

Engineering support systems for industrial machines and plants

Youichi Nonaka¹, Takahiro Nakano¹, Kenji Ohya¹, Atsuko Enomoto¹, Gábor Erdős^{2,3}, József Váncza^{2,3}, Gergely Horváth^{2,3}

- 1) Hitachi, Ltd., Research & Development Group, Yokohama, Japan
- 2) Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary
- 3) Department of Manufacturing Science and Technology, Budapest University of Technology and Economics

Abstract In the business of industrial machines and plants, rapid and detailed estimates for planning installation or replacement of equipment, or maintenance work are key requirements for meeting the demands for greater reliability and lower costs, and for maintaining safe and secure operation. These engineering services should be supported by specific computer-aided methods. When replacing equipment at complex buildings or plants with high equipment density, the existing state of the installation locations and transportation routes for old and new equipment need to be properly measured. We have met this need by developing a specific object recognition technology based on 3D measurement, and by developing high-speed calculation technology of optimal routes for installation parts. This article provides an overview of these development projects with some real business application results.

1 Introduction

Satisfying the complex web of client requirements and site conditions when constructing a new or retrofitting and old facility such as an elevator system, power plant, chemical plant or oil refinery requires a wide variety of engineering activities. The set of tasks to be accomplished includes environmental assessment, civil engineering and construction, equipment design, equipment procurement, installation, as well as trial operation and handover. Maintenance also requires advanced and detailed engineering work to diagnose component equipment, machinery, and devices, and to repair or replace them as needed to maintain safe and stable operation. When providing engineering services to clients for new plant construction or maintenance, detailed and rapid estimates of costs and work schedules need to be created. To meet these needs, we have developed a portfolio of technologies aimed at supporting more advanced engineering work through use of latest information technology (IT).

This chapter specifically handles retrofitting and replacement of industrial machines and plants. These being between 20 and 50 years old equipment sometimes only have original design drawings in two dimensions (2D), or have undergone so much change over the years that their cables, plumbing or equipment systems have become unrecognizable from the original drawings. Basing a project plan on the existing state is an important requirement in these cases.

Retrofitting and replacement projects consist mainly of removing the items that need to be upgraded, and installing the new equipment. When planning each operation, a key requirement for any plants is to identify the actual state of the surrounding environment to answer questions such as whether there are any obstacles in the transportation routes for removal and installation, or whether reliable connections can be made to existing equipment items. Specifically, the site is surveyed to identify locations to be added or moved, locations that have been transformed by many years of operation, and other site-specific factors. The results of these studies are then used to create structural and process designs.

To increase the efficiency of process ranging from design to installation, and to eliminate the need for skill in site studies, we have worked on using 3D measurement by long-distance contact-free laser scanners to enable rapid measurement of existing site conditions. 3D laser scanners have recently become widely used in fields such as civil engineering, construction, and surveillance. However, measured point clouds contain noise, and since point cloud data is massive, generating as-built models requires extensive manual labour.

In addition, when retrofitting or replacing substation equipment, a large amount of plant-assembled and plant-inspected equipment items are successively installed at the site. The equipment is heavy, so they are lifted by crane for transport, positioning, and connection work. These processes require studying the installation sequence and creating work plans after taking into account difficulties in making equipment parts fit each other in three dimensions, and the temporary placement of equipment delivered to the site. That is, the process of planning the installation of substation equipment items requires a lot of experience and specialized knowledge.

We met these requirements by developing an as-built modelling technology based on three-dimensional (3D) measurement and a technology for planning approaches to retrofit/replacement based on 3D models.

Section 2 reviews related works and studies of the corresponding field. Section 3 focuses on object recognition in existing plants, while Section 4 discusses replacement tasks in plants, during maintenance work. Section 3 uses examples from an elevator shaft repair commission, while illustrations of Section 4 are taken from a plant overhaul project. Section 5 concludes the chapter.

2 Related works and technology trends

In industrial engineering (IE) there is a common understanding that the primary prerequisite of performing maintenance, repair and overhaul (MRO) activities is to have a reliable digital *semantic model* of the overall target object [10,12,26]. Such a model, which is referred to in the closely related field of architecture, engineering and construction (AEC) as *building information model* (BIM) represents the complex industrial facility in terms of its components, along with their geometric and any other relevant properties and relationships. Unlike a traditional computer-aided design (CAD) model it is a semantically rich representation that can be used by a number of different stakeholders of a facility with relatively long life-cycle, including MRO planners and operators, too [18,27]. However, just such industrial objects—like elevators, plants, thermal or nuclear power stations, manufacturing facilities to name a few—have in many cases only two dimensional blue print documentation (if any) [19], and it is rather the rule than the exception that there are mismatches between any kind of model and the reality [8]. Planners responsible for MRO activities have to face risks due to errors of unrecorded modifications, deformations, as well as missing records of incidental equipment such as suspending fixtures and cranes. Tailoring the model to reality time and again is also essential when monitoring the progress of construction projects and registering what has (or has not) been built according to plan or specifications [25]. Hence, it is not enough to create a semantically rich model of the industrial object, but this model should capture its actual *as-is state* [22].

In the past decades, these double requirements opened broad and overlapping research fields both in IE and AEC. With the advancement of computer vision and especially *3D laser scanning* technologies it became possible to scan the surface of even very large-scale objects and to create their as-built representation in terms of a set of points with 3D Cartesian coordinates—a so-called *point cloud* [1,4,11]. In parallel, methods for the efficient storage and retrieval of this often enormous amount of measurement data have been developed, too [23]. However, the *reverse engineering* problem of converting this raw representation into a concise, semantically rich model remained a research challenge till today [11,27,28].

A detailed survey of methods supporting the (semi-) automated *reconstruction of as-built building information models* out of laser-scanned point clouds is provided in [27]. Accordingly, as components of a complex object have shape, identity and relationships, the overall problem involves three essential sub-problems: geometric modelling, object recognition and object relationship modelling. While advance in all the above research areas have been achieved, and even some specialized commercial systems have appeared, so far there has been no integrated, generic solution for building up automatically a structural model of a complex industrial object departing from its point cloud data.

