., hydraulic crane or tower crane); however, the framework assumes that this structure is defined as an input. The solver is able then to generate a feasible path for each type of crane based on its defined kinematic structure.

The *C*-space is generated based on the crane information and environment definitions. Under the "Kinematics Properties," the number of DOFs of the equipment determines the dimensionality of the *C*-space, and the types of these DOFs specify its topology. More modifications to the *C*-space are done to accommodate the engineering constraints and dilation for ensuring realistic and safe motion paths for the construction equipment. In the "Environment Definition," the construction site environment is presented with static or dynamic obstacles. Static obstacles are those obstacles that represent the initial information about static objects in the construction site that are known in advance by the planner so they can be considered during the planning phase. Examples of these obstacles include buildings, electrical poles, etc. Three-dimensional models for these obstacles can be created using capturing hardware [e.g., 3D laser scanners (Shih 2002)] or can be modeled based on engineering and architectural designs. Dynamic obstacles are detected and updated while executing the initial plan. For example, in the case of multiequipment motion planning, higher priority equipment is considered as a dynamic obstacle for a lower priority one. With this definition, objects defined as dynamic obstacles have changes in their geometry because of their movement or simply because they have changes in their shape. Models of the building under construction should be updated based on the project progress monitoring. However, the update rate needed for the building under construction is much less than the update rate of equipment path planning. Thus, 3D models of buildings can be still considered as static obstacles updated between successful equipment tasks. Both static and dynamic obstacles are converted from their geometrical

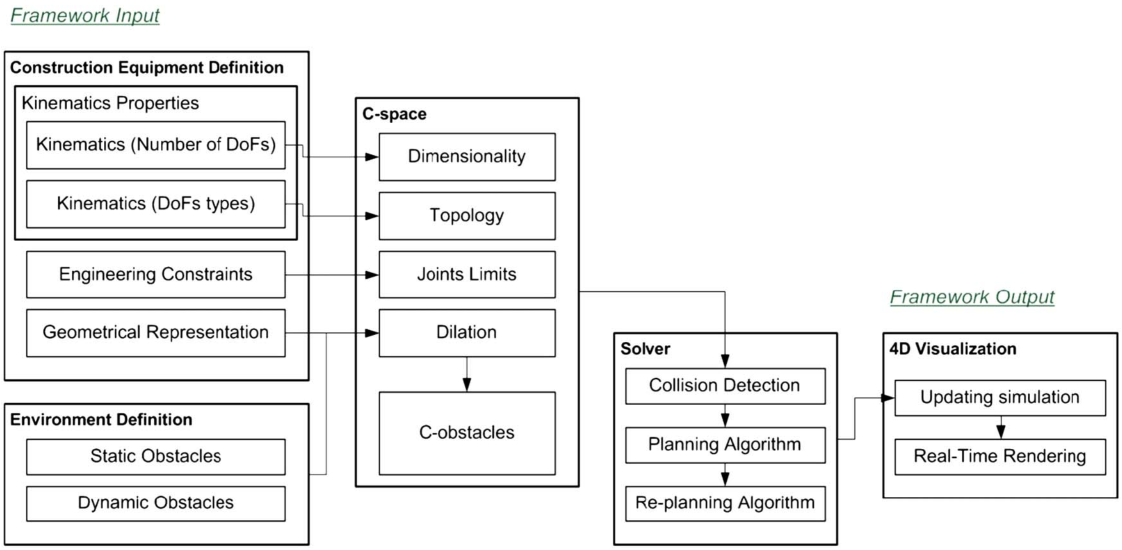


Fig. 1. Framework for construction equipment motion planning/replanning

representation in the work space to *C*-obstacles representation in the *C*-space. The result of this conversion completes the generation of the *C*-space as it will be used by the solver to compute the optimized path as explained in the "Solver" section.

The visualization component, including the calculated path of the crane, represents the framework output. This component is essential for viewing and analyzing the results of the solver in an interactive simulation environment. All results are rendered in a four-dimensional (4D) simulation environment where users can easily navigate in the 3D space using navigation tools. It is necessary for the visualization to be able to render in real time because it should reflect the dynamic updates and the replanning results as the solver modifies the motion paths. The high level of interactivity in the simulation environment enables the users to experiment with the dynamic nature of the construction site by interactively controlling the dynamic obstacles.

Fig. 2 shows the flowchart of the proposed near real-time motion planning framework. Fig. 2(a) starts by defining the motion planning problem (DEF subflowchart) that is presented in Fig. 2(b). The next step is planning the initial path based on the problem definition. The details for the planning process are shown in the PLN subflowchart in Fig. 2(c). In case the planning process succeeded, the crane starts updating its state based on the path for the first simulation step. If the planning failed to calculate a path, the simulation ends. Dynamic obstacles (e.g., equipment and workers) are scanned and detected after each simulation step so that they can be avoided by the replanning algorithm. The updated state is checked next against dynamic obstacles that are controlled interactively in the system. If the updated state is valid, it is rendered and the simulation advances to the next step. Before repeating the process for the next simulation step, the current step is checked to test if it is at the goal, and in this case the simulation ends. In case the current state is not valid because of dynamic obstacles, a replanning process is executed and checked for success, where it is either successful and visualized, or failed and simulation ends.

Fig. 2(b) shows an overview of the process of defining the motion planning problem in the proposed framework. It starts with two simplification processes for the crane definition. One is for simplifying the kinematic structure, while the other is for simplifying the geometrical representation to bounding boxes that are explained in the "Modeling" section. The next two processes are responsible for increasing the safety of the generated path as described in the "Safety Consideration" section. The first process generates critical volumes for preventing the solver from generating paths in unsafe volumes. The other process expands the obstacles defined in the environment by applying dilation to avoid unsafe paths that are too close to obstacles. The final step in the motion planning problem definition is the generation of the *C*-space that can be searched by the planning/replanning algorithm.

