Automated Simulation of the Erection Activities in Virtual Construction

ShihChung Kang (jessy@stanford.edu), Eduardo Miranda (emiranda@stanford.edu), Stanford University

1 Introduction and Motivation

The goal of the research is the development of a computer system to plan, simulate and visualize erection processes in construction. In the research construction cranes are treated as robots with predefined degrees of freedom and crane-specific motion planning techniques are developed to generate time-optimized and collision-free paths for each piece to be erected in the project. Using inverse kinematics and structural dynamics simulation, the computer system then computes the crane motions and velocities necessary to achieve the previously calculated paths. The main benefits of the research are the accurate planning and scheduling of crane operations leading to optimization of crane usage and project schedules, as well as improving overall crane safety in the project. This research is aimed at the development of systems that will allow computer-assisted erection of civil infrastructure and ultimately to achieve fully-automated erection processes using robotic cranes.

Cranes are one of the most important and heavily used resources in a construction site. Previous research has highlighted the central role that cranes have on the control and pace of construction operations (Gray, 1983). In the case of medium and high rise buildings, cranes are the most important equipment resources at a site as most of the material to be placed in the buildings is transported using cranes. Hence, an inefficient use of this resource will have a direct effect on the erection schedule and on the overall construction schedule. Accuracy in estimating activity duration is one of the key prerequisites for successful construction planning and in ensuring the completion of a project on time. Despite the importance of crane lifting operations in estimating speeds of erection, in current practice, crane planning and erection schedules is primarily done using very rough estimates of production rates such as average number of tons erected per day or average number of pieces erected per day. While these production rates provide some rough estimates of the speed of erection processes, they are prone to large errors due to project specific variations. For example, the type and weight of pieces being erected can change significantly from one project to another, from one type of element being lifted to another (e.g. weight of a column versus the weight of a filler beam), or can even have large variations for the same type of element as it changes from one location of the project to another (e.g. weight of columns in bottom floors versus the weight of columns in top floors). While erection schedules based on number of pieces are, in general, better than those based on weight, they also are prone to significant variations. On one hand in current practice, seldom there is an actual count of the number of pieces to be erected on a project. On the other, the hoisting times of particular pieces can have very large variations with respect to the average hoisting times that are used in the rough erection schedules. For example, pieces to be erected in the second floor will have significantly shorter hoisting times than those located in the 30th floor of a building. A computer system that models the actual time involved in the transportation of each of the pieces being erected could yield significantly more accurate erection plans and erection schedules than those currently in use today.

Planning of the optimal path for each piece that needs to be erected can be a very complicated and very time consuming process if done manually. However, computers can greatly assist in the execution of many paths. Given an origin, destiny and special constraints such as the state of construction of the structure being erected, position of cranes, power lines, etc the system will design a collision-free path that minimizes the time required to move the piece. Using inverse kinematics the system will then compute the motions required by each degree of freedom of the crane in order to follow the trajectory. Once an algorithm is developed to select the optimal and safe path for each piece, the computer can repeat the process for each piece to be erected in the project. The research will take advantage of recent advances in motion planning techniques combined with powerful computational capacities of today's computers. Furthermore, the research will make use of recent advances in computer graphic technology to provide realistic visualizations that will enable crane operators, construction managers and subcontractors to visualize erection operations in the computer before they take place.

The research will result in increased productivity of erection processes by optimizing the time required to transport each piece in the project and in the case of multiple cranes by minimizing crane waiting times. By providing detailed piece-by-piece planning, the system will provide more accurate erection schedules that will allow a better coordination of erection construction operations with other construction activities at a site. Furthermore, the research will increase safety of crane operations by providing the design of crane motions that lead to collision free paths and by minimizing vibrations of elements being erected that could hit construction personnel. Finally, the research will provide excellent visualization capabilities that will allow seeing in great level of detail individual lifts or complete erection processes in the computer before they take place in the construction site. Such a tool can also provide a crane simulator for training of crane operators.

2 Previous Research

One of the most studied aspects of crane is the optimum location of cranes. For example, Rodriguez-Ramos and Francis (1983) developed a mathematical model to establish the optimal location of a single tower crane within a construction site. A simple graphical procedure was developed for locating the best position of the crane which has minimal total transportation cost between the crane and construction supportive facilities that are serviced by the crane. The mathematical model considered radial and angular movement of a horizontal tower crane, and derived the following equation to represent the total transportation cost.

$$F(\theta, r) = \sum_{j=1}^{n} W_j \left(\frac{A_j(\theta)}{V_a} + \frac{|r - r_j|}{V_a} \right) = F_1(\theta) + F_2(r) \qquad (1)$$

where n is the number of pieces to be erected located angle θ_j and radius r_j, θ, and r define the location of the hook of the crane and W_j is defined as the (nonnegative) transportation cost weight factor, equal to the cost per unit angular or radial travel time multiplied by the estimated number of trips or cycles made in a certain given time period between the crane's unknown location and existing construction supportive facility j, $A_j(\theta)$ is the angular movement of the boom, V_a is angular velocity of the trolley (rad/sec) and V_R (m/sec or ft/sec) radial velocity.

