Optimization and Evaluation of Automatic Rigging Path Guidance for Tele-Operated Construction Crane

H.L. Chi[a], S.C. Kang[b], S.H. Hsieh[b] and X. Wang[a]

[a] Australasian Joint Research Centre for Building Information Modelling (BIM), Curtin University, Australia

[b] Department of Civil Engineering, National Taiwan University, Taiwan

E-mail: hung-lin.chi@curtin.edu.au, sckang@ntu.edu.tw, shhsieh@ntu.edu.tw, xiangyu.wang@curtin.edu.au

Abstract -

The objective of this research is to develop, optimize and evaluate an approach for automatically generating rigging path guidance in order to provide sufficient assistance on the user interface of tele-operated crane. Besides operational safety and efficiency, this approach focuses on solving operability issues of following path guidance on a tele-operated crane system. To provide path guidance that can realistically be performed in a tele-operated rigging scenario, three factors are proposed for consideration in the development of such an approach: *Visibility*, *Control Features*, and the *Dynamic Environment*. A simulation platform in the virtual environment, called *PathGuider*, is developed for evaluating the use of the proposed path guidance considerations. Demonstrations for visibility and the dynamic environment issues have been performed and they show that the guidance path can be re-planned in real-time once the guided trajectories are being sheltered by surrounding buildings from operators' view or dynamic objects interfered with it. To evaluate the efficiency of the developed re-planning algorithm, a benchmark has been developed. The results show that the calculation time of cases considered are all within real-time span. The proposed path guidance approach reveals its efficiency and feasibility in providing useful assistance to tele-operated crane's operators.

Keywords -

Tele-operation; Optimization; Crane Operation; Rigging Path Planning; PRM

1 Introduction

Tele-operation, the remote control of machines, has in recent years been increasingly applied in practice and been the subject of research [1]. Tele-operation allows for a reduction of human presence at a site, manipulation on avatars to finish tasks and hence significantly reduces safety risks. Tele-operation technologies have been widely applied in diverse fields such as rescue missions, mining, and space exploration. They also show promising results on improving productivities of manufacturing fields.

With the increase in research effort focused on tele-operation applications for improving construction operations, the interface design of construction equipment remains an important topic for further study. Due to variation in operation functions and automation assistance with different construction equipment, the interface design can directly influence the efficiency and effectiveness with which operators can finish their tasks. Yokokohji *et al.* [2] surveyed state-of-art robots for rescue missions and proposed design guidelines for the tele-operation interface, saying that the global view which covers the remote machine itself should always be included in the user interface for operators to monitor the overall situation while robots are working in challenging environments. This has also motivated further examination of the relationship between humans and tele-operated user interfaces in this research.

During the last two decades, a significant amount of research in the area of rigging path-planning problems for construction cranes has been aimed at using computers to automatically generate the rigging path solutions. For current development of motion planning in construction, it is usually utilized in simulations at the planning stage before actual constructions. Sivakumar *et al.* [3] used heuristic functions to seek out collision-free paths for dual-crane cooperative rigging activities, while Ali *et al.* [4] targeted the same problems but instead used a genetic algorithm and achieved more efficient results. Kang and Miranda [5] utilized the probabilistic roadmap method (PRM) and developed and implemented three different path-planning methods to handle different path-planning problems while minimizing computation cost. Zhang *et al*. [6] have analyzed and implemented path re-planning methods for cranes using rapidly randomized tree (RRT) methods. Chang *et al*. [7] developed a planning method simplified search dimension of rigging elevation to efficiently solve cooperative rigging activities.

Although extensive research has been conducted on the above-mentioned aspects, the path-planning methods for tele-operation and operational feasibility for real crane operators are rarely discussed. They are two important issues influencing rigging path guidance for remotely and manually controlled cranes. Removing the operator from the field introduces a challenge as a result of limitations in viewing the operations. Due to inevitable restrictions in the field of view of the remote camera(s) and in the quality of the visual displays fed back from the remote worksite, the operator is often unable to maintain a level of situational awareness sufficient for safe and efficient task execution [8]. Although multiple cameras mounted to view the field of operations would be of benefit to remote operation, guidance paths should also be generated that consider the field of vision of the cameras. In this research, an approach is developed to automatically generate rigging paths for operators to remotely control the cranes in real-time. The guidance paths are generated using the PRM in C-space to enable a collision-free operation field, and taking into account the three specified feasibility factors for tele-operation.