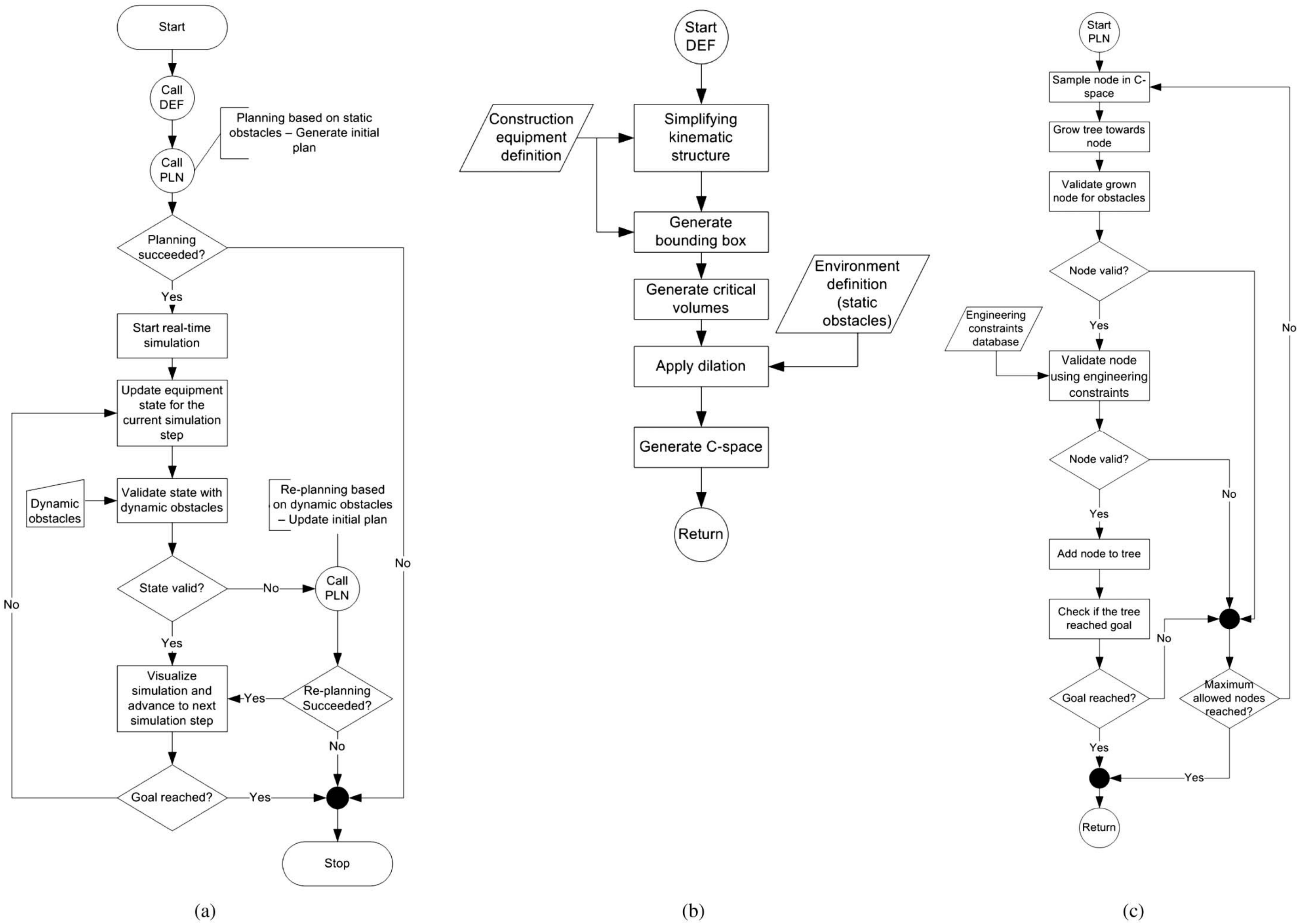

Start
Call DEF
Planning based on static obstacles – Generate initial plan
Call PLN
Planning succeeded?
Yes
Start real-time simulation
Update equipment state for the current simulation step
Dynamic obstacles
Validate state with dynamic obstacles
Re-planning based on dynamic obstacles – Update initial plan
State valid?
No
Call PLN
No
No
Yes
Visualize simulation and advance to next simulation step
Yes
Re-planning Succeeded?
No
Goal reached?
Yes
Stop
(a)
Start DEF
Construction equipment definition
Simplifying kinematic structure
Generate bounding box
Generate critical volumes
Environment definition (static obstacles)
Apply dilation
Generate C-space
Return
(b)
Start PLN
Sample node in C-space
Grow tree towards node
Validate grown node for obstacles
Node valid?
Yes
No
Engineering constraints database
Validate node using engineering constraints
Node valid?
No
Yes
Add node to tree
Check if the tree reached goal
No
Goal reached?
Maximum allowed nodes reached?
Yes
Yes
Return
(c)

Fig. 2. Flowcharts for proposed real-time motion planning framework: (a) general framework; (b) motion planning problem definition; (c) planning/replanning algorithm

Fig. 2(c) represents the general flow of the planning/replanning algorithm proposed in this framework. This algorithm is based on rapidly-exploring random trees (RRTs), which is a randomized sampling planning algorithm (LaValle 1998) as will be explained in the "Solver" section. It starts by sampling a random node in the C-space and growing a tree toward the sampled node. The resulting node is then validated in two phases for obstacles and for engineering constraints, respectively. If any of these two validations failed, a new node is sampled and the process is repeated. In the case of success, the node is added to the tree and the process is repeated as long as the grown tree has not reached the goal and the maximum allowed number of nodes has not been reached.

Modeling

A crane in this framework is considered as a robot, which is composed of a series of links (rigid bodies) connected by joints that allow relative motion of neighboring links. With this robotic model, defining the kinematic properties includes defining the hierarchal structure of the links, local coordinate systems (frames), and joint type, which is either a sliding joint (prismatic) or a rotational joint (revolute). These properties can be defined mathematically in a homogeneous transformation matrix using the Denavit-Hartenberg (DH)-notation (Denavit and Hartenberg 1955). The result is a computational model that is used to control the simulation model. Fig. 3 shows the kinematic modeling of a hydraulic crane where four degrees-of-freedom (DOFs) are defined, two revolute joints (swing of the boom θ_1 and angle to the ground θ_2), and two prismatic joints (boom extension d_3 and cable extension d_4). On each joint a local coordinate system is attached. This kinematic model includes an additional motion constraint for reorienting 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 cable's orientation; where in that case, the C-space dimension will become a five-dimension space and the configuration vector will be $q = (\theta_1, \theta_2, d_3, \theta_4, d_5)$ while the controllable DOFs are only (θ_1, θ_2, d_3, d_4). This leads to solve the motion planning problem as if it is a nonhonomic, which is not true because the cable motion constraint can be expressed as a configuration constraint as shown in Fig. 4