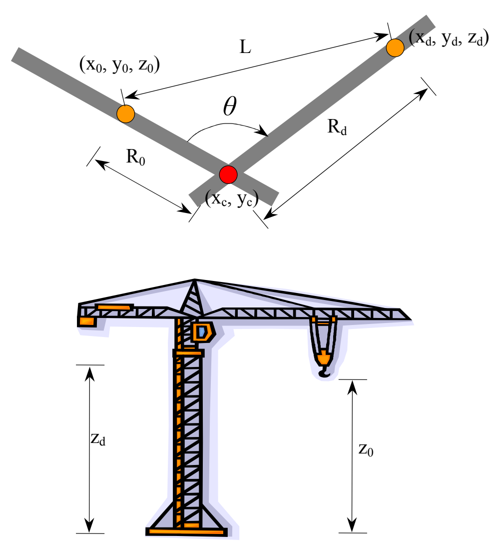


Fig 1. Horizontal tower crane model

Equation (1) above cannot describe two of important operations in a tower crane: 1) vertical motion; 2) simultaneous movement (angular and radial). It assumes that one motion occurs at a time. However, vertical movement accounts for about 45% to 80% in a hoisting time (Anson and Wang 1994), and most crane operators always move the trolley in angular and radial direction simultaneously.

Furusaka and Gray (1984) also developed a mathematical model to determine the optimal location for a tower crane aiming at minimizing the total crane transportation time. However their model required as input the number of days that each crane is used. Gray and Little (1985) developed a systematic approach to the selection of an appropriate crane for a construction site. They described the process and criteria for the selection of two categories of crane, namely, tower cranes and mobile cranes. The selection process was in the form of decision flow charts. A computer-based expert system was developed and used to simplify the selection process. Subsequently, Choi and Harris (1989) adopted the basic mathematical expressions of Rodriguez-Ramos and Francis for computing the angular and radial movement. However, they also considered that the angular and radial movements were carried out simultaneously with the hoisting movement.

More recently Zhang et al. (1999) improved the mathematical model which considered both horizontal and vertical motions of a tower crane. By analyzing the coordinates of starting and destination points, their model found distance to be traveled for hook hoisting, trolley rotation, and trolley radial movement. Traveling time of each motion was obtained by dividing the distances in each degree of freedom by an average velocity. Furthermore, the possibility of simultaneous motions was simulated by introducing two random variables, α and β. The first random variable α represents the degree of coordination of hook movement in radial and tangential directions in the horizontal plan; and β reflects those in the vertical and horizontal plane. The travel time was computed as a linear combination of horizontal and travel times as $T = max(T_h, T_v)+\beta\ max(T_h, T_v)$ where T is the total erection time T, $T_{v,}$ is the vertical travel time, and T_h is the hook horizontal travel time. Similarly the horizontal travel time was computed as a linear combination of the radial and tangential motions as $T_h = max(T_a, T_v)+\alpha\ max(T_a, T_v)$ where T_r is the travel time for trolley radial movement and T_a is the tangent travel time. Radial, tangential and vertical travel times were computed assuming constant velocities using $T_v=|z_d-z_0|/v_h$, $T_r=|R_d-R_0|/v_r$, and $T_a=\theta/v_a$, where v_h, is the hoisting velocity of the hook, v_r is the trolley radial velocity, and v_a if the jib angular velocity. Referring to figure 4, to move the hook from (x_0, y_0, z_0) to (x_d, y_d, z_d), and center of the tower located at (x_c, y_c) the distances R_0 and R_d required to compute the trolley travel time are computed as $R_0 = \sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$ and $R_d = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$.

In the model used by Zhang et al. (1999) and Tam et al. (2001) there are two extreme situations for α: simultaneous tangential and radial movement occurs when α = 0, and consecutive movement occurs when α = 1. For β, there are also two extreme situations, simultaneous movement occurs in horizontal and vertical planes when β=0, and consecutive movement occurs when β=1. They indicated that α and β varied depending on the skill of operator and the spaciousness of the jobsite. They indicated that ideally these parameters need to be calibrated by observed data from construction sites. However, they assumed values of α=0.25 and β =1 (that vertical motion never occurs simultaneously with horizontal motions).

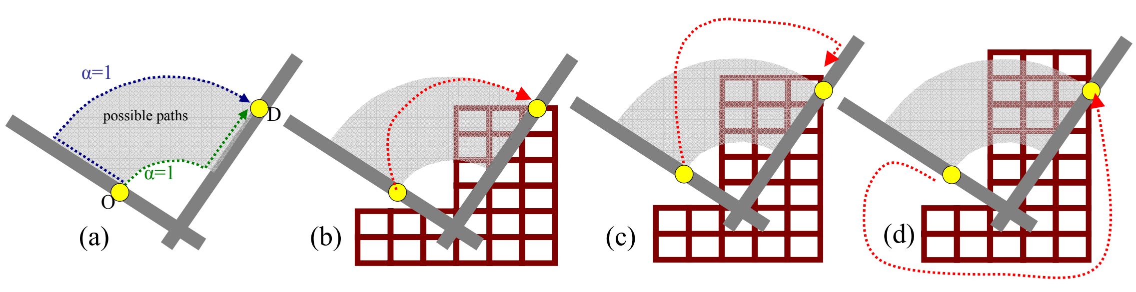


Fig 2. Possible horizontal trajectories in tower crane lifting operations.