2 Automatic Rigging Path Guidance

Considering the topics of automatically generating appropriate rigging paths by computers, many research have developed and provided variable solutions, which shows great impacts on improving construction automation. However, current research usually focuses on safety and efficiency aspects rather than operational feasibility issues and tele-operated scenarios. As tele-operation guidance of this research, the generated paths can only be useful if they fit the common sense of crane operations and operators are willing to reference them. So operational feasibility of the suggested paths becomes an essential issue and we put focus on it during developing the proposed approach.

2.1 Pilot Test

In order to solve the feasibility issues of suggested rigging paths, the requirement of operable motion has been identified. Actual end users participated in a usability test to evaluate a developed user interface (UI) [9] for a tele-operated crane. In this test, a KUKA KR 16 robot arm was used as a crane prototype and four monitors displaying different angles of view (top, right-side, left-side, and global view) are also provided to remotely display conditions on the construction site model. The user interface with augmented reality (AR) technology–enhanced rigging path guidance can be seen in the top and global views. This path is generated automatically by simply using PRM in Cartesian space.

Five professional crane operators were recruited and participated in an in-depth user test to verify the usability and feasibility of the developed UIs. All the operators held professional licenses from the Industrial Safety and Health Association (ISHA) of Taiwan, and each had over ten years of experience. The testing times ranged from 63 to 92 minutes. The operators were asked to execute the simulated rigging tasks previously designed, and were then interviewed after completing the tasks.

From the observations of the tests and the subsequent interviews, it was found that one suggestion was most often provided by the operators, which was that the virtual rigging path guidance should be improved. The path taken during rigging depends on the style of each operator. However, all operators agreed that paths are not simply composed of straight lines, and must be generated according to the natural movement of the crane so that they can be easily followed. Guidance paths if possible should not be generated over the heads of field workers or existing buildings, and should be kept at low elevations. The view angle can also limit an operator’s depth perception if the guidance path is only shown on the top and global views. Considering the feedback from all test participants, it was concluded that three feasibility factors need to be considered to improve the current mechanisms of path guidance generation: *visibility*, *control features*, and the *dynamic environment*. The basic principle and these factors are described below.

2.2 Approach Overview

The proposed approach is basically established on a famous motion planning method called Probabilistic Roadmap (PRM) [10]. All the algorithms, as path planning considerations, developed in this research can be integrated into the process procedure of PRM and further considerations can also extend by the same way in the future. The general procedure of PRM can be seen as Figure 1.

The user should identify the starting and the end point in the space first. Then, the PRM method randomly samples geometrical points in the space and combine them with adjacent ones including the starting and end point. All the sampled points should not overlap with other objects within the space. It then becomes a graph structure G that every geometrical point represents vertex V and the connections between each vertex represent edge E. The formulation of a graph structure shows as follows:

$$G = (V, E) \quad (1)$$

After a graph structure established, it becomes an optimization problem which maximizes users' requirements to find out edge combinations between the starting and end point by using graph traverse algorithms. In the common situation, distance between two vertices represents the cost of their edge. And the shortest path algorithms have capability to find out the shortest solution with minimum cost edges passed by, which optimally solves the demand of efficient purpose. In other words, the solution shows efficient rigging path suggestion regarding to moving distance and time spent in this research.

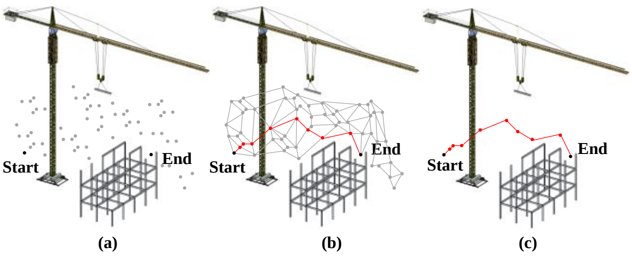

Figure 1. The general procedure of the Probabilistic Roadmap (PRM): (a) assign a start and an end point and randomly sampling on the space; (b) connect points as a graph; and (c) traverse the graph to find a collision-free and feasible path

In addition, collision detection becomes an important process needed to be considered during path finding procedure. It can make sure each of path segments is not collided with surrounding obstacles in the space. The paths blocked by obstacles cannot be operated and is not correct suggestion as candidate solutions for guiding operators. The collision problems can be prevented by ray tracing method, which determines whether the two vertices on two sides of an edge can "see" each other or not. In this research, we adopted this method for solving collision-free issues.