The actual solutions vary with the domain; for instance, in AEC much effort was put into developing methods that can recognize *indoor scenes* [21] and the

most characteristic components of a building, such as floors, walls, cabinets, ceilings, and opening like doors and windows [27,28]. Recognition is typically concentrating on the surface of objects, via polygonal meshes and parametric surface models fitted to the point cloud. Hence, these methods generate models of complex objects in terms of structured surface meshes. A recent development aims at a reconstruction of buildings which maintains room *topology* and global wall connectivity [20]. Having a specific scope on construction sites, [30] tackles the recognition of heavy dynamic construction equipment like crawler cranes. In industrial engineering, the well-proven notion of *features* is applied almost unanimously, whose recognition can depart from a mesh-type input data that is fitted to the measurement point cloud [29]. A recent survey provides a comparative analysis and outlook of academic and state-of-the-art commercial methods capable of recognizing *as-built 3D layout* and in particular, *pipeline systems* in large-scale civil infrastructure facilities by making use of photo- and videogrammetry, as well as terrestrial laser scans [25]. Under practical conditions, due to complexity, noise level and incompleteness of data the methods are hardly applicable without extensive (and expensive) human assistance. Hence, more research is expected, especially in a direct collaboration of academy and industry [25].

Our earlier research aimed at *adapting* the existing CAD model of a complex industrial object to the point cloud measured on its actual surface was just an instance of such a joint study [8]. The workflow included the efficient storage of massive measurement data, segmentation of a triangulated, mesh-based CAD model into features, as well as matching and adapting the features to the data. The method was applied in a real-world setting, using the CAD model and point cloud data of an industrial plant that contained planar and cuboid objects, as well as complex and dense systems of pipes. Continuation of this research led to a method that was capable of reconstructing the *structural model* of as-built industrial facilities purely from on-site point cloud measurement data [9]. Focus was set on finding the internal structure of complex objects hidden behind the massive point cloud by exploiting connectivity information in the data and the linear characteristics of the typical components such as pipes, beams or other structural elements. This novel method, along with examples of its application in a specific domain will be presented in some detail in Section 3 below.

The other main field of research related to our topic is automated *disassembly and assembly planning* which involves in the MRO domain two kinds of sub-problems: (dis)assembly sequence planning, as well as (dis)assembly path planning. Sequence planning concerns the problem of finding a feasible sequence of operations that put (remove) components of a complex object to (from) their place. While the reconstruction of the object's structure is such a prerequisite of sequence planning which can be resolved in specific cases (see above, or [1] in particular), so far there are only few and scattered attempts to derive the sequence of (dis)assembly operations from the model of the object at hand [13]. However, in the past two decades much effort was put into the solution of the problem of transporting components, equipment and other objects in a relatively densely

occupied industrial environment [6]. This latter so-called carry-in and carry-out problem (whose generic version is the classic piano movers' problem) is essentially a task for *path planning* which centers around generating a collision-free path from an initial picking point to a target point in an environment filled with obstacles. Further to accounting for part and obstacle geometries, as well as for physical forces such as gravity or friction, the path has to be optimal according to some criterion like energy, time, safety, etc. [19]. A recent review provides a comprehensive taxonomy and characterization of (dis)assembly path planning problems, together with a well-structured presentation of up-to-date solution approaches [13]. A generic, broadly applied method of path planning in constrained spaces is that of *road mapping*. Even though depending on its actual application domain like robotics, (dis)assembly, unmanned aerial vehicle (UAV) planning, MRO planning, etc., road mapping developed numerous variants, its core concept is based on a network of collision-free configurations where adjacent nodes can simply be reached from each other. Road maps can be pre-computed by probabilistically sampling the space [14] before searching for a solution (by using some classical shortest-path graph search or A* algorithm), or generated on the fly, as in the case of rapidly-exploring random trees. In any case, finding narrow passages between collision-free areas poses a serious challenge, especially for planners operating in a 3D cluttered environment. The key to that issue is the characterization of the space and adapting appropriately the sampling strategy which heavily relies on collision detection. For instance, so as to increase computational efficiency we applied an octree-based voxel representation of the free space, combined with its parallelized exploration [7,19] (for details, see Section 4 below).

Finally, carry-in and carry-out operations are executed typically by *cranes*. While planning the path of such auxiliary equipment is in most of the cases out of the scope of investigations [13], the practically highly relevant problem of mobile crane walking and path planning is tackled in [16]. Here, crane configurations are considered together with typical site constraints and the geometry of lifted equipment. The proposed method determines the pick and (collision-free) operation areas, and then calculates the walking path of the crane. A related problem is *crane lifting* in a complex environment when a collision-free and cost-optimal lifting path is to be generated by considering inputs such as the plant environment, crane mechanical data, crane position, as well as pick and end lifting configurations. An overview of state-of-the-art computer-aided crane lift planning methods is presented in [5], along with a specific genetic algorithm based technique that proved to be highly efficient thanks to its parallelized implementation.

3 Recognition and modelling technology

3.1 The object recognition problem

The object recognition and modelling technology is based on some generic assumptions that ensue from the application domain. First, even though measurement data may be acquired from a number of different positions of the scanner, all points are registered in the same reference coordinate system. The proprietary point cloud data format of the actual scanning system is also transformed to a uniform representation. Multiple scans can to some degree alleviate the difficulties caused by occlusion, but the measurement data remains intrinsically noisy and partial. As for the structure of the scanned object, one can assume that it is constructed out of typically linear extruded elementary components, such as pipes, beams, pillars, walls or cuboid objects. Some of these objects may be even of standard size (like various types of beams). However, we emphasize that the availability of the CAD model of the object is not required.

Hence, the *inputs* of the recognition process are the following:

- 3D point cloud of the measured complex object,
- prior knowledge of the types of its elementary components, and optionally,
- additional information on the exact geometries of the potential elements, such as catalogue of standard beams.

The *result* of recognition process is a compact representation of the measured object consisting of its identified elementary components, together with their actual geometric parameters and their connectivity relations. Furthermore, each point of the cloud has to be indexed either with the components found or marked as unidentified.

The strong engineering motivation of the application implies twofold performance *criteria*: minimizing overall processing time (including manual and computational), and achieving as high recognition accuracy as possible.

3.2 Assumptions and representation

The representation and method developed for transforming the large-scale 3D point cloud data into a structured model of a complex object is based on some fundamental engineering principles.

- *Aggregation* is applied when collecting points of the cloud into a discrete, uniformly sized, 3D grid structure and working with these voxels instead of the points in some of the calculations.
- *Filtering* is used to remove noise from the data.
- *Segmentation* is applied to decompose a larger space investigated into regions of manageable size. The recognition process can run in each region simultaneously, while some overlap between the regions warrants that connectivity information is not lost.

- *Connectivity* of voxels is maintained and exploited so as to recognize topological relations between elementary components whose surface is represented—even partially—by the voxels.
- *Linearity* of elementary components that build up a complex engineering object is assumed.