While the model used by Zhang et al. (1999) and Tam et al. (2001) have provided improved estimates of hoisting times, they primarily rely on estimate of random variables α and β. More importantly they do not take into account the actual path or trajectory followed in each lifting operation. There are many possible paths than are ignored and therefore significant errors may be introduced. In order to illustrate figure 2 show possible paths for moving an object from point O (origin) to point D (destination). Figure 2a shows the two possible horizontal paths that would be followed if there is no simultaneous horizontal crane motions (α =1). The blue path indicates a path in which the crane operator first moves outward the trolley and then rotates the jib clockwise, while the green path indicated first rotating the jib and then moving the trolley. Values of α smaller than one take into account paths located within the gray area shown in the figure. An example is shown in figure 4b. However, if there are obstacles within this region, the operator would be forced to follow paths that are outside of this region that would not correspond to values of α between 0 and 1. Two possible examples are shown in figures 2c and 2d.

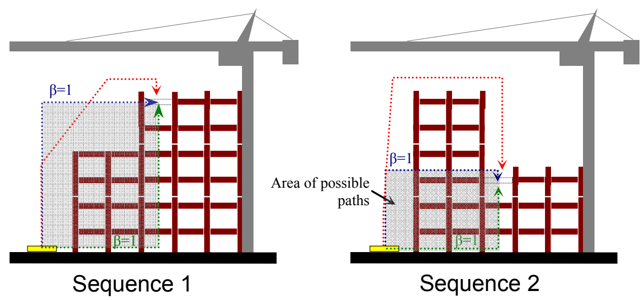


Fig 3. Simultaneous vertical and horizontal crane motions.

A similar situation exists with the combination of horizontal and vertical crane motions. Figure 3 shows two stages of the erection of a building in which the structural elements have been lifted in different sequences. The approach that was followed by Zhang et al. (1999) and by Tam et al. (2001) other investigators computes hoisting times based on the coordinates or the origin and destination but not the actual path. Figure 3 shows in blue and green the paths that crane operators would follow by considering the value of β assumed by this investigators which assumes that vertical motions never occur simultaneously with horizontal motions. The green path assuming that crane operator would first move the trolley and rotate the jib followed by lifting of the hook, while the blue paths assume that operator would first lift the hook to the height of the destination and then would move the trolley and rotate the jib to bring the piece in its required location. Clearly none of the two paths would represent a realistic path, because pieces already erected would not allow these paths. The gray zones in the figure indicate the regions of possible paths with values of β between 0 and 1. Again from the figure it is clear that these paths may not realistically represent real paths. Two possible paths are shown in red, which show that hoisting times could be significantly different from those computed with the approach suggested by these investigators.

Without considering actual operational trajectories of tower cranes, it is difficult to develop an ideal crane model to generate and evaluate the construction schedule before a construction. The research will introduce motion planning methods to facilitate the simulation of actual crane operations and result in detailed, accurate, and optimal construction schedules.

3 Direct and Inverse Manipulator Kinematics

Manipulator kinematics describes the motions of the tower crane and its hook's motion in a numerical format. The manipulator enables us to "operate" a tower crane in a computer to complete tasks in a virtual environment by using parameters. In other words, given the parameters which represent the degrees of freedom of the crane (jib rotation, lowering or rising of the hook, trolley movement), we can use manipulator kinematics to find the hook position. Inverse kinematics, on the contrary, for given the hook motion, and we can get the crane's motion.

The research treats the tower crane as a robot and uses methods employed in robotics to represent the motions of construction equipment. A robot configuration is a specification of the positions of all robot points relative to a fixed coordinate system. Usually a configuration is expressed as a vector of position or orientation parameters. Configure space (C-space) of a robot is a space of all its possible configurations, which are able to describe the attitude of a robot in Cartesian space, i.e. real world space.

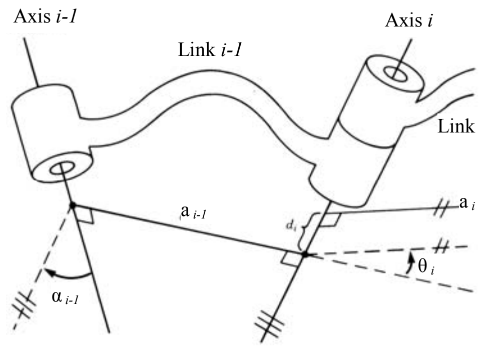


Fig 4. Link connection in a robot (Craig 1989).