$$\theta_4 = 90° - \theta_2 \tag{1}$$

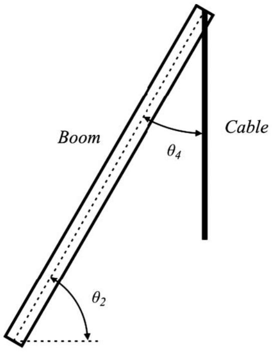
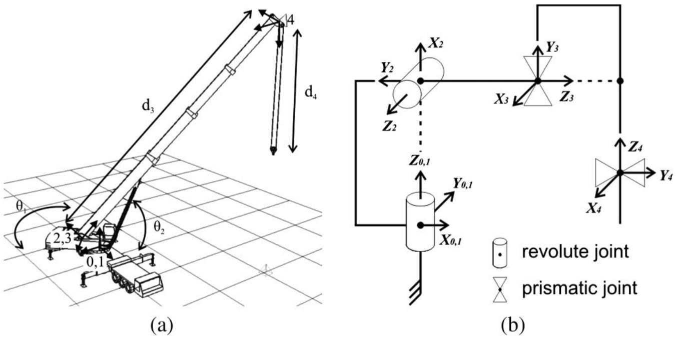


Fig. 4. Configuration constraint between boom rotation (θ_2) and cable rotation (θ_4)

Simplifying Simulated Kinematic Structure for Solver

As explained previously, a hydraulic crane has four DOFs where the boom extensions are considered as one DOF as introduced previously. However, this simplified structure is not used for simulation or control because in both the real and the simulation models there are four DOFs for the boom extension. Solving the problem of path-planning with full kinematic structure of the crane raises the complexity of the problem. This can be avoided by solving the problem using the simplified structure and then transferring the results to the full kinematic structure for simulation. Therefore, the planning results for the boom extension should be transferred from one prismatic joint to four joints of the full structure as shown in Fig. 5.



(a) (b)

Fig. 3. Defining kinematic structure for hydraulic crane: (a) frames attached to crane components; (b) schematic for hydraulic crane based on DH notation

Transferring the results requires deriving a heuristic relationship between the simplified and the full kinematic structure. This relationship should satisfy the rules of actions where the boom extends sequentially in the full structure starting from the base part of the boom. The following heuristic equation establishes this relationship

$$q_i = \max\left[\min\left(q_{i-\max}, q_{\text{simple}} - \left[q_{\text{simple}-\max} - \sum_{j=1}^{i} q_{j-\max}\right]\right), 0\right] \tag{2}$$

where q_i = extension value of the prismatic joint number i (where i starts from the tip of the boom); $q_{i-\max}$ = maximum value the prismatic joint i can have; q_{simple} = current value of the joint in the simplified structure; and $q_{\text{simple}-\max}$ = maximum range of joint limit in the simplified structure. This value should equal the sum of maximum ranges of all joints in the original structure (i.e., $q_{\text{simple}-\max} = \Sigma q_{i-\max}$).

Bounding Boxes for Optimizing Collision Detection Calculations

Having accurate and detailed models for visualization can negatively affect the collision detection computation time because more

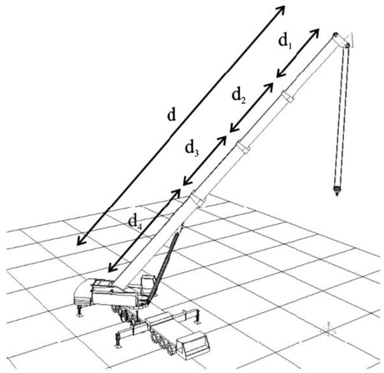


Fig. 5. Transferring planning results from one prismatic joint to four joints

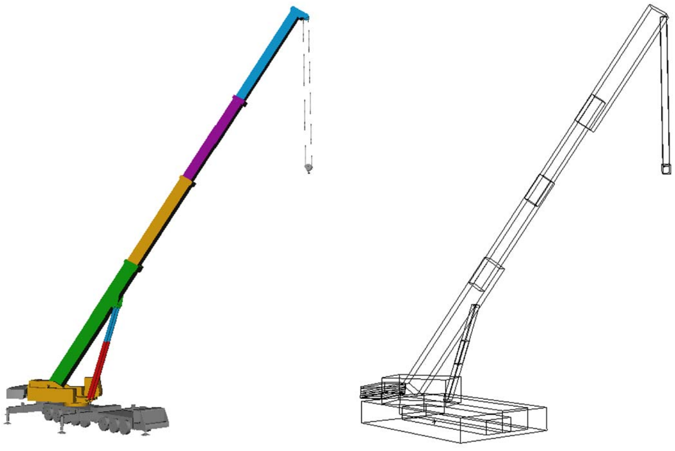


Fig. 6. Two versions of the same crane: (a) 3D model of hydraulic crane; (b) bounding boxes for collision detection

details mean more triangles and more time to perform collision detection queries. To enhance the quality of the visualized models while reducing the time of collision detection calculations, low-resolution proxy shapes are generated from the high-resolution models (Akenine-Moller et al. 2008). Usually these proxy shapes are bounding boxes that are attached to each element's center. Fig. 6(b) shows a low-resolution proxy model generated for the hydraulic crane shown in Fig. 6(a).

Engineering Constraints

In the proposed framework, engineering constraints should be considered through all motion planning steps in order to guarantee safe paths in terms of stability of the equipment along the entire length of the path. This requirement leads to a planning process that is interwoven with engineering constraints validation. While the solver is taking decisions for the next step of the path it is generating, engineering constraints are validated and included as additional decision factors for the solver.