Hence, the representation has altogether four layers. The basic representation is that of the registered measurement data points, given as coordinates in a common Cartesian system. Next, points are clustered into voxels that are signified by their centroid points. Then, neighboring voxels are captured in a *voxel connectivity graph* (VCG) where nodes denote voxels, while edges stand for any two voxels which are adjoining in space. A region under study is typically covered by a number of disjoint VCGs. Finally, each VCG has also a more refined model where the linearity of components is explicitly exploited. This is a so-called *branch connectivity graph* (BCG) where the nodes stand for typically linear branches composed of specific connected subsets of adjacent voxels of a VCG, while edges represent connections between the branches. The BCG provides a more articulated representation of the measurement data and hints at the presence of typical object types.

3.3 Workflow of object recognition from point cloud data

Specific illustrations for the stages are taken from a case study performed in an industrial domain, where recently the method has been applied routinely for elevator renewal.

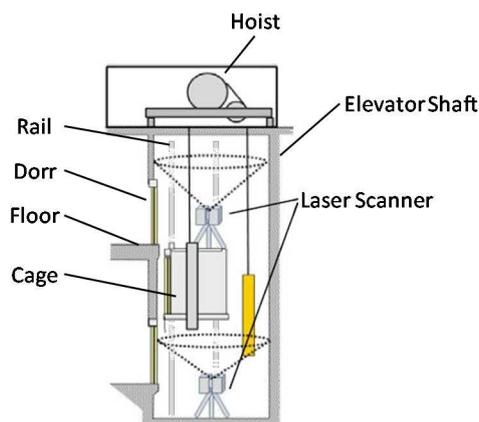


Figure 1 3D laser scanners in elevator shaft (vertical cross section).

Figure 1 presents an image of aged elevator shaft scanning work in the renewal business. 3D laser scanners on the pit and on the cage obtain point cloud data of all the objects involved, such as shaft, rails, doors, cage, and so on, which are deformed and/or tilted by many years of operation. That is, the purpose of object

recognition is to detect objects such as shaft, rails, doors, cage, and quantify those deformation and tilting automatically.

A workflow has been developed for solving the above problem that consists of the stages of *preprocessing*, *point cloud filtering and connectivity graph construction*, as well as elementary *object and connectivity recognition*. Figure 2 presents this workflow, while the next subsections describe in short the major processing stages (for more details, see [9]).

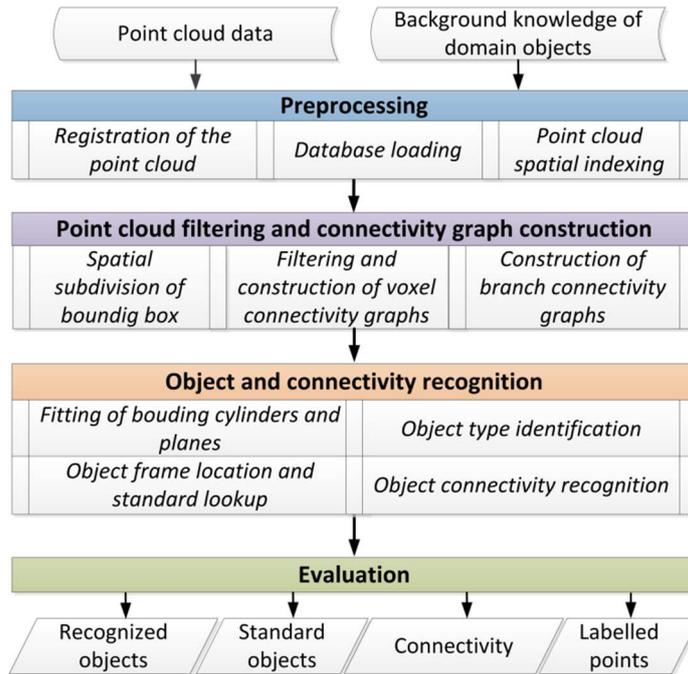


Figure 2 Workflow of the object recognition process.

3.3.1 Preprocessing

First, an affine transformation is performed to register points taken from various scanner locations in a common reference coordinate system. Next, the space is decomposed into 3D *voxels* of a given size. However, efficient storing and querying large datasets containing up to even billions of points requires appropriate indexing schemes and database management techniques. Since the most frequent operation is bounding box query, such a spatial indexing is used that stores data indexed by the basic voxels so that points located close to each other in the actual domain are stored also physically close to each other in the database. The indexing scheme applies an octree-based decomposition of the space [23].

Figure 3 shows the raw point cloud data captured in the shaft of an elevator. This relatively small dataset contains ca. 40 million points.

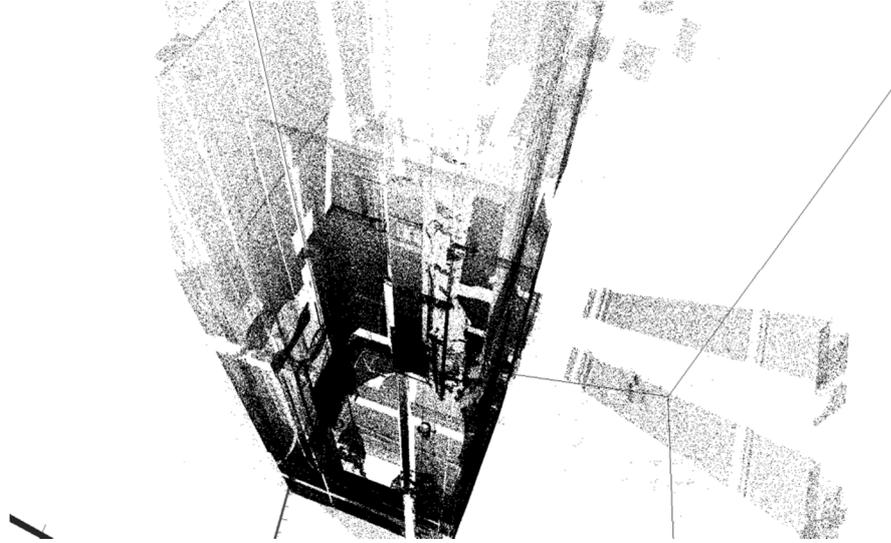


Figure 3 Point cloud data of the elevator shaft.

3.3.2 Point cloud filtering and connectivity graph construction

In order to keep the size of the raw point data as well as the complexity of recognized connected object structures manageable, the space of the complete object is first *segmented* into disjoint spatial regions. Points belonging to the same region are processed together, while data of different regions are processed independently (and, optionally, in parallel). The size of a region depends on the number and granularity of the measured points. The generic rule is that data belonging to one region should fit into the memory of the actual computing facility. For instance, in the actual elevator case study there is no need to segment the space into regions.