Besides securing the potential collisions of the rigging objects, all the movement of crane components should be collision-free as well. The method to guarantee the collision-free trajectories of all the crane components is Configuration Space (C-space) transformation. In this research, as shown in Figure 2, a tower crane is used as a rigging platform to develop the proposed approach. The position of the rigging object can be represented with respect to Cartesian coordination system (X, Y, Z). It can also be represented with respect to the tower crane's configuration. A configuration of the tower crane is given by (r, θ, l), which is a 3-DOF manipulator. The symbol r denotes the radius of the boom rotation as well as the current distance between the mast and the trolley; θ represents the rotation angle of the tower crane, and l represents its current hoisting length. All the movements of the tower crane become one single point on the C-space. And every obstacle in Cartesian space transforms to the ones in different shape in the C-space, which is called C-obstacle. The C-obstacle represents a batch of postures (r, θ, l) of crane we should avoid to perform because collisions can happen on somewhere of the crane.

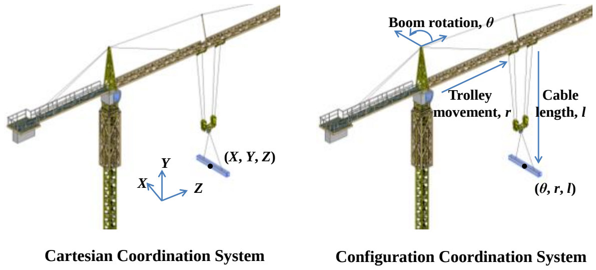

Figure 2. The difference between the Cartesian and configuration coordination systems

Rather than sampling feasible points for path finding on Cartesian space, we perform sampling procedure on the C-space according to the different postures (configuration) of the tower crane. The feasible vertices and their connected edges are generated in this three-dimensional C-space using PRM. Once we found the collision-free path avoiding from C-obstacles, points on the path represent continuously movement of the tower crane and guarantee every components of the crane doesn't encounter the collisions. This is basically the principle of finding safety rigging paths in this research.

A* search algorithm is used as the shortest path searching method in this research. Instead of Dijkstra's algorithm which evaluate all combinations of vertices in the graph to find an optimal solution, A* search introduces a partial heuristic function for assessing the quality of current combinations and eliminating unlikely solutions during search processes. As can be seen in Equation (2):

$$f(x) = g(x) + h(x) \quad (2)$$

How to identify the cost of a path passing vertex x, is decided by two parts. The first part is $g(x)$, which represents the function calculating the exact cost from the starting vertex to current vertex x. It is the same as conventional search principles which focus on what choices we have based on the current search status. About the second part $h(x)$, it represents a heuristic estimate function to calculate the expected cost from current vertex x to the destination. Unlike Dijkstra's algorithm which guarantees optimal result but requires considerable computing time, A* search algorithm provides a balance mechanism between efficiency and performance by eliminating unlikely solutions with heuristic judgments and at the same time keeping optimization promise. It is thus integrated as the primary search stagey in this research.

The developed path finding algorithm, named *PathSuggestor* in this research, can be seen as Table 1. Assuming that the C-space transformation has been performed and all the C-obstacles have been identified at the pre-processing stage, the user initials a starting point and an end point as input parameters. Then we randomly sample feasible points in the C-space. For establishing a graph structure, every connections (edges) between each sampled adjacent point need to be examined to see whether they are connected without collided with any C-obstacles. We use *OperableCheck* function to archive this objective. Afterward, we get a cost matrix representing the weight of each edge on a graph. It then is used as a parameter of *AStarShortestPath* function. The function is typically an A* search algorithm to find out an efficient path from P_{init} to P_{end}, which minimizes the cost in sum of every edge it passed by.