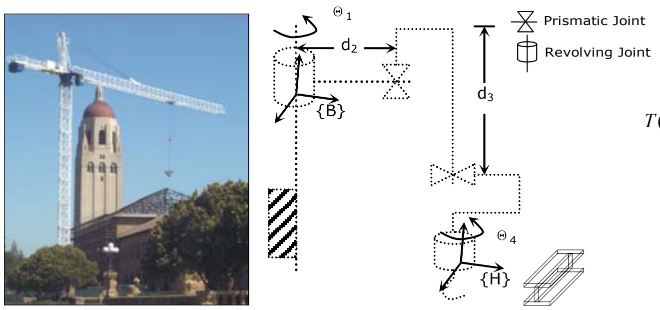
A robot essentially is a set of rigid bodies connected in a chain by joints. In robotics the rigid bodies are called links. As shown in figure 8 there is a joint between a neighboring pair of links (Craig 1989). The Denavit-Hartenberg notation (Denavit and Hartenberg 1955) is a commonly used method to describe a robot kinematically. The method defines each link by four parameters: two for representing the link itself and the other two for describing the connection to neighboring links. In each link, only one of the four parameters is a variable and the other three are constants parameters. Motions in each link of the robot can be achieved by constructing a transformation matrix *T* for each link (each degree of freedom) in the robot. The general format of the transformation matrix has the following form:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

Once the transformation matrix for each link has been obtained, developing the kinematic equation is straightforward. We can multiply all link transformation matrices, and find the transformation function ${}^{0}_{N}T$ from link 1 to link N as follows (Craig 1989; McKerrow 1991; and Niku 2001)

$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \ldots {}^{N-1}_{N}T \quad (3) $$

The transformation, ${}^{0}_{N}T$, is a function of n joint variables. The research will use the transformation function (equation 3) to obtain the motion of the hook from the different motions of the crane. This process is called ***direct kinematics of manipulators***. On the contrary, obtaining (solving) the crane's motions from the position and motion of the hook is called ***inverse kinematics of manipulators***. The direct kinematics of manipulators and inverse kinematics of manipulators are important tools for reformatting the tower crane from a Cartesian space to a configure space (C-space). The transformation is critical for applying motion planning algorithms.



hook orientation; hook position

$$T(\theta_1, d_2, d_3, \theta_4) = \begin{bmatrix} c_1c_4 + s_1s_4 & s_1c_4 - c_1s_4 & 0 & -d_2s_1 \\ s_1c_4 - c_1s_4 & -c_1c_4 - s_1s_4 & 0 & d_2c_1 \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For simplifying the representation, denote: $c_1=\cos\theta_1$, $s_1=\sin\theta_1$, $c_4=\cos\theta_4$, $s_4=\sin\theta_4$

Fig 5. Manipulator kinematics of a horizontal tower cranes.

The research uses Denavit-Hartenberg notation to analyze tower crane. A tower crane can be treated as four degree-of-freedom (4DOF) robots, where each degree of freedom represent each motion of the crane which can be represented by four variables, θ_1, d_2, d_3, and θ_4. As shown in figure 5, θ_1 represents the rotation of the jib, d_2 represents trolley radial movement, d_3 represents the lowering or rising of the hook, and θ_4 represented the rotation of the hook. A robotic schematic representation of a tower crane is shown in the center of figure 5. Using Denavit-Hartenberg notation, we can find the transformation function corresponding to each link (each motion in the crane). Then the overall link transformation $T(\theta_1, d_2, d_3, \theta_4)$ can be obtained by multiplying all transformation functions. The link transformation corresponding to tower cranes is shown on the right of figure 5. The variables θ_1, d_2, d_3, and θ_4 are space vectors which are used to construct the C-space of a horizontal tower crane

In the C-space, we are able to describe a scenario using the minimal set of needed parameters. For example, the motion of a tower crane can be described by only four variables, θ_1, d_2, d_3, θ_4. Positions to which the crane is not allowed to move in because they would cause a collision of the crane or of the object being lifted against another object can then be characterized as regions in the C-space, called C-obstacle. Hence, as shown in figure 6, the problem of finding a collision-free erection path can be simplified as to find a path that does not goes into C-obstacle regions in a C-space. Methods to find the collision-free path will. Be introduced in the following chapter.

4 Motion Planning and Collision Detecting

4.1 Motion Planning

The research applies Probabilistic Roadmaps (PRM), a motion planning algorithm, to find the moving trajectories of pieces being transported by the crane(s) (Latombe 1991; LaValle 1998; LaValle 2001). PRM is a simple and efficient randomized algorithm for solving single-query path planning problems in multiple dimensional C-space. The first step in using PRM is to obtain the C-obstacle by mapping the obstacles from a Cartesian space into the C-space. The second step is to obtain random samples points from the C-space outside of C-obstacles. This is achieved by generating random points in the C-space and verifying whether they lie within each of the obstacles. If these obstacles are moving, this verification needs to be done at a sufficient frequency relative to the velocity at which the crane and objects are moving such that would avoid a possible collision. Possible collision-free

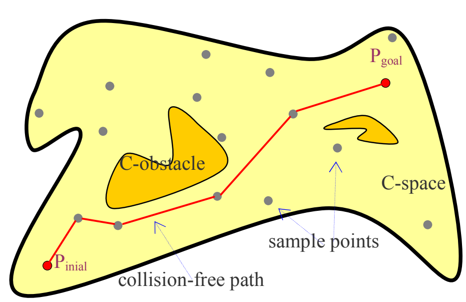


Fig 6. Motion planning in C-space.

paths are found by joining sampled points outside of the C-obstacles to form a path from the origin to the final destination of each piece. An optimum path is then found by comparing the times required by the various possible collision-free paths. Finally, the optimum collision-free path is mapped to the Cartesian space and the required crane motions are computed to achieve the desired path.