The point cloud is collected through a series of onsite measurements, hence due to occlusion, shadowing, and inaccessibility on the one hand, and reflections on the other hand, the data is incomplete and burdened by noise. *Filtering* and *VCG composition* remove the noise from the input data and select such connected subsets of voxels that are good candidates for object recognition. The procedure composes a VCG where both the number of points in each voxel and the number of voxels in each connected component are over some specific, pre-determined thresholds. Voxels (and included measurement points) not meeting any of these criteria are discarded in the course of an iterative filtering process. By interleaving filtering and VCG construction, both scattered and isolated points are removed from further processing. Figure shows the VCGs generated for the elevator data: voxel size is 1x1x1 cm, minimum point density is 25 point/voxel, and at least 500 voxels should be connected. This way one gets ~50 disjoint connected sets of ca. 500.000 voxels.

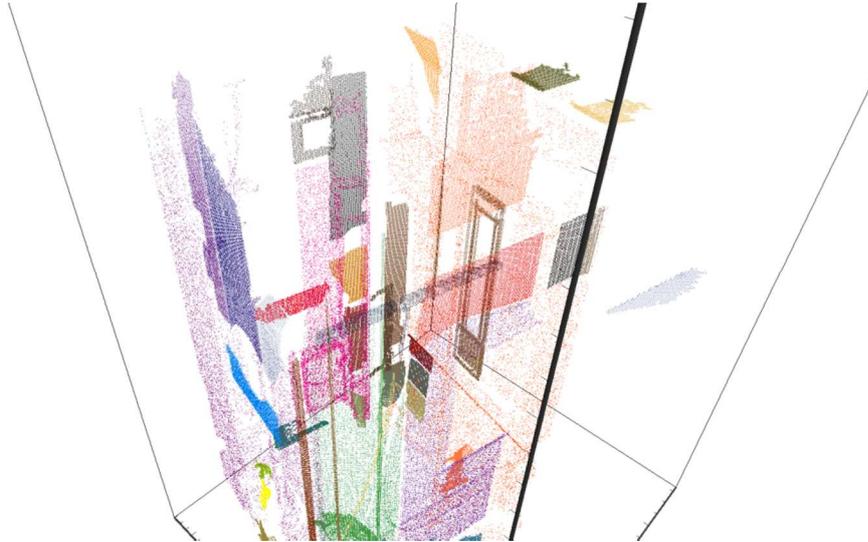


Figure 4 VCGs of the elevator data.

Subsequent steps of the workflow can focus on those areas of the space that are not only densely populated by points, but contain also candidates of large enough compound objects. However, in a VCG some disjoint connected components can be too complex and unstructured for recognition. Hence, these are deconstructed into smaller connected subsets by exploiting the linearity assumption. The typical elementary components are extruded objects like pipes, beams (with various profiles), rails, etc., which can be represented by connected *voxel branches* stretching in some characteristic direction. Extruded beams are typical, planes are special types of such extruded linear objects. The *BCG construction* method applies projection and a specific region growing method to find both quasi-linear structures as well as their connections. First, in a given direction linear arrangements of connected voxels (so-called fibers) are sought with a length over a threshold. Fibers with adjacent voxels form a branch which is augmented with isolated voxels in its immediate proximity. Connectivity of branches that have adjacent voxel pairs is recorded. Finally, after removing voxels of the BCG found so far the procedure is iteratively repeated for other directions. Figure provides two different looks of the BCGs generated for the elevator data (with minimal fibre length of 50 cm). Voxels of the same color belong to the same branch of the BCG.



Figure 5 BCGs of the elevator data depicted in two different views.

3.3.3 Object and connectivity recognition

Object recognition is aimed at identifying and characterizing the elementary components of the complex engineering object. It processes the branches of the BCG one by one. Points belonging to the voxels of a branch are taken as evidence for the existence of some specific object type. Hence, in the next steps measurement points are again directly processed. For instance, Figure presents the BCGs of the whole elevator dataset together with points of five selected linear branches.

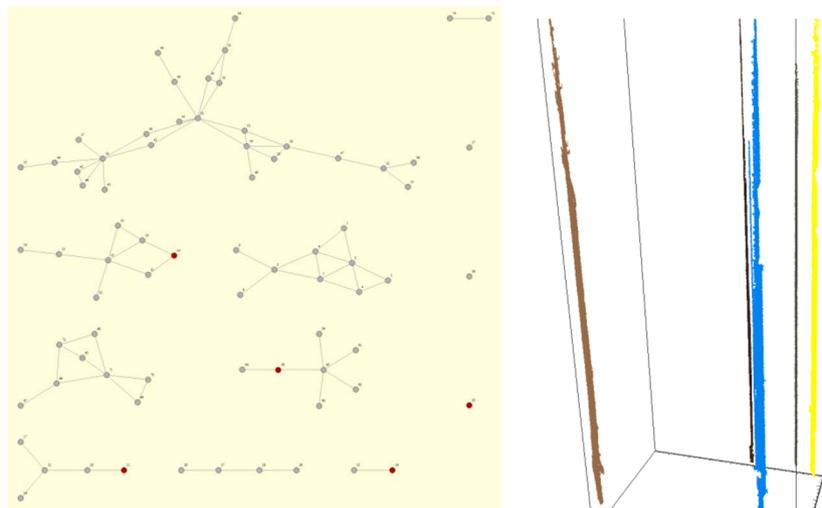


Figure 6 BCGs of the whole dataset and points of five selected branches (denoted by red nodes in the graph).

First, the characteristic *axis* of a branch is found together with its start and end points. Next, the *type of the elementary object* is determined by means of bounding planes fitted to the point set. Plane fitting is executed iteratively until

new planes are found with support of a predefined, minimal number of points. Planes are then projected along the axis, resulting approximate linear contours of the cross section of the supposed object. In principle this should be sufficient for identifying the shape of an extruded object, but due to noise and occlusion, in real datasets these lines are typically multiple and inaccurate. However, essential information on the type of the object can be gained by investigating the intersection of these lines and identifying its shape pattern. Hence, quasi-parallel lines running close together are substituted by a single representative, and the resulting pattern of crossing lines is processed further. After distinguishing cylindrical objects, structural elements such as pillars and beams of various profiles are sought. Their main feature is that they are linearly extruded 3D versions of some 2D linear contour. For identifying the particular object types, a *shape grammar* was developed that labels intersections of contour lines as end or middle points, and suggests a classification.