Table 1. *PathSuggestor* algorithm

Algorithm *PathSuggestor*(P_{init}, P_{end}): Find a collision-free, shortest, and operable path for rigging guidance.
P_{init}: The starting point of the rigging object.
P_{end}: The destination of the rigging object.
P: Suggested shortest rigging path.
M: Cost matrix of each geometrical point pairs.
n: Sampled geometrical points matrix.
Count: Path re-search counter.

```
n ← GridSampling()
// Randomly sampling vertices in C-space
M ← OperableCheck(n)
// Check collisions among every edge
Count ← 0
WHILE Count ≠ α
   P ← AStarShortestPath(P_init, P_end, M)
   // A* shortest path search
   IF CheckVisibility(P) = FALSE
   // If P doesn't fit, search again
      Count ← Count + 1
      IF Count = α
      // If Count equals to threshold, abort
         P ← NULL
         BREAK
      CONTINUE
   ELSE
      BREAK
   //Otherwise, P fits the requirement
END WHILE
RETURN P
```

For considering operational feasibility on tele-operation scenarios, *PathSuggestor* approach is developed for handling *visibility*, *control features* and the *dynamic environment* concerns. They are described in the following sub-section individually. It should be noticed that *PathSuggestor* approach or even original PRM algorithm did not consider the effects from physic-based behaviors, such as rigging cable swing, in this research.

2.3 Visibility

Considering operational feasibility of suggested rigging paths, viewing limitations of cameras on the tele-operated crane need to be addressed and eliminated from the path searching space. With the lack of visualized information and immersive senses on tele-operation scenarios, these remote cameras become operators' "eyes" and operations are significantly depend on how these camera are set around. Without sensing clear images on the space, it is useless to guide operator passing those unknown regions. So the suggested rigging paths should not pass blind zones where the visions of cameras are blocked by surrounding buildings or objects at construction site. Even in the conventional operation scenarios, it is the rule of thumb for rigging activities as well.

To preserve the operator's depth perception and visual information, paths that are partially or completely unmonitored or monitored by one camera only are eliminated from the tele-operation scenarios. Every geometrical points on the candidate path is examined in term of visibility using the ray tracing method to observe if any obstacles are located between the point and camera positions. If the views are blocked, the candidate path will be discarded and another path is re-planned using the same evaluation process. A path may only be generated within regions monitored by at least two cameras to ensure that the operability and safety of users following the path is not compromised.

As for *PathSuggestor* algorithm mentioned in previous sub-section, a *Count* variable has been added to count the times of path re-search. After calling *AStarShortestPath*, the system will call *CheckVisibility* to check whether the visibility of candidate path encounters limitations or not. In the *CheckVisibility*, we check every combination of vertices on candidate path P_i and camera positions P_j. If one of these combinations cannot see each other, discard candidate path and search another solution again with increment value of *Count*. If *Count* doesn't exceed the threshold α, the re-search iteration will continue until the candidate path fits visibility requirements. Otherwise, it will stop and report that no solution can be found to users.

2.4 Control Features

From the observations and feedback of crane operators, it was found that operators tend to operate the crane along trajectories which allows them to maintain a low elevation and not to cross over any existing objects on the construction site. This is due to safety concerns regarding potential falling payloads and an attempt to minimize any potential damage to the surrounding environment. In addition, a general control style of the operators is that they tend to maintain the same rigging direction unless the environment changed or they encountered obstacles.

In consideration of these issues, the edges of sampled graph in C-space were weighted as in Equation (3):

$$C^{*}_{ij} = (W_E + W_O + W_C + 1)C_{ij} \tag{3}$$

where C_{ij} denotes the traveling cost from point i to j in the C-space sampled graph, and C^{*}_{ij} denotes weighted cost considering the control features described above. Meanwhile, W_E represents the weight of the rigging elevation, which is directly proportional to the difference of elevation value z between point i and j in Cartesian space. The bigger difference between elevation of point i and j, the higher weight influencing the travel cost from point i to j is:

$$W_E \propto z_j - z_i \tag{4}$$

The weight W_O governs crossing over existing objects on the construction site. Every object, including dynamic objects, should be examined in relation to whether or not their overhead Cartesian space overlaps the trajectory of the candidate point pair $n_i n_j$. If so, a penalty value α will be assigned between the related C-space point i and j. If not, the weight will not influence the cost in this case, as:

$$W_O = \begin{cases} \alpha, & Overhead\ \overrightarrow{n_i n_j}) = true \\ 0, & Overhead\ \overrightarrow{n_i n_j}) = \ false \end{cases} \tag{5}$$