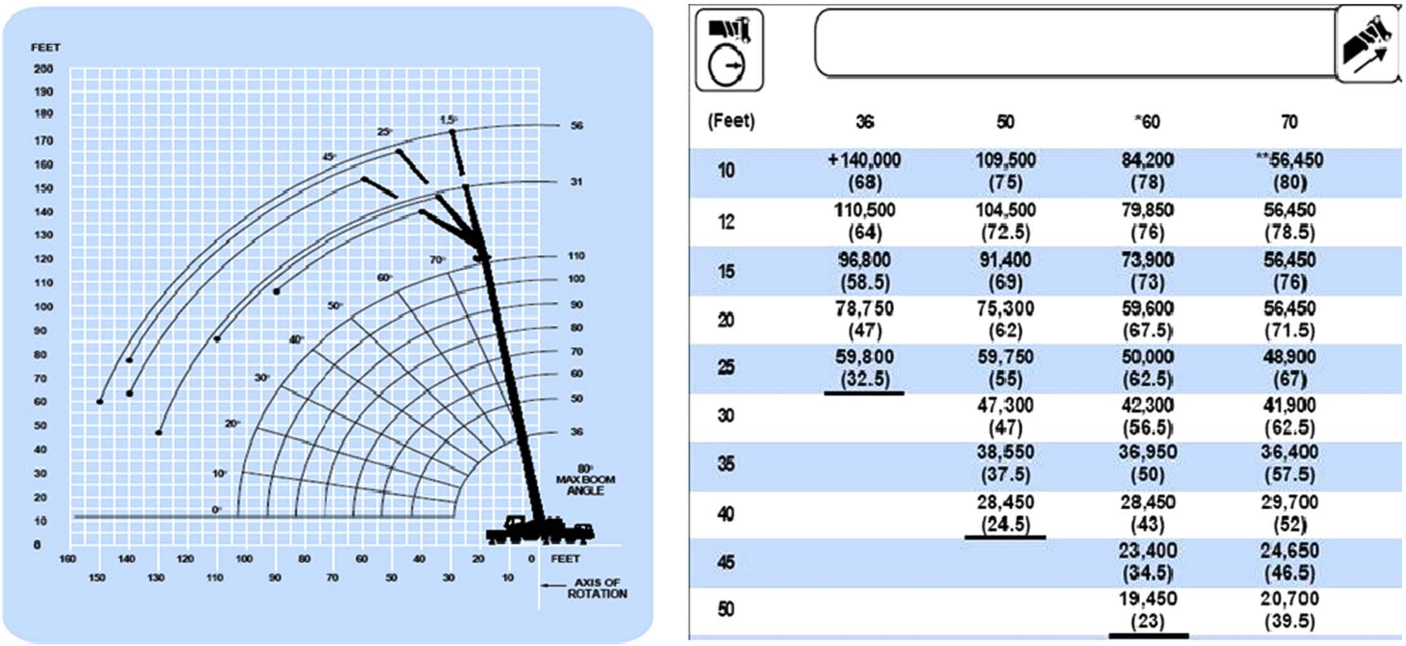
The engineering constraints of cranes are mainly from working ranges and load charts. Working ranges show the minimum and maximum boom angle according to the length of the boom and the counterweight [Fig. 7(a)]. Load charts give the lifting capacity based on the boom length, boom angle to the ground and the counterweight [Fig. 7(b)]. The data of the load charts are stored in a database that can be accessed when generating paths.

Validating engineering constraints is based on a search algorithm that is developed to validate solver queries by searching for the loading case that satisfies the query in the specific data cluster. If the algorithm finds this loading case, a decision of acceptance is returned to the solver for the stated query; otherwise, it will reject the query and the solver has to try a different path that satisfies the engineering constraints. It should be noted that engineering constraints are dependent on the configuration of the crane (e.g., boom extension and boom angle to the ground) and these parameters have different values for each query the planner does. For that reason, it is not possible to consider engineering constraints as a preprocessing step.

Safety Considerations

Critical Volumes

Certain volumes in the construction space are considered unsafe if the path of a crane intersects them. Fig. 8 shows an example of such volumes where the path of one crane goes through the space between the boom and the cable of another crane. Such cases are not desirable for safety issues and it is better to avoid generating paths in such volumes (Fari 1998; British Standard 2000). In the framework, these volumes are called critical volumes. The proposed approach to avoid generating paths that intersect with these volumes



(a)

(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)

(b)

Fig. 7. (a) Working range of crane; (b) load chart of crane (reprinted with permission from Grove Crane 2008)

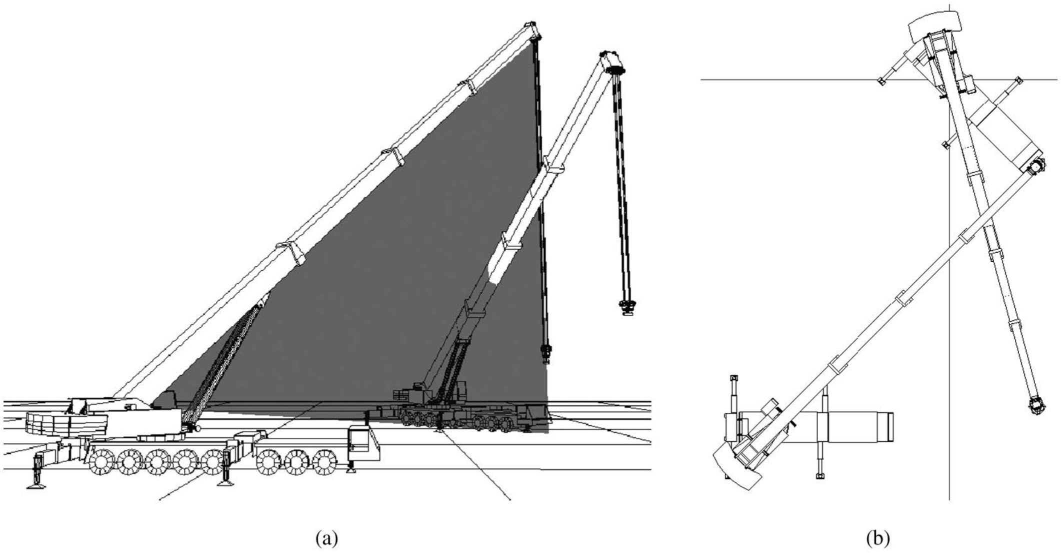


Fig. 8. The path of one crane goes through critical volume that is bounded under boom and cable of other crane: (a) perspective view; (b) top view

is to construct 3D surfaces from the points that define the critical volume's extents. These surfaces should be recomputed based on the spatial state of the equipment and used as obstacles to prevent the solver from generating paths that go through critical volumes.