Having objects with recognized types, our earlier CAD model matching procedure is applied to determine the values of basic parameters, like center line and radius of pipes, or sizes of cuboids. Here, an iterative search maximizes the degree of match of the target object with the relevant set of points. A new method was developed for fitting various types of beams to standard elements of a catalogue. Finally, object recognition is completed with determining the local reference frames of the elementary objects. Figure presents three recognized L-beams in the elevator data set, along with two unidentified branches.

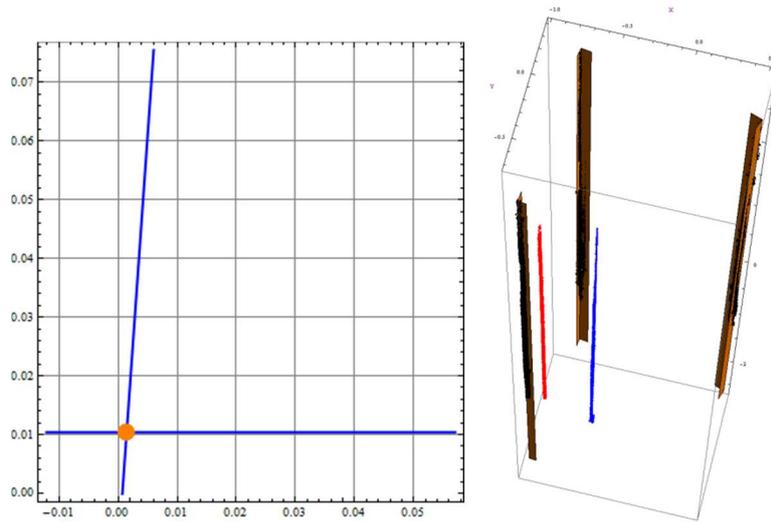


Figure 7 Recognized L-beams by shape grammar. Objects behind the red and blue branches could not be recognized.

In the last step, *connectivity relations* between the recognized elementary components are taken from the BCG. If the object recognition process was unsuccessful, the respective branch is labelled as unidentified. Networks of connected pipes or beams can be obtained by inducing a subgraph of given type of nodes and by determining their connected subgraphs. The final representation is rich enough for making inferences on the connectivity of different types of objects; e.g., one can deduce which beam provides support for a pipe system.

As for the performance of the overall recognition method, the four-layered representation using points, voxels, voxel connectivity and branch connectivity graphs was powerful for recognizing both elementary components and their topological relations. The objects could be recognized even from noisy and incomplete data. In contrast to earlier approaches, our method could identify generic extruded objects, i.e., not only pipes, but also various types of beams and pillars. Routine applications in the elevator domain have shown the practical applicability, while tests run on large plant datasets have proved the scalability of the method whose performance can be improved by parallel processing.

4 Route planning technology for replacement task

4.1 Impact of route planning on replacement task

This section focuses on *route planning* technology for replacement task in renewal/retrofit business of plants. Insufficient route planning prevents competitive price setting that induces to lose business chance.

The route is required to have no collision with plant structures and equipment. For the collision check, conventional 3D CAD systems have been utilized. Engineers have to develop paths manually and the collision check on the CAD system consumes a long time: typically it takes five hours or so for planning a single path for carrying in a boiler into a plant building.

It often turns out that the paths are not feasible on site. In real situation, there are many uncertainties between a maintenance work plan and the real state of the facility, such as worker's availability uncertainties, hoist's availability uncertainties, situation uncertainties of placed materials, and so on. In the maintenance work planning phase, no one can quantify those uncertainties, and no one can make the best carry-out/carry-in path. Therefore, according to those strong requests of real business, the system must produce plural variant paths as candidates for multiple situations.

Replacement objects are carried with overhead cranes because of their weight, which can be over a ton. Suspended components direction is changed by workers at each turning point. This means, fewer turning points are preferable from the view point of cost of the crane operations because the rotations of a suspended component require a long operational time and many travel rails for the overhead

crane. It is also important to have large workspace at turning points to rotate the replacement component avoiding collision.

From these candidate routes, an on-site supervisor decides the *optimal carry-in/carry-out route*. The decision is dependent on the state of maintenance situation. This means the automatic planning system have to calculate the paths in a very short time to establish an interactive operation between the system and the supervisor.

4.2 Subjects of route planning technology

4.2.1 Objective of route finding algorithm

In general, route finding algorithms aim to optimize length of the route [7]. However, typically the shortest route is not sufficient in a plant building. There are two important aspects to find a *suitable route* for carrying a large component either inward or outward a building when performing a replacement task.

The first one is the size of corner space for direction change of the carried component, carried out by maintenance workers. Each corner needs to be large enough to rotate the component without collision; the sizes of components are several meters. Therefore, volume of each corner space on the route is an important indicator to evaluate the route. The volume of corner space is called as *space margin* from now on.

The second issue is the *number of corners* on the route. At each corner, workers rotate the carrying component and the posture of the carrying component is changed. Fewer corners are preferable from the view point of easiness and cost reduction of crane manipulation.

Third indicator is the *time of calculation*, which has to be as short as possible, ensuring the interactive use of the system.

4.2.2 Crane suspension posture

Even if the optimal carry-in/out route is found with respect to space margin and number of corners, feasibility of crane operation can hardly be ensured, because collision may occur on the route. Detailed *trajectory of position and posture* on the route should be simulated in advance, before executing the operation, especially at corners. Corner orientations are crucial as unexpected collisions may occur so crane operations have to be reworked.

4.2.3 Scale of route finding problem

The most fundamental and stable algorithm for route finding is the Dijkstra method [24] which is applicable for many problems by designing the objective function. The fastest *route finding* algorithm ALT [14] is enhanced based on Dijkstra method. Lozano and Wesley introduced an idea of configuration space to route finding algorithm that enables to find posture and position sequence of the mechanism with several degree of freedom [17].

Table 1 indicates the order of computational effort required for solving various characteristic route-planning problems. In every case, the route finding problem is represented on a graph network. Number of vertices in the graph network indicates the scale of the problem.

Table 1 Scale of the route finding problem.

No.	Problem	Vertices n	Computation order ratio $O(n^2)$
1	San Francisco Bay Area map[7]	321,270	1
2	Plant building (3D position)	973,210	9.2
3	Full USA map[7]	23,947,347	5,600
4	Plant building (5D position and posture)	249,141,760	600,000

No.1 and No.3 is solved by ALT algorithm on road maps. Each vertex represents a junction of roads. No.2 is a problem for finding optimum route as position sequence in a plant building. The 3D model of the building is divided into small cuboids whose size is less than 0.125m^3 . Each cuboid is represented as a vertex in the graph network. No.4 expands the problem of No.2 on order to find posture sequence on the route. Configuration space of the crane is of 5 dimensions, adding 2 degrees of freedom (DOF) rotating along 2 axes. The configuration space expands the number of vertices by the multiplication factors of the numbers of possible rotations along two axes. In case of No.4, 16×16 samples for two axes rotation is timed to each cuboid that ends up 249,141,760 vertices in No.4.