The weight W_C governs changing the rigging direction. Less rigging actions decrease the risk during rigging operations, fit with a human control style, and save on construction costs. The value of the weight from point i to j depends on the direction of the previous candidate point pair $n_r n_i$. Here, n_r denotes the neighboring point of n_i, which is selected previously in a candidate path. If the direction of $n_i n_j$ does not equal that of $n_r n_i$, a penalty value β will be assigned between the related C-space points i and j. If not, then the weight will not influence the cost in this case, as:

$$W_C = \begin{cases} \beta, & \overrightarrow{n_i n_j} \neq \overrightarrow{n_r n_i} \\ 0, & \overline{n_i n_j} = \overline{n_r n_i} \end{cases} \tag{6}$$

The weighting mechanism is integrated into the search method and applied as a heuristic function (as *OperableCheck*) for evaluating the cost of candidate point pairs on the sampled graph. The weighting terms of Equation (3) can have an influence on each other. For example, if the candidate point pair along an elevation direction is selected at the beginning of the search process, the value of W_C influences the search results to continue selecting point pairs in the same direction. However, the increasing value of W_E will eventually affect the search process in such a way as to change the searching direction.

In addition to safety concerns, human control capacity must also be considered during crane operations. Crane operators can manipulate, at most, two DOFs of a crane simultaneously. For example, though manipulations like rotating the boom while hoisting allow for a decreased operation time, they are subject to the operators' perception capacity. However, the paths generated by the PRM may not take this factor into consideration and can sometimes involve the manipulation of more than two DOFs. This may not be feasible in practice and operators need to create an alternative rigging plan by themselves. Even the suggested path is feasible to operate, the operator needs to pay more attention on extra DOFs which may exceeds operator's perception capacity and cause potential safety issues in crane operations.

To deal with the capacity problem, it is proposed to use a grid sampling method in C-space when searching for a feasible path. As shown in Figure 3, the edges of the graph can only be linked to adjacent points. Points must be linked by creating a rectangular shape for a single DOF configuration, and points can be linked diagonally for a 2-DOF configuration. The path found on this graph is guaranteed to be within the capacity of human manipulation to perform by using combinations of the control sticks.

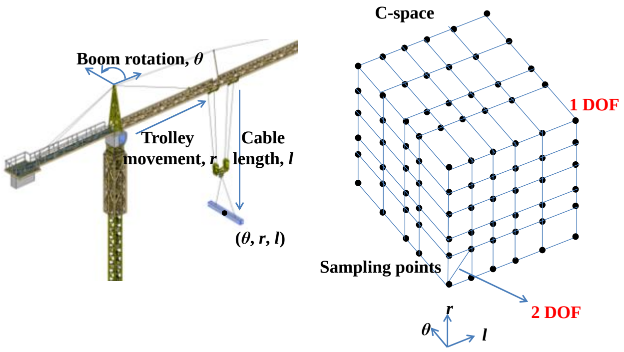

Figure 3. Overview of the grid sampling method

The grid sampling method is implemented as *GridSampling* function in the *PathSuggestor* algorithm. In each C-space's dimension, we randomly sample a number of digits to become an element serial. The length of element serials is ceiling of $\sqrt[3]{n}$, where n denotes the desired number of samples. After all three dimensions are sampled, the combination of all three element serials becomes the result matrix of grid sampling. It then returns for furthering operable check and the search graph building procedures.

2.5. Dynamic Environment

When considering the complexity of a modern construction site, it is very likely that rigging tasks can be interfered with by the surrounding elements, such as other construction vehicles or machines. This is potentially a very serious problem which needs to be specially handled in tele-operation scenarios. Without a human presence, it is difficult to handle any unexpected scenarios if the construction plans for the tele-operated machines on the site are not well-established. The participants of the test also pointed out the fact that the re-planning ability of rigging plans is a requirement of a tele-operation system to handle the dynamics of a construction.

Considering the usage of PRM in this research, it is time-consumed for handling dynamic objects. The large computational demand is because once the status of an object is updated, the entire C-space needs to be rebuilt in order to identify any potential collision regions. To improve the computational efficiency, the Octree structure for collision detection was integrated. This can be used to partially rebuild those regions of space that have potentially changed due to dynamic objects. In the redesigned C-space, each element (r, θ, l) contains the information about the relative Cartesian position (x, y, z) and is categorized according to the geometric relationship of the Cartesian space.