The Probabilistic Roadmaps motion planning method samples the random points in entire C-space, so its efficiency can be improved by incorporating new sampling techniques that take into account the logistics of construction sites as well as the particular motions of construction cranes. For example for safety of personnel near the loading point, it is better if the crane rising the object with enough clearing above personnel and trucks before initiating any radial or tangential motions. Similarly, motions near the unloading (destination) point may be constrained by particular maneuvers that must occur to facilitate the placement of the piece in its final position. For example steel fillers beams may need to be lowered with a small rotation with respect to a vertical axis relative to its final position in order not avoid hitting shear tabs.

The research develops new sampling strategies which will reflect the tower crane operation in a real construction sites. This means, as shown in figure 7a, sampling in the initial and final portions of the path will be constrained and modified by safety and by maneuvers required to place pieces being erected. Another example is shown in figure 7b in which a piece is being moved from point A to point B. A pure random sampling technique would generate collision free points anywhere within the outermost circle. A possible collision free path could be, for example, to follow a straight path between the two points. However, this path would require the trolley to first move inwards (towards the tower) and then outwards (toward the tip of the jib). A more efficient sampling technique is to first look for collision-free point within the gray area shown in the figure 7(b). Furthermore, for a given angular (rotational) velocity of the jib vibration caused by centrifuge forces will be minimized if the motion stays along the second circular as long as possible (without moving the trolley) before starting the simultaneous rotation of the jib and the outward translational motion of the trolley toward point B. The more efficient and realistic sampling strategy would then be to generate more sample points first along this second circle as the crane departs from point A towards point B. A third example is shown in figure 7(c). In which a first set of sample points are generated in vertical paths in the vicinity of loading and unloading points and around the envelope created the current state of the structure being erected in order to generate collision time-efficient free paths faster. In particular, the figure shows how the system would first generate sampling point at a constant height above the height point of the structure.

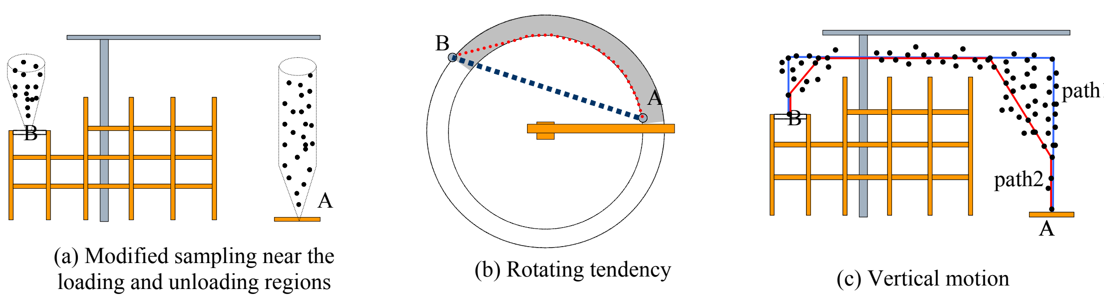


Fig 7. Improved sample strategies for the tower crane simulator.

4.2 Collision Detecting

Since most of current collision detecting methods are applied in robotics or computer graphics, the methods are generally more complex than necessary when using for construction purposes and relatively difficult to be implemented efficiently. Collision detecting is one of the most frequently-used function for planning the motion of a tower crane. The efficiency of collision-checking method will directly influence the efficiency of motion planning is to find a continuous collision-free path for operating a tower crane in a construction environment. By introducing reasonable tolerances, a simple but efficient algorithm is developed in the research.

Two main collision detecting methods, bounding volume hieracrchy (BVH) methods (Quinlan 1994; Gottschalk et al. 1996; Klosowski et al. 1998) and feature-tracking methods (Lin and Canny 1991; Cohen et al. 1995; Mirtich 1998), are broadly employed to find the collision free paths for robots or computer graphics purposes. BVH methods precompute each of elements of robots and obstacle a hieracrchy boundary volumes (e.g. spheres or boxes) that approximates the geometry of the object at successive levels of details. Since the geometrical features have been precalculated as a hierarchical data structural, to detect the collisions between two objects, BVH methods search the hierarchy boundary volumes from top to down to quickly discard large details of objects. Feature-tracking methods, on the other hand, keeps tracking vertices, edges, and faces of two objects to determine if the objects remain separated along the path. The methods assume the features between objects change very small in each time increment, so the new features can be computed efficiently from the old ones. These assumptions sometimes lead to inefficient tests in each tiny increment along a path to avoid missing collisions.