No.4 problem is larger than 600,000 times than the problem of No.1. If computation time for one route finding of problem No.1 takes 0.1 sec/route, then the computation time of No.4 is assumed as about one day/route which is not feasible in any business sense.

4.3 Break down of route planning problem

Route finding algorithms working in the configuration space heavily consume computation time. To overcome the complexity of the problem, we also apply here the *principle of decomposition* and breaking down a big problem into smaller sized sub-problems.

Some articles propose the breaking down approach to carry-in/out route planning [2,7,19]. The problem is broken down into 3 stages as below;

- *Stage 1*: In 3D space of point, the optimal route is found maximizing the space margin and minimizing number of corners on the route. These indicators of the objective function of the optimal search aim to ease and increase efficiency of crane manipulation with avoiding collision.
- *Stage 2*: In each corner of the found route, posture and position trajectory is simulated based on the crane suspension dynamics.

- *Stage 3*: Collision between the carrying component and the building is evaluated along the route trajectory. If collision occurs, posture to avoid collision is searched in each point.

The workflow of the route planning system is shown in Figure 8. In Stage 1 and Stage 2, the problem size becomes small enough to be computed in feasible time. However, for Stage 3, its problem size is still far too large. Hence, by exploiting the inherent parallelism of the problem, collision check in Stage 3 is performed by general-purpose multiprocessors, like graphic processing unit (GPU).

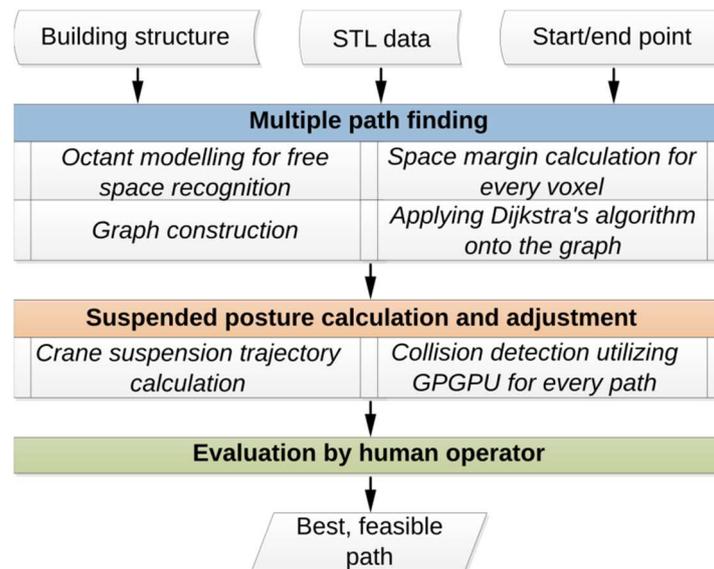


Figure 8 Workflow of path finding.

4.3.1 Route finding algorithm for carry in/out plant building

Requirements are listed in Section 4.2.1. So as to satisfy the requirements the problem is decomposed into a *multiple path finding problem*. The decomposition is achieved by adopting Dijkstra's algorithm for single path finding and constraining travelable path iteratively in descending order of performance indexes of paths finding. Evaluation criteria and their importance were decided through discussion with power plant engineers. Accordingly, the algorithm gives priority to the requirements in the following order:

- Number of turning points*. More turning points imply that the operation becomes more complex and expensive, moreover the production cost is also increased because crane suspension has to travel more.
- Space margin*. The larger space margin causes the operation cost to be lower, especially when making rotation in turning points.

- c) *Path length*. Shorter path length leads to lower operation time, which in turn lowers the operation cost.

According to this key idea, a path planning algorithm has been developed. Figure 9 presents the basic idea of the route finding algorithm for carrying component in /out of plant building. The steps of the algorithm are the following:

- 1) The empty space around the plant's 3D CAD model is divided into voxels.
- 2) In every voxel, the space margin is calculated as the distance from the center of voxel to the nearest point of the building structure. Space margins are registered as main characteristics of the voxels.
- 3) Voxels are divided if their size exceeds a predefined minimal size. Divided voxels inherit the space margin of their original voxel.
- 4) The graph network is structured by the voxels. Here, a node indicates a voxel and an edge indicates adjacency between two voxels. However, it is not allowed to connect two voxels located in upper and lower oblique direction because crane cannot transfer suspended component up or down obliquely. Each node has its space margin registered in Step 3.
- 5) The first path is generated by Dijkstra's algorithm from the given start point toward the given target point.
- 6) So as to get dissimilar route variants, from among the voxels in which the recently found path goes through the voxel with the largest space margin is erased and the graph network is modified accordingly.
- 7) Step 5 and 6 are repeated until all paths of required number are found.

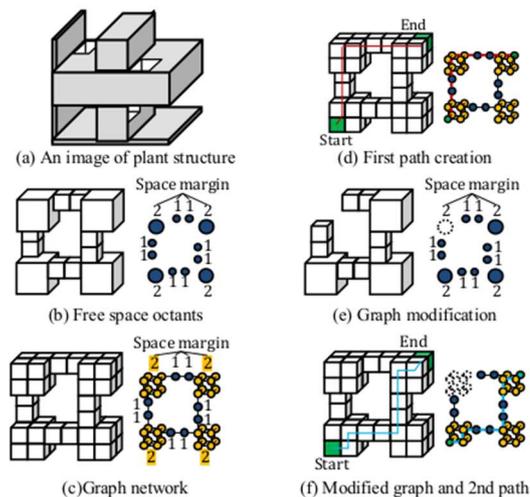


Figure 9 Route finding algorithm for replacement task.

Figure 9 represents the flow of the algorithm with images. The small images show the following steps:

- a) indicates an example of a 3D CAD model.
- b) indicates a result of Step 2
- c) left part is the result of Step 3, while right part shows the graph network of Step 4.
- d) is the result of Step 5. Green boxes denote start and target points.
- e) indicates result of Step 6.
- f) Shows the result of Step 7.

4.3.2 Simulation of crane suspension posture

A *crane suspension* is modeled as the kinematics shown in Figure 10. Two chain blocks are modeled as prismatic pairs which simultaneously expand the length of two wires when the posture of suspended component is changed.