Firstly, all the elements in the C-space are input into the tree root. The space is then split into eight octants, segmenting the related Cartesian space, and the elements covered in each octant are recorded into the corresponding child structure. Secondly, each octant is split and the elements are recorded in the same way, and this is repeated until the desired level is achieved. The layout of the octree structure with respect to the current environment is built in this fashion. By noting the positions of any dynamic objects (can be done by localization technologies in construction) and traversing octree structure, the groups of elements (C-space regions) whose Cartesian positions are near the dynamic objects can be retrieved. Next, it is attempted to rebuild these elements with the collision detection methods. In an ideal case, the number of elements need to be rebuilt can be reduced from N to approximately $N/8M$. The variable N represents the number of elements (resolution) in the entire C-space. And the variable M represents the level number of octree structure. The algorithm of this updating procedure, named *UpdatePartialSpace*, is described as Table 2.

The following algorithm, as shown in Table 3, is called at every time frame of crane operation. Every time when an operator is performing operation according to the suggested path, the algorithm will monitor each moving obstacles in the surrounding environment. It can be done by attaching markers or devices and use positioning technologies to track them, such as GPS or UWB. The algorithm can update the cost matrix M, generated by *PathSuggestor* algorithm, to M' according to *UpdatePartialSpace* function.

Table 2. *UpdatePartialSpace* algorithm

Algorithm *UpdatePartialSpace*(S, $P_{obstacle}$): Using octree structure to identify and update edges' collision conditions of partial C-space near obstacles.
S_{cspace}: Partial C-space needs to be updated.
S: Partial Cartesian space related by S_{cspace}.
s: Split sub-space of S.
n: Sampled geometrical points inside S_{cspace}.
$P_{obstacle}$: The Cartesian position of the moving obstacle.

IF S.*Covered*($P_{obstacle}$) = **TRUE**
// If S includes the moving obstacle
  IF S.*SubSpace*() = **NULL**
  // If S is leaf node in octree, return related n
    $S_{cspace} \leftarrow$ *Mapping*(S)
    $n \leftarrow$ *GetSamples*(S_{cspace})
    RETURN n
  ELSE
  // Otherwise, recursively search its sub-spaces
    FOREACH s in S.*SubSpace*()
      $n \leftarrow n$ + *UpdatePartialSpace*(s, $P_{obstacle}$)
ELSE
// If the moving obstacle is out of S, return empty
  $n \leftarrow$ **NULL**
  RETURN n

Table 3. *PathRePlanning* algorithm

Algorithm *PathReplanning*(): Update cost matrix according to the position of moving obstacles, and re-plan the path if it is necessary.
$P_{obstacle}$: The Cartesian position of the moving obstacle.
L: The list of moving obstacles.
n: The list of sampled vertices needed to be updated.
P: The current suggested path.

FOREACH $P_{obstacle}$ in L
// Collect sampled vertices needed to be updated
  $n' \leftarrow n'$ + *UpdatePartialSpace*(S, $P_{obstacle}$)
$M' \leftarrow$ *PartialOperableCheck*(n', M)
// Update partial cost matrix
FOREACH P_i in P
  IF n'.*Contain*(P_i) = **TRUE**
  // If a point on the path needs to be updated
    $P \leftarrow$ *AStarShortestPath*($P_{current}$, P_{end}, M')
    // A* shortest path search
    BREAK
RETURN P

Once the list of sampled points needed to be updated (n') contains the point (P_i) along the suggested path, which means the moving obstacle is intersecting or near somewhere along the suggested path, the algorithm will re-plan another path according to current rigging position for operator's reference. As such, this approach makes the re-planning process more efficient when the positions of dynamic objects change, particularly in the case of a complex construction site containing many obstacles.

3 Implementation and Evaluation

To demonstrate and evaluate the feasibility of the proposed path guidance approach, a simulation system was implemented namely *PathGuider* in a virtual environment. The C# language was used to implement the system on a .NET developer platform. Also included was a graphic engine, implemented with the Microsoft® XNA framework to simulate the rigging tasks and environment, and a physics engine, implemented with the PhysX library to handle the collision detection and ray tracing processes. The implemented algorithms include basic PRM method, *PathSuggestor*, *UpdatePartialSpace* and *PathReplanning*. As for the *OperableCheck* function, W_E is assigned as a function: $W_E = 0.002h$, where h denotes elevation. α equals to 0.2, and β equals to 0.2. Further study about how to choose optimized weighting configuration will be conducted in the future.