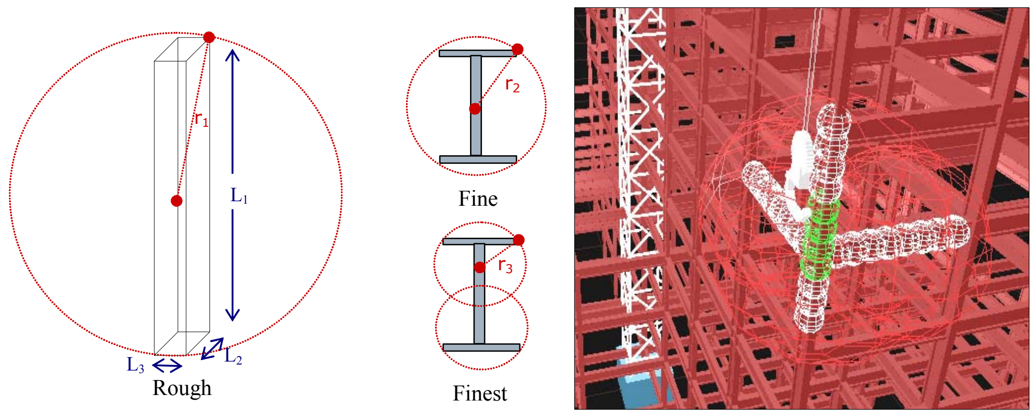
Algorithm 1 *CollisionDecting (cuboid1, cuboid2):* check two cuboids. If collision, return COLLIDED, else return DISTANCE between cuboid

```
€ ← a disance can be regarded as a collision
DISTANCE← roughCheck(cuboid1, cuboid2)
If DISTANCE < € then
    DISTANCE←fineCheck(cuboid1, cuboid2)
    If DISTANCE < € then
        DISTANCE←fineCheck(cuboid1, cuboid2)
        If DISTANCE < € then
            return COLLIDED
return DISTANCE
```

Using BVH methods for planning crane operations require pre-calculation the hierarchical boundary volumes (BVs) of all crane elements and obstacles in the space. Because we plan the entire crane motions in a computer instead of planning motions in real time like a robot, the pre-calculation is redundant. Although features-tracking methods are able to find the exact value of distances between objects, the methods cost too much for the unnecessary accuracy for crane operations. Hence, introducing reasonable tolerances, the research develops a cheaper and more efficient algorithm.

The algorithm takes full advantages of the features of crane operations and construction elements to improve the efficient of collision detecting. In general, crane operators tend to maintain a conservative distance between crane and obstacles to keep away from collisions. Except approaching to the target position of the rigging element, operators prefer to move the element along paths in an open space instead of passing through lots of obstacles. Therefore, a rough and conservative collision detecting method is ideal to be used in this scenario. The shapes of construction elements, on the other hand, are typically long cuboids (rectangular boxes) or at least can be represented as several cuboids. The boundary of a W type steel, for example, can be simply represented as a cuboids. Therefore, the system uses cuboids as the outer boundary of objects, and develops an algorithm for detecting the collision between cuboids. The method greatly reduces the computing cost.

As shown in algorithm 1, the collision detecting method calculates the distance between objects in three levels, rough check, fine check, and finest check. Assume the objects are both cuboids, each cuboid has length L_1, L_2, L_3, where $L_1 \geq L_2 \geq L_3$ (Figure 8). The rough check uses the longest length L_1 to form an external sphere as the outer boundary of each object. Checking the distance between two objects is simply to calculate the distance between the two spheres. The fine check uses second longest length L_2 to construct spheres to construct the outer boundary of an object. Therefore, to describe an object requires a pile of spheres. The number of spheres, M, is the ceiling number of the ratio between longest and second longest length (M=$\lceil L_1 / L_2 \rceil$). The finest check uses the shortest length in the cuboid to construct MxN of spheres, where M=$\lceil L_1 / L_3 \rceil$ and N=$\lceil L_2 / L_3 \rceil$. Although the algorithm may not return the exact value of the distance between objects, the value remains always conservative and with reasonable errors which are acceptable for construction purposes.



(a) rough, fine, and finest external spheres (b) collision detecting when hook close to the building

Fig 8. Using external spheres to detect collision.

Table 1 lists the errors and costs of collision checking in different levels when using the algorithm 1. In the rough level, we only use one sphere to represent one structural element. The cost of collision detecting in this level is the cost of calculating the distance between two spheres, and the error, in general cases, is approximately half of the longest length (L_1). Since structural elements of a typical building range from 6 meters to 12 meters, the errors of the rough check will be approximate 300 to 600 center meters. In the fine check, the cost is about 10-15 times more than rough check, and errors drop to 20 to 40 cm. The finest level needs to check 50-150 spheres in each object, but rewards center meter accuracy, which is acceptable for the crane operation. In most cases, the algorithm only checks the rough spheres between objects. Once finding collision between the rough spheres, the algorithm applies fine or finest for more accurate results.

The algorithm is efficient and easy to be implemented, and provides both collision detecting and distance checking. The errors are acceptable from construction point of view, and result in conservative distance and will benefit to find safer paths for the operators.