Suspended angles and wire tensions, which are important for safe manipulation, are shown in Figure 11. Operation sequence of the crane is planned by *checking collision and safety via simulation*. Position and posture trajectory is derived from integration of the acceleration solved dynamics model expressed as differential algebraic equations (for details, see [7]).

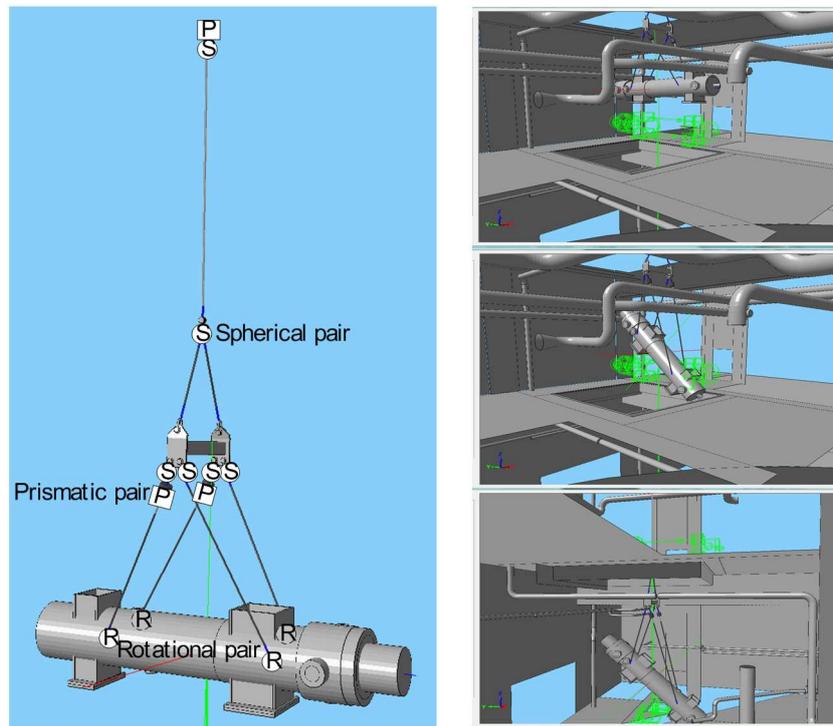


Figure 10 Dynamic model of crane suspension.

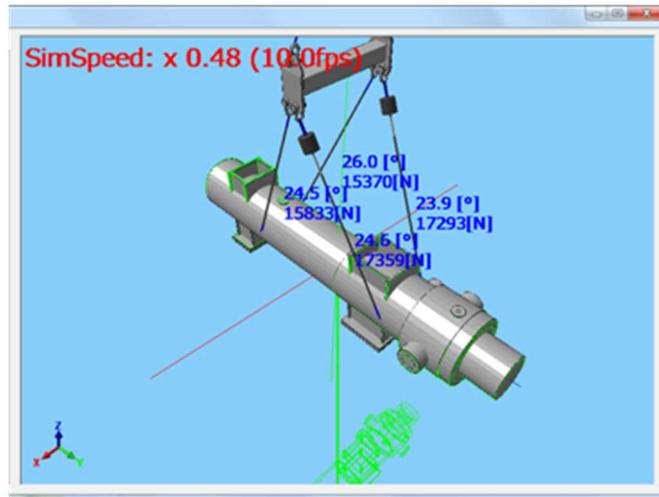


Figure 11 Suspension angles and wire tensions.

4.3.3. Concurrent collision check process with GPU

Collision check is a time consuming geometric computation process. However, it is possible to handle collision check independently for each pose of the suspended component. Therefore, concurrent processing on *GPU* is an effective solution for this problem. Collision evaluation of each interpolated trajectory pose is concurrently computed on GPU [19].

In Figure 12, white circles indicate postures without collisions, and the black circles indicate postures with collision. Each sampling posture rotated around the *x* and *z* axis is evaluated as for collision occurrence by arithmetic processing units called CUDA core® on GPU. Collision checks are processed concurrently therefore the overall time of such tests decreases considerably.

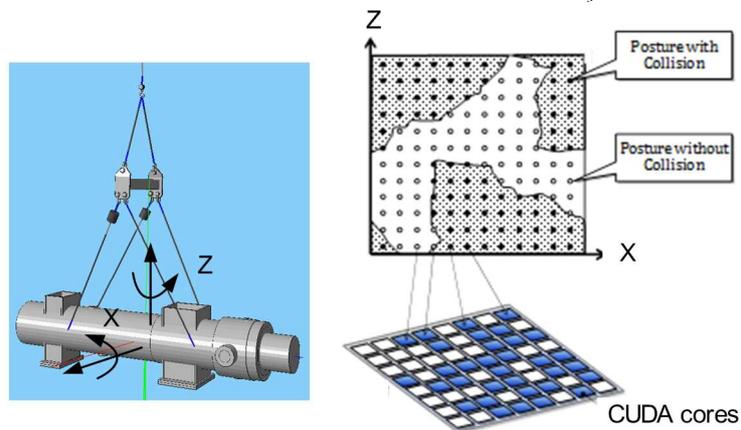


Figure 12 Collision check with CUDA core.

4.3.4 Solution examples of route planning

Below, two plants and characteristics results of route planning are shown as examples. Specifically, Figure 13 presents the models of two plants together with a selected, executable path in each plant.

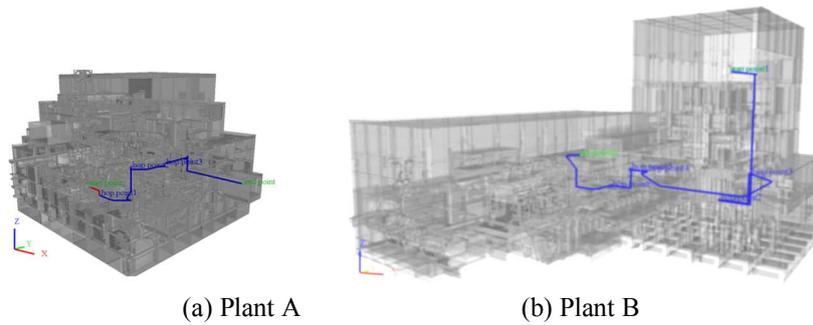


Figure 13 Acquired routes.

Table 2 presents the corresponding total computation times of the verified routes. The total time for generating a route includes route finding, position and posture trajectory generation and collision check. Voxelization and the way how the graph network was built up resulted in a search space which was drastically reduced. Hence, in both cases of industrial size and complexity, the planning methods proved to be efficient and practically applicable.

Table 2 Total computation time.