In the test scenario of *PathGuider* system (as shown in Figure 4), a 3-DOF tower crane, a mobile crane, and an under-constructed steel frame building similar to that in the pilot test were set up. The movement of the tower crane was guided by the suggested paths, and the mobile crane, which is free to move around the construction site, will be treated as a dynamic object in the environment.

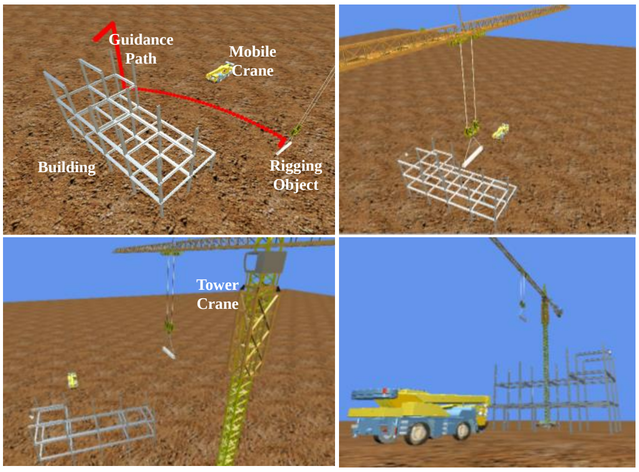

Figure 4. A test scenario setting for *PathGuider* system

For evaluating whether the proposed path guidance algorithms have achieved our design intentions, we made demonstrations and a benchmark for adopting algorithms with *visibility* and the *dynamic environment* consideration. An i7 Dual-Core 2.8 GHz computer with 4 GB memory was used in these tests. In searching for a suitable path, the sampling number of point was 500. The resolution of the C-space was 200×180×200.

By testing the algorithm considering visibility issues, we selected single camera scenario to demonstrate how it works. In the beginning of the demonstration, the path planning algorithm generates a rigging path from a specific starting point to a destination. A first perspective view is set up and the user can control the camera's viewing angle and position freely. By moving the camera, the user is expected to see real-time path regeneration happened while the viewing direction of camera is blocked by existed building component.

To evaluate the efficiency of the developed re-planning algorithm, a benchmark for the dynamic environment has been developed. The demonstration is set up as test scenario mentioned above. Among the pre-processing stage of establishing C-space, we use five octree structures with different levels (from 3 to 7). In the virtual environment, a sphere-like object was randomly placed in the areas surrounding the guidance paths in order to observe the calculation time for the re-planning process and ensuring collision-free path generations. For each octree structure, we recorded the calculation time for 400 times of the conditions which the placed position of the sphere-like object did not trigger the re-planning action. Similarly, the calculation time for 100 times of the conditions which the placed position of the sphere-like object triggered the re-planning action has been recorded. In this benchmark, we are able to know the relationship between established octree's level and re-planning calculation time. Also, the efficiency of the proposed re-planning algorithm can be identified.

4 Result and Discussion

The demonstration of the proposed algorithm considering the visibility issues is illustrated at Figure 5. From a conventional view of the tower crane cabin, the planned path has been partially blocked by existed steel frame structure (Figure 5a). The proposed algorithm responses in real time to re-plan another path which can be observed completely for the view (Figure 5b). This algorithm is also valid with more views in a tele-operated scenario.

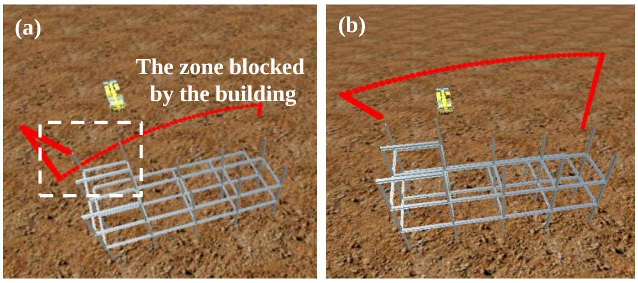

Figure 5. The demonstration of the path guidance re-planning considering visibility issues: the path (a) is partially blocked and (b) is re-planned after the blocked condition

The re-planning process benchmark for varying level numbers of octree structures was tested. For individual structure, 500 cases were collected. The results show that it is not significantly beneficial to increase the numbers of structure levels during the actual rebuilding and re-planning processes. However, the computation time for regular checks decreases significantly, and the checking processes are executed at every time step. It shows that a suitable selection of octree level numbers, five in this case, can achieve the most efficient results on average. In addition, the results of the cases considered are all within 200 ms. This shows that real-time rigging path planning for tele-operated cranes is feasible and workable.