Dilation

Increasing the safety of the generated paths can be accomplished by avoiding semifree paths. A path is considered semifree if the equipment touches obstacles without overlapping (Latombe 1991). Dilation is applied as a uniform extension around obstacles to avoid having semifree paths. Dilation in constrained spaces can sometimes waste good paths or can return no paths at all even if there was one. This is because when applying dilation around narrow parts in the space, some paths will be eliminated. This behavior is acceptable in most construction applications because it is preferable to have no feasible path than unsafe paths that go between tight obstacles.

Solver

The solver is the main part of the framework and it is used for motion planning and replanning.

Planning Phase

The planning algorithm that is used in this research is based on a random sampling algorithm. This type of planning algorithm attempts to solve a query as fast as possible and does not focus on the exploration of the entire search space. Sampling-based methods employ a variety of strategies for generating samples (collision-free configurations of the robot) and for connecting the samples with paths to obtain solutions to path-planning problems. RRT (LaValle 1998) demonstrated the tremendous potential of sampling-based methods. The key idea of this algorithm is to search the space by incrementally growing a tree from the initial configuration until the tree reaches the goal configuration. RRT is efficient in dealing with robots with many DOFs and with many different constraints. Despite its efficiency, RRT has many versions that can suite different planning problems. However, paths generated by RRT are generally guaranteed to be near optimal based on predefined criteria (LaValle 2006).

Several extensions to the basic RRT algorithm are required to improve the speed of the search and make it suitable for construction equipment planning. A new algorithm called RRT Biased LimCon (RRT-biased limited greedy connect) that considers efficiency, optimality, and applicability for cranes is developed. This algorithm is utilized as the core planner in this framework. The proposed algorithm utilizes the Connect Connect function to grow the tree with minimum number of nodes thus resulting in an efficient calculation time and applicable paths with minimum number of actions. Additionally, the Connect function is modified to consider the engineering constraints and the rules of action to ensure safety and applicability of the generated paths for cranes. Optimality is considered in the proposed algorithm by implementing probabilistic biasing towards the goal and by modifying the Connect function to be greedy towards the goal instead of the sampled point. These modifications generally prevent having redundant and/or reversing movements. However, when avoiding dynamic obstacles (e.g., another crane) the reverse motion may happen in order to allow this movement by setting a low value for the goal biasing probability to increase the chances of avoiding dynamic obstacles. More information about the proposed algorithm including computational details, comparisons, and tests are introduced in the accompanying paper (AlBahnassi and Hammad, unpublished data, 2010).

Replanning Phase

In the case of construction sites, the initial information concerning the environment is incomplete and the environment itself is dynamic. Therefore, rapid replanning for cranes is essential to repair motion paths and cope with environment updates. Applying RRT for replanning can be time consuming because it abandons the initial tree and grows a new tree from scratch each time the environment updates. For this reason, a replanning algorithm is investigated for repairing the tree when new information concerning the configuration space is received. The concept of this algorithm was first presented in (Ferguson 2006) as dynamic rapidly-exploring random trees (DRRT). DRRT depends on efficiently removing the invalid parts and growing the remaining tree until a new solution is found. This algorithm has been modified in this framework to fit the proposed RRTBiasedLimCon algorithm (AlBahnassi and Hammad, unpublished data, 2010). Because the process for detecting obstacles and updating the path plan is done intensively (i.e., every simulation step), the motion planning algorithm is extended to plan efficiently in dynamic environments and for multi-equipment scenarios by utilizing the DRRT concept.

Multiequipment Planning

The proposed replanning algorithm can be applied in a prioritized approach for efficient multiequipment motion planning. Using this approach, each of the cranes is assigned a priority. Next, the cranes are picked in order of decreasing priority. For each picked crane a path is planned, avoiding collisions with the static obstacles as well as the previously picked cranes, which are considered as dynamic obstacles. This approach is tested with two different case studies as will be shown in the "Implementation and Case Studies" section. Defining the priority for each crane is considered in this framework as part of the user inputs where the site manager is responsible for defining these priorities based on safety and productivity considerations (e.g., the crane that is lifting elements for tasks on the critical path of the project should have the highest priority).

Visualization

To achieve better understanding of planning and replanning processes, the results of 4D simulations have to be visualized while considering real-time rendering and interactivity.

Implementation and Case Studies

The presented framework was used to develop a prototype system called ICE-Planner, which stands for Intelligent Construction Equipment Planner.

The proposed prototype is composed of several components. Each of these components is responsible for a specific task, which provides a flexible system for modeling, solving, and visualizing real-time motion planning problems interactively. Fig. 9 shows the different components that form the system and their relationships. The components in the figure are separated into three blocks based on their tasks. The first one is responsible for defining the motion planning problem using the modeling tools found in the 3D software package. The middle block is the main part for solving both the planning and replanning problems based on the modeled environment in the first block. The last block is responsible for visualizing the results using the rendering capabilities in the 3D software for this task.