Plant	Vertices in graph network	Total computation time (sec)
A	262,144	13.7
B	2,097,152	142.5

4.3.5 Results

The proposed planning approach greatly supports the maintenance work of technical equipment both of smaller (like elevators) and larger (like power plants) scale when carry-in/carry-out tasks have to be planned and performed. The method fulfils the following key engineering requirements:

- The route planning algorithm computes one path in less than a minute.
- The algorithm finds multiple, dissimilar paths, considering the space margins and the number of turning points.
- The algorithm generates pose trajectories that avoid collisions around turning points.

5 Conclusions

This chapter presented the latest object recognition and planning technologies we have developed for supporting the retrofitting and maintenance works of complex technical objects. These technologies are used for reverse engineering in conformance with site surveys, and for preliminary engineering in the service of construction works by means of plant models. The methods are now being routinely applied to elevator systems and thermal power plants, and trial use has started for substation replacement projects.

As IT functions become more advanced, recognizing worker behaviours in addition to objects will become practical, and it will be important to manage the progress of complex retrofit/replacement projects in real-time with IT systems. These advances will enable higher utilization rates and longer equipment life, enabling highly efficient construction and maintenance of safe and reliable social infrastructure platform. We will continue to develop technologies to meet this objective.

Acknowledgments

The Hungarian authors thank for the support of the Hungarian Scientific Research Fund (OTKA), Grant No. 113038.

References

1. Adolphy S, Grosser H, Kirsch L, Stark R (2015) Method for automated structuring of product data and its applications. *Procedia CIRP* 38:153-158.
2. Okabe A, Enomoto A, Fujii N, Nonaka Y, Rasch J, Schulte S, Engelhardt M, Kolibabka J, (2016) Automatic route-finding with non-monotonic-trend factor for large-scale plant maintenance task, *Procedia CIRP* 41:550-555.
3. Arayici Y, Hamilton A, Gamito P (2006) Modelling 3D scanned data to visualise and analyse the built environment for regeneration. *Surveying and Built Environment* 17(2):7-28.
4. Bosché F (2010) Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics* 24(1):107-118.
5. Cai P, Cai Y, Chandrasekaran I, Zheng J (2016) Parallel genetic algorithm based automatic path planning for crane lifting in complex environments. *Automation in Construction* 62:133-147.
6. Chang H, Li TY (1995) Assembly maintainability study with motion planning. In *Robotics and Automation, Proceedings of the 1995 IEEE International Conference*, Vol. 1, pp. 1012-1019.
7. Enomoto A, Fujii N, Nonaka Y, Rasch J, Schulte S, Engelhardt M, Shibuta K (2015) Multiple path finding system for replacement tasks. *Procedia CIRP* 33:3-8.
8. Erdős G, Nakano T, Váncza J (2014) Adapting CAD models of complex engineering objects to measured point cloud data. *CIRP Annals - Manufacturing Technology* 63(1):157-160.
9. Erdős G, Nakano T, Horváth G, Nonaka Y, Váncza J (2015) Recognition of complex engineering objects from large-scale point clouds. *CIRP Annals - Manufacturing Technology*

- 64(1):165-168.
10. Farnsworth M, Tomiyama T (2014) Capturing, classification and concept generation for automated maintenance tasks. *CIRP Annals - Manufacturing Technology* 63(1):149-152.
 11. Fathi H, Dai F, Lourakis M (2015) Automated as-built 3D reconstruction of civil infrastructure using computer vision: achievements, opportunities, and challenges. *Advanced Engineering Informatics* 29(2):149-161.
 12. Geng J, Tian X, Bai M, Jia X, Liu X (2014) A design method for three-dimensional maintenance, repair and overhaul job card of complex products. *Computers in Industry* 65(1):200-209.
 13. Ghandi S, Masehian E (2015) Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design* 67:58-86.
 14. Goldberg, A. V. (2005-10) Efficient point-to-point shortest path algorithm. Technical Report MSR-TR-2005-132
 15. Hsu D, Kavradi LE, Latombe JC, Motwani R, Sorkin S (1998) On finding narrow passages with probabilistic roadmap planners. In *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pp. 141-154.
 16. Lei Z, Han S, Bouferguène A, Taghaddos H, Hermann U, Al-Hussein M (2014) Algorithm for mobile crane walking path planning in congested industrial plants. *Journal of Construction Engineering and Management* 141(2):05014016-1-10.
 17. Lozano-Pérez T, Wesley MA (1979) An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM* 22(10):560-570.
 18. Németh I, Püspöki J, Haraszko Cs, Mátyási Gy, Nagy T, Freeman Ch, Scott RW, Baldwin JS (2013) 3D design support for rapid virtual prototyping of manufacturing systems. *Procedia CIRP* 7:431-436.
 19. Nonaka Y, Enomoto A, Fujii N, Kolibabka J, Rasch J, Schulte S, Engelhardt M, Kaneko J, Ichijo T (2014) Fast path finding system with GPGPU computing for replacement tasks in plant maintenance. *Procedia CIRP* 22:23-32.
 20. Ochmann S, Vock R, Wessel R, Klein R (2016) Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics* 54:94-103.
 21. Oesau S, Lafarge F, Alliez P (2014) Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing* 90:68-82.
 22. Pătrăucean V, Armeni I, Nahangi M, Yeung J, Brilakis I, Haas C (2015) State of research in automatic as-built modelling. *Advanced Engineering Informatics* 29(2):162-171.
 23. Schön B, Mosa ASM, Laefer DF, Bertolotto M (2013) Octree-based indexing for 3D pointclouds within an Oracle spatial DBMS. *Computers & Geosciences* 51:430-438.
 24. Schrijver, Alexander (2002) *Combinatorial Optimization Volume A*, pp.96-106, Springer
 25. Son H, Bosché F, Kim C (2015) As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey. *Advanced Engineering Informatics* 29(2):172-183.
 26. Stark R, Grosser H, Müller P (2013) Product analysis automation for digital MRO based on intelligent 3D data acquisition. *CIRP Annals - Manufacturing Technology* 62(1):123-126.
 27. Tang P, Huber D, Akinci B, Lipman R, Lytle A (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction* 19(7):829-843.
 28. Xiong X, Adan A, Akinci B, Huber D (2013) Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction* 31:325-337.
 29. Wang J, Gu D, Yu Z, Tan C, Zhou L (2012) A framework for 3D model reconstruction in reverse engineering. *Computers & Industrial Engineering* 63:1189-1200.
 30. Wang C, Cho YK (2015) Smart scanning and near real-time 3D surface modeling of dynamic construction equipment from a point cloud. *Automation in Construction* 49:239-249.