Table 1 Error and computation cost for different sphere types

Sphere Type	Error	Cost (unit)*	Error range **	Cost range **
Rough	$\sqrt{((L_1/2)^2+(L_2/2)^2+(L_3/2)^2}-L_3/2$	1	300~600 cm	1
Fine	$\sqrt{(2*(L_2/2)^2+(L_3/2)^2}-L_3/2$	$\lceil L_1/L_2 \rceil$	15cm~50cm	10-20
Finest	$(\sqrt{3}-1)L_3/2 \approx 0.36L_3$	$\lceil L_1/L_3 \rceil * \lceil L_2/L_3 \rceil$	5~15cm	15-250

* cost of calculating the distance between two spheres ** common steel sections using in building structures

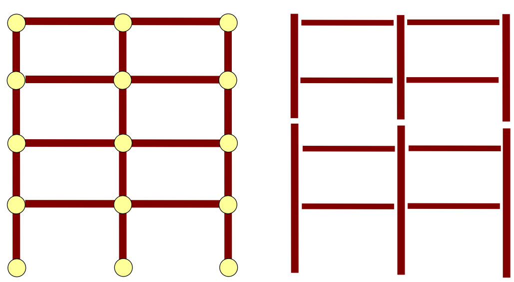
5 Construction Model and Visualization

5.1 Construction Model

Construction models are the models contain geometry information of a building. Based on the construction models, motion planning methods can be applied to plan the erection activities of each structural element, and combine the activities as an overall crane schedule.

The construction models can be generated from engineer models, i.e. structural models. During the design phase of a project, structural engineers construct engineer models by imputing geometry information to in structural analysis software, such as SAP 2000. To simulate a static and dynamic behaviors of a structure, engineer models locates joints in each connection between beams and columns (figure 9a). However, these joints are not actual joints in construction sites. It is very common to see the columns are two or three times longer than the floor height and connect with another column between the floors.

Algorithm 2 provides a method to convert an engineer model to construction one. The main efforts of the algorithm is to retrieve the geometry information of the columns from all elements and combine columns vertically based on construction constrains, such as the limit of transportation length or the limit of hoist weight of equipment. Finally, the construction model will be built by the combined columns and other non-column elements.



(a) engineer model (b) construction model

Fig 9. Construction model and engineer model

Algorithm 2 *buildCostructionModel* : Read structural analysis output file such as SAP 2000 (.s2k) to retrieve geometry information of all elements

```
elementArray ← engineer model
for each element in elementArray
    if element is a column then
        Add the element to collumnArray
    else
        Add the element to construction model
for each column in columnArray
    if the column is bottommost then
        Add the column to groundCollumnArray
for each groundColumn in groundColumnArray
    Combine columns to under construction constrains
    Add the longer column to construction model
```

5.2 Visualization

A visualization interface and crane dynamic behavior will be implemented in research. Site layout and the tower crane will be visualized in a 3D world. The system provides an interaction interface and will allow users control the view point and speed of the animation. Users will be able to shift the view points between front view, side view, top view, and operators' view or navigate the virtual environment. Visualization will illustrate the path to be followed by each piece, which can be used as a training tool for crane operators. Visualization of these paths helps improve the efficiency of

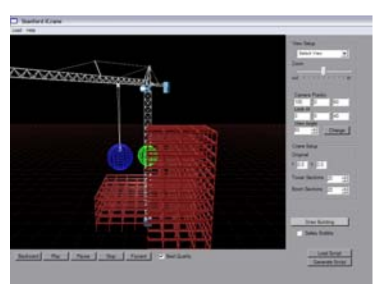


Fig 10. Snap shot of the visualization of motion planning of a tower crane operation.

crane operators. Furthermore, visualization of these paths can help improve the safety of crane operations by visualizing the crane motions in the computer before they are actually executed in the construction site.

Figure 10 is a snapshot of the computer system developed in the research. The system was implemented in an OpenGL platform in Microsoft .NET environment. The 3D building model is developed in SAP2000 and imported into the planning system. Using inverse kinematics, the system transformed the crane model and building model to C-space. This building model was regarded as obstacles in C-space, called C-obstacle. Given the initial and target points, the crane was able to find a collision free path by using PRM. Afterward, the path was mapped to the Cartesian space where it was displayed and animated in an OpenGL window.

In the visualization module, users are able to choose different play speeds, pause, rewind, or forward the animation to understand the erection processes clearly. Visualization will be able to be done for specific pieces, for groups of pieces, for specific periods of times (e.g. one hour, one shift, one day, one week, etc.) or for the whole project duration. The goal of the visualization is to provide a powerful tool to assist crane operators, project superintendents and project managers in making decisions regarding the project. For example, users will be able to visualize the progress that w ill be done during the next week of the project.

6 Conclusion and Future Work

A new method to automate and simulate erection activities is revealed in the research. Unlike currently erection schedules, mainly prepared by very rough estimates such as average number of tons erected per day of number of pieces erected per days, the system will provide detailed and accurate erection schedules by generating actual trajectories and crane motions. These erection schedules will be visualized in a 4D environment in which project managers will be able to visualize the state of the erection on any day, hour, or even tiny time increment of the schedule.