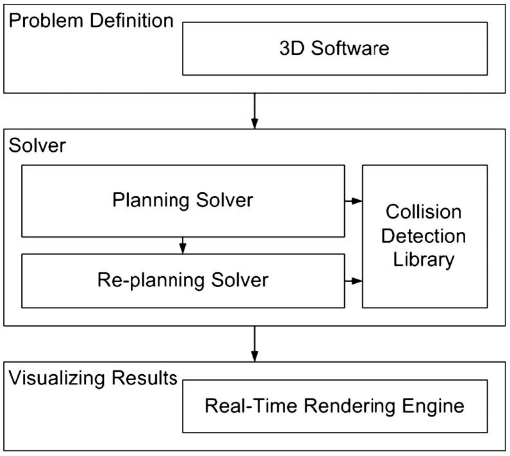


Fig. 9. Main system components and their relationships

To save effort and time, certain libraries and a 3D software package are adopted to develop ICE-Planner. Autodesk Softimage is 3D software for visual effects and game production. Softimage is used in the system as the main environment for creating and defining the motion planning problem in addition to interacting and rendering the results in real time using OpenGL (OpenGL 2010) application programming interface (API). All 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. In the solver component, a C++ library called the motion strategy library (MSL) (Motion Strategy Library 2009) was used for solving planning queries in the system. This library provides a wide range of randomized motion planning algorithms including RRTs. This library has been modified and extended to fit with ICE-Planner. Modifications are mainly for updating the code to be compatible with the Softimage C++ API. The library extensions add new classes 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. More classes are added to the library for outputting the planning results as a 4D simulation in Softimage and utilizing its OpenGL renderer to visualize the results in real time. Additional extensions to the library are made to create a new planning algorithm based on the proposed enhancements in (AlBahnassi and Hammad, unpublished data, 2010). Along with MSL, the proximity query package (PQP) (2009) was used for performing collision detection queries on obstacles found in the environment. The PQP library has been also modified and extended to be integrated seamlessly in ICE-Planner. In addition to the code modifications for compatibility issues, additional classes are implemented to access the tessellated geometry directly from the Softimage scene at every simulation step. In ICE-Planner, each simulation step is defined as one frame and it is assumed that each second is composed of 30 frames. This assumption provides enough accuracy for the PQP library to detect dynamic obstacles in construction sites.

The replanning solver was developed from scratch based on the DRRT algorithm with the proposed modifications and enhancements introduced in (AlBahnassi and Hammad, unpublished data, 2010). The replanning solver depends also on the PQP library to perform collision checks.

The implemented framework is evaluated using a case study involving two hydraulic cranes each with four DOFs. In addition, following the same procedure discussed for hydraulic cranes, another case study is done involving two tower cranes each with three DOFs. The environment setting contains a steel frame structure that is composed of 536 members. In each case study, one of the two cranes is considered as a dynamic obstacle moving in the construction site near the other crane for which the path planning is done.

Validating Results of Case Studies

Fig. 10 shows the environment in addition to the hydraulic cranes. The right-side crane (Crane-1) is considered for the path planning based on the initial and goal configurations. The left-side crane (Crane-2) is considered as a dynamic obstacle that can be preplanned as a high priority crane or manipulated interactively while the simulation is running. The steel frame structure is considered as a static obstacle. In the captured snapshots, Crane-1 starts executing the path in order to move from its initial configuration to its goal configuration, both of which are shown in simulation step 1. At simulation step 120, the system detects Crane-2 as a dynamic

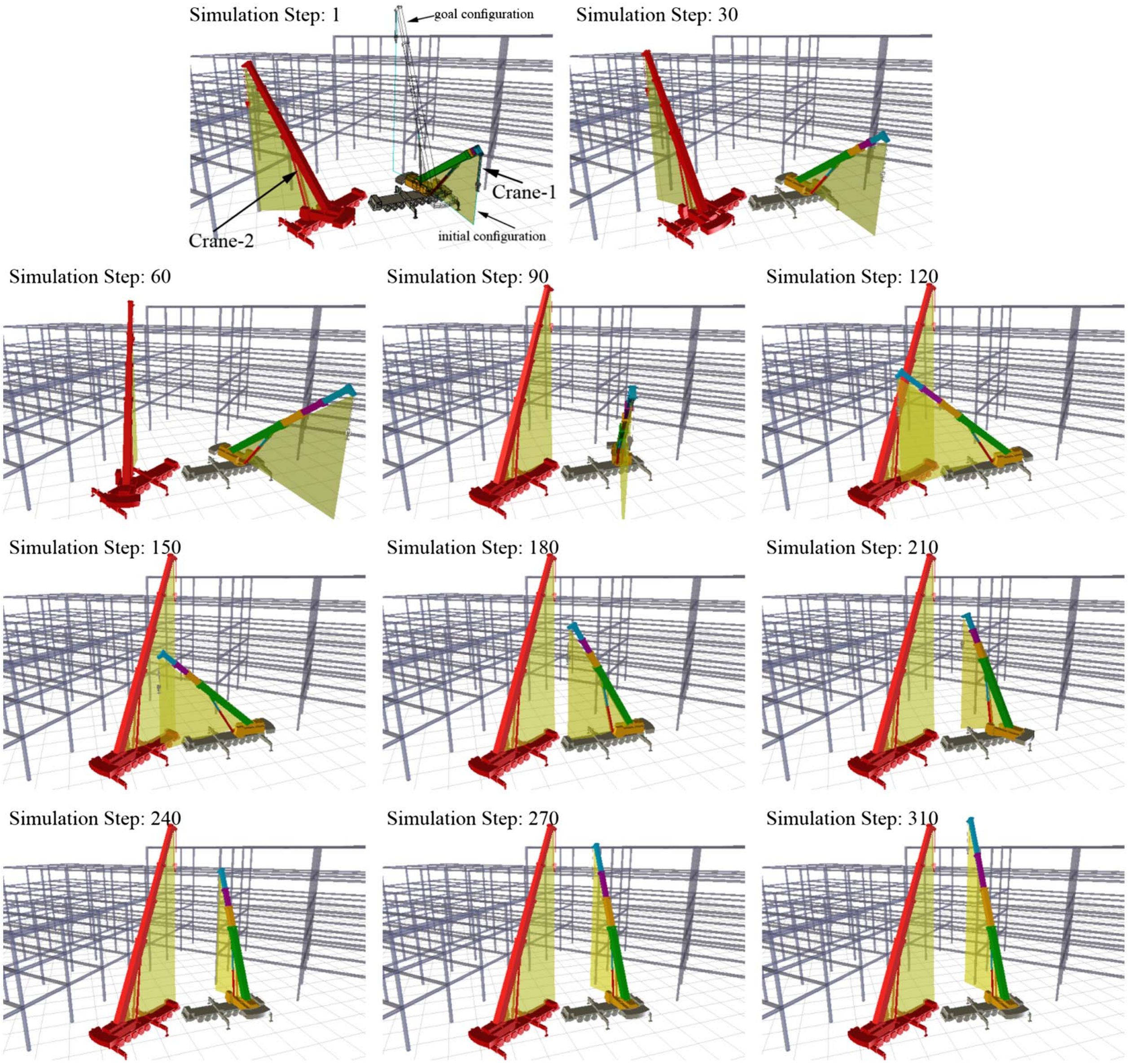


Fig. 10. Simulation snapshots for hydraulic crane case study

obstacle and a replanning phase of Crane-1 is initiated. The replanning updates the original path of Crane-1 to avoid the other crane and continues executing the path until it reaches the goal at simulation step 240.