5 Summary and Conclusion

A rigging path-planning approach modified from PRM for tele-operated scenarios was developed in this research. By conducting pilot test with crane operators on a prototype tele-operated crane, three operational feasibility factors were identified: *visibility*, *control features*, and the *dynamic environment*. In consideration of the visibility issues, the guidance path can only be generated within regions monitored by at least two cameras, in order to ensure the user's operability when they are following the paths. A weighting mechanism for control features to adjust the travel cost between every configuration space (C-space) point pair was developed as well. By adapting the selection mechanism, the guidance path can avoid crossing over existing objects, maintain rigging directions, and maintain low elevations as far as is possible. The suggested path can also be followed with at most 2 DOFs movement under operators' perception capacities. In consideration of the dynamics of a construction site, an octree structure for partially rebuilding the C-space was integrated to speed up the re-planning process.

To demonstrate and evaluate the developed approach for rigging path guidance, a simulation system, named *PathGuider*, was implemented with a test scenario similar to that of the pilot test for crane operators. Two demonstrations, showing the real-time path re-planning ability and securing visibility of the proposed algorithms, were performed and presented promising results. A benchmark was developed for measuring computation time of re-planning with respect to the octree level numbers. The results showed that the re-planning process using 500 sampled points to ensure a collision-free guidance could be completed within 200 ms. This shows that real-time rigging path planning for tele-operated cranes is feasible and workable. It is envisaged from the study that the developed path planning approach can generate paths that are not only safe and efficient, but also operationally feasible for tele-operated scenarios. The control features could be further investigated by defining its optimization objectives and conduct simulations, to get ideal weighting configurations (W_E, W_O, and W_C) for general rigging scenarios. It can be extended to apply in other kinds of construction equipment by defining the weighting mechanism through human factors discovered in operating such equipment.

6 Acknowledgement

The authors would like to thank the National Science Council of Taiwan for the financial support they have provided to us under the project NSC 98-2221-E-002-185-MY3.

References

[1] Bernold, L. E. Control Schemes for Tele-Robotic Pipe Installation. *Automation in Construction*, 16(4): 518-524, 2007.

[2] Yokokohji, Y., Tsubouchi, T., Tanaka, A., Yoshida, T., Koyanagi, E., Matsuno, F., Hirose, S., Kuwahara, H., Takemura, F., Ino, T., Takita, K., Shiroma, N., Kamegawa, T., Hada, Y., Zheng, X. Z., Osuka, K., Watasue, T., Kimura, T., Nakanishi, H., Horiguchi, Y., Tadokoro, S. and Ohno, K. Design Guidelines for Human Interface for Rescue Robot. *Rescue Robotics*, Springer, London, 2009.

[3] Sivakumar, P. L., Varghese, K. and Babu, N. R. Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search. *Journal of Computing in Civil Engineering*, 17(3): 197-207, 2003.

[4] Ali, M. S., Babu, N. R. and Varghese, K. Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm. *Journal of Computing in Civil Engineering*, 19(2): 182-193, 2005.

[5] Kang, S. and Miranda, E. Planning and Visualization for Automated Robotic Crane Erection Processes in Construction. *Automation in Construction*, 15(4): 398-414, 2006.

[6] Zhang, C. and Hammad, A. Improving Lifting Motion Planning and Re-Planning of Cranes with Consideration for Safety and Efficiency. *Advanced Engineering Informatics*, 26(2): 396-410, 2012.

[7] Chang, Y. C., Hung, W. H. and Kang, S. C. A Fast Path Planning Method for Single and Dual Crane Erections," *Automation in Construction*, 22: 468-480, 2012.

[8] Wang, X. and Dunston, P. S. Mixed Reality - Enhanced Operator Interface for Teleoperation Systems in Unstructured Environment. In *Proceedings of the 10th Biennial ASCE Aerospace Division International Conference on Engineering, Construction and Operations in Challenging Environments*, League City/Houston, Texas, 2006.

[9] Chi, H. L., Chen, Y. C., Kang, S. C. and Hsieh, S. H. Development of User Interface for Tele-Operated Cranes. *Advanced Engineering Informatics*, 26(3): 641-652, 2012.

[10] Kavraki, L. E., Svestka, P., Latombe, J. C. and Overmars, M. H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4): 566-580, 1996.