The research develops a computer system to generate and visualize erection activities by given building and tower crane models. The system develops a new crane-specific sampling technique for planning erection paths. An efficient and simple collision detecting method is developed and work well for planning paths. An algorithm of converting engineer model to construction model is developed as well. The computer system allows engineers to import an engineer model and obtain the animation of erection activities. The early simulation will help avoid potential construction problems in design phase, and result in more efficient and better-quality projects.

The research can be further developed by adding the motion planner for multiple-cranes and involving the dynamic behaviors of cranes to improve the safety of operational trajectories generated by the system. In the short term the result of the research provides an excellent tool for planning erection and to evaluate the constructability of various design alternatives and various construction sequences. In the long term, the research will be the basis for computer-assisted erection processes, in which the computer generates efficient and collision-free paths and subsequently crane motions that reduce or eliminate loss of productivity while at the same time improve the safety of crane operations. The computer will assist crane operators in a similar way in which computer assist pilots to fly airplanes. The research will also provide the basis for the development of robotic cranes to fully automate construction erection processes in a similar way in which car assembly and other industrial processes have been automated today.

Acknowledgements

This work has benefited greatly from discussions with Prof. Jean-Claude Latombe and Mr. Xiaoshan Pan. The preliminary prototype of the crane model was developed for the motion planning course taught at Stanford University by Prof. Latombe. This first prototype was developed by the first author along with Xiaosphan.

7 References

Anson, M., and Wang, S. Q. (1994). ‘‘Hong Kong performance yardsticks for concrete placing during building construction,’’ *Transportation of Hong Kong*, pp. 1–18.

Choi, C.W., and Harris, F. C. (1991). “A Model for Determining Optimum Crane Position,” *Proceeding of Institution in Civil Engineers,* Part 1, 1991, pp. 627-634.

Cohen, J. D., Lin, M. C., Manocha, D., and Ponamgi, M. (1995). “I-COLLIDE: An interactive and exact collision detection system for large-scale environments,” *Proc. Sym. on Interactive 3D Graphics*, pp. 189–196, 218.

Craig, John (1989), *Introduction to Robotics Mechanics and Control 2nd edition*, Addison-Wesley Publishing company, Inc.

Denavit, J., and Hartenberg, R. S. (1955). “A Kinematic Notation for Lower-Pair Mechanism Based on Matrices,” *Journal of Applied Mechanics,* pp. 215-221.

Furusaka, S., and Gray, C. (1984). “A Model for the Selection of the Optimum Crane for Construction Sites,” *Construction Management and Economic,* 2, pp. 157-176.

Gottschalk, S., Lin, M. C., and Manocha, D. (1996). “OBBTree: A hierarchical structure for rapid interference detection,” *Computer Graphics*, vol. 30, no. Annual Conf. Series, pp. 171–180.

Gray, C. (1983). “Buildability- the construction contribution,” Occasional Paper No, 29, London, Chartered Institute of Building.

Gray, C., and Little, J. (1985). “A systematic approach to the selection of an appropriate crane for a construction site,” *Construction Management and Economics, 3*, pp. 121-144.

Klosowski, J. T., Mitchell, J. S. B., Sowizral, H., and Zikan K. (1998). “Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs,” IEEE Trans. Visual. Comput. Graphics, vol. 4, no. 1, pp. 21–36.

Latombe, J. (1991). *Robot Motion Planning.* Boston: Kluwer Academic Publishers.

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

LaValle, S. (2001). “Motion Strategy Library, Version 1.2.,” Computer Science Dept., University of Illinois. Retrieved September 20 2003, from http://msl.cs.uiuc.edu/msl/.

Lin, M. C. and Canny J. F. (1991). “A fast algorithm for incremental distance calculation,” in IEEE Int. Conf. on Robotics and Automation, pp.1008–1014.

Mckerrow, P. (1991). *Introduction to robotics*. Sydney: Addison-Wesley Pub. Co.

Mirtich B. (1998). “V-clip: Fast and robust polyhedral collision detection,” *ACM Trans. on Graphics*, vol. 17, no. 3, pp. 177–208.

Niku, Saeed B. (2001). *An introduction to robotics analysis, systems, applications,* Upper Saddle River, N.J.: Prentice Hall.

Quinlan, S. (1994). “Efficient distance computation between non-convex ob-jects,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 3324–3329.

Rodriguez-Ramos, W. E., and Francis R. L. (1983). “Single Crane Location Optimization,” *Journal of Construction Engineering and Management, 109*(4), pp. 387-397.

Tam, C. M., Tong, K. L., and Chan, K.W. (2001). “Genetic Algorithm for Optimizing Supply location around tower crane,” *Journal of construction engineering and management,* 127(4), pp. 315-321.

Zhang, P., Harris, F. C., Olomilaiye, P. O., and Holt, G. D. (1999). “Location Optimization for a Group of Tower Cranes,” *Journal of construction engineering and management, 125*(2), pp. 115-122.