Fig. 11 shows the same environment applied to the tower crane case study where the left-side crane (Crane-2) is the dynamic obstacle for the right one (Crane-1). At simulation step 1, Crane-1 starts executing the path towards the goal shown at this step. As Crane-1 advances towards the goal, the system detects a collision with the other crane at simulation step 150. The replanning phase is then executed and the path is updated. The new path drives Crane-1 to rotate in the other direction in order to reach the goal as shown in simulation steps (150–400).

Evaluating Results of Case Studies

Each test was repeated 50 times with different random seeds to evaluate the results with different random trees. Table 1 and 2 show the results summary for both planning and replanning for the two case studies.

The reported times for the performance are based on the CPU time for an Intel T5550 Core2Duo processor (1.83 GHz). The planning time is the CPU time required to compute the initial path from the initial location of the crane to its goal. The replanning time is the total CPU time for all replanning operations needed to move the crane from the initial location to the goal while avoiding dynamic obstacles.

As shown in Table 1, the duration for calculating the initial path for the hydraulic crane was nearly five times the duration for the tower crane. Two main factors are responsible for the difference in planning time between the two cases. The first one is dimensionality, where in the case of the hydraulic crane the solver has to solve a 4-DOF robot while for the tower crane it is only 3-DOF robot. The second factor is the motion constraints evaluation that is applied in the hydraulic crane case study and not applied in the tower crane case study. These constraints include several mathematical expressions for simplifying the boom structure as explained in the "Modeling" section in addition to the motion constraint for

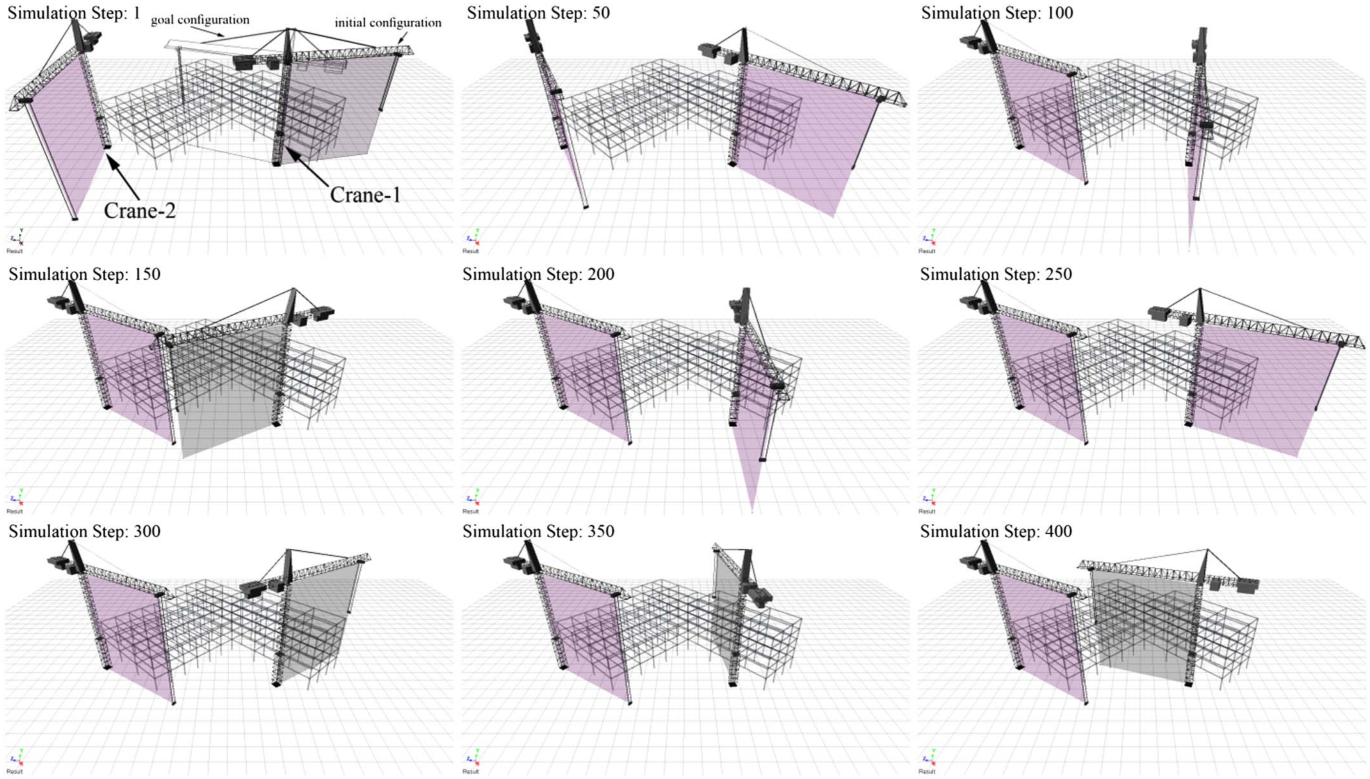


Fig. 11. Simulation snapshots for tower crane case study

controlling the direction of the cable to follow the gravity vector. The numbers of successful nodes for the hydraulic crane case study and the tower crane case study are 85.60 and 64.98 nodes, respectively. The hydraulic crane uses more nodes to calculate the path because of the extra dimension of the *C*-space it has. In both cases, the results were all applicable paths that can be executed by a crane operator or by an autonomous robotic crane. Table 2 shows acceptable results for the re-planning that can be improved by using a more powerful CPU. The number of nodes on the path to drive the crane from its current location (location where the dynamic obstacle is detected) to the goal is small and applicable for cranes (4.6 nodes for the hydraulic crane and 4.13 nodes for the tower crane).

Table 1. Results Summary for Planning Case Studies

	Hydraulic crane	Tower crane
Average for planning time (s)	5.256	1.071
Standard deviation for the planning time (s)	3.963	1.358
Average number of successful nodes	85.600	64.980
Standard deviation for the number of successful nodes	74.421	112.284
Average number of nodes on path	8.100	8.420
Standard deviation for the number of nodes on path	2.685	4.155

Table 2. Results Summary for Replanning Case Studies

	Hydraulic crane	Tower crane
Average for replanning time (s)	1.736	0.035
Standard deviation for the replanning time (s)	3.831	0.819
Average number of successful nodes	24.238	19.22
Standard deviation for the number of successful nodes	21.253	33.56
Average number of nodes on path	4.600	4.130
Standard deviation for the number of nodes on path	1.546	1.966

Conclusions and Future Work

A framework for construction equipment motion planning and replanning was discussed in this paper. This framework automatically generates crane paths in the planning phase and alternative paths in the execution phase if a potential collision is detected. This automation results in improving the safety of crane operations and in eliminating the need for manual path planning The advantage of the proposed method over previous work in that it presents a unified framework that can be used for efficiently planning multicranes in near real time in dynamic environments. The framework is efficient because it does not solve the problem as a composite planning problem for several cranes. Instead it solves it as a single-crane problem in a dynamic environment where other cranes are considered as dynamic obstacles for the current crane. Therefore, the proposed approach goes beyond previous research and it is more applicable to construction sites. This framework extends the previous research of crane path planning by (1) introducing several methods for satisfying safety requirements of cranes, such as the critical volumes, dilation around obstacles, and engineering constraints from the working ranges and load charts of cranes; (2) considering the dynamic properties of construction sites efficiently by proposing a replanning algorithm, which is able to provide information for operators in near real time to assist them in manipulating equipment more efficiently and safely; (3) considering multiequipment planning by integrating the replanning algorithm in a prioritized motion planning approach; and (4) developing several techniques to enhance the efficiency of the prototype system and to enable near real time visualizing and simulation. The developed prototype system was used to evaluate and validate the framework intensively in two case studies. The results showed good indicators for the applicability of the system in near real-time motion planning for construction equipment by integration with suitable hardware and tracking sensors that are able to provide accurate information in near real time.

Although this framework is engineered to solve general cases of heavy construction equipment such as concrete boom pumps, currently it is validated only for hydraulic and tower cranes. Further testing and evaluation with other types of construction equipments is required to prove its general applicability. Future research will consider different types of physical motions (e.g., the effect of the wind on the lift) and the deformation of the boom of the crane.

References

Akenine-Moller, T., Haines, E., and Hoffman, N. (2008). *Real-time rendering*, 3rd Ed., A.K. Peters, Wellesley, MA.

British Standards. (2000). "Code of practice for safe use of cranes—Mobile cranes." *BS 7121-13*, London.

Chi, H. L., Hung, W. H., and Kang, S. C. (2007). "A physics based simulation for crane manipulation and cooperation, computing in civil engineering 2007." *Proc., 2007 ASCE Int. Workshop on Computing in Civil Engineering*, ASCE, Reston, VA, 777–784.

Choset, H. et al. (2005). *Principles of robot motion—Theory, algorithms, and implementations*, MIT Press, Cambridge, MA.

Cranimation [Computer software]. CraniMAX GmbH, Zweibrücken, Germany. ⟨http://www.cranimax.com⟩ (Dec. 2009).

Denavit, J., and Hartenberg, R. S. (1955). "A kinematic notation for lower-pair mechanism based on matrices." *Trans. ASME*, 23, 215–221.

Fari, H. W. (1998). *Crane safety on construction sites (manual of practice No. 93)*, *Task Committee on Crane Safety on Construction Sites*, ASCE, Reston, VA.

Ferguson, D., Kalra, N., and Stentz, A. (2006). "Replanning with RRTs." *Proc., IEEE Int. Conf. on Robotics and Automation*, IEEE, Piscataway, NJ, 1243–1248.

Ghavami, M., Michael, L. B., and Kohno, R. (2004). *Ultra wideband signals and systems in communication engineering*, Wiley, Hoboken, NJ.

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

Hornaday, W. C., Haas, C. T., O'Connor, J. T., and Wen, J. (1993). "Computer-aided planning for heavy lifts." *J. Constr. Eng. Manage.*, 119(3), 498–515.

Kamat, V. R., and Martinez, J. C. (2001). "Visual simulated construction operations in 3D." *J. Comput. Civil Eng.*, 15(4), 329–337.

Kang, S. C., Chi, H. L., and Miranda, E. (2009). "Three-dimensional simulation and visualization of crane assisted construction erection processes." *J. Comput. Civil Eng.*, 23(6), 363–371.

Kang, S. C., and Miranda, E. (2009). "Numerical methods to simulate and visualize detailed crane activities." *Comput. Aided Civ. Infrastruct. Eng.*, 24(3), 169–185.

Latombe, J. C. (1991). *Robot motion planning*. Kluwer, Boston.

LaValle, S. M. (1998). "Rapidly-exploring random trees: A new tool for path planning." *TR 98-11*, Computer Science Dept., Iowa State Univ., Ames, IA.

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

LiftPlanner [Computer software]. LiftPlanner Software, Burnsville, MI. ⟨http://www.liftplanner.net⟩ (Dec. 2009).

Motion Strategy Library (MSL) [Computer software]. Univ. of Illinois, Champaign, IL. ⟨http://msl.cs.uiuc.edu/msl/⟩ (Dec. 2009).

NIOSH. (2009). "The National Institute for Occupational Safety and Health." Centers for Disease Control and Prevention, Atlanta. ⟨http://www.cdc.gov/NIOSH/⟩ (Dec. 2009).

OpenGL. (2010). ⟨http://www.opengl.org⟩ (Jan. 2010).

Proximity Query Package (PQP). (2009). ⟨http://www.cs.unc.edu/~geom/SSV/⟩ (Dec. 2009).

Shih, N. J. (2002). "An application of a 3D scanner in the representation of building construction site." *Proc., 19th Int. Symp. on Automation and Robotics in Construction (ISARC 19)*, Int. Association of Automation and Robotics in Construction, 337–342.

Softimage [Computer software]. Autodesk, Inc., San Rafael, CA. ⟨http://www.softimage.com⟩ (Jan. 2010).

Zhang, C., Hammad, A., and Bahnassi, H. (2009). "Collaborative multi-agent systems for construction equipment based on real-time field data capturing." *Electron. J. Inform. Tech. Construct.*, 14, 204–228.

Zhang, C., Hammad, A., Zayed, T. M., Wainer, G., and Pang, H. (2007). "Cell-based representation and analysis of spatial resources in construction simulation." *J. Autom. Constr.*, 16(4), 436–448.