Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Recognition of complex engineering objects from large-scale point clouds

Gábor Erdős[a,c], Takahiro Nakano (3)[b], Gergely Horváth[a,c], Youichi Nonaka (3)[b], József Váncza (1)[a,c]

[a] *Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary*
[b] *Hitachi, Ltd., Research & Development Group, Yokohama, Japan*
[c] *Dept. of Manufacturing Science and Technology, Budapest University of Technology and Economics, Budapest, Hungary*

This work was aimed at reconstructing the structural model of as-built industrial facilities like plants purely from on-site point cloud measurement data. 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. A workflow is presented with emphasis on data filtering, connectivity graph construction, as well as the recognition of elementary objects and their relations. Results are demonstrated using data of an industrial case study.

Computer aided design (CAD), Object recognition, Point cloud

1. Introduction

The motivation of this research was provided by industries that construct, maintain, and retrofit complex engineering objects like plants or manufacturing facilities. When operating in industrial plants, such as thermal and nuclear power stations or factories, the route plans for transporting pipes, equipment and other objects into the plant area are typically prepared manually by engineers who can only refer to 2D blueprints [1]. Hence, workers of maintenance, repair and overhaul (MRO) activities must face risks due to flaws rooted in unrecorded modifications, deformations and missing accounts of incidental equipment such as suspending fixtures and cranes. In general, the essential condition of performing MRO tasks is to have an accurate model of the overall object [2][3]. Even though models are available, there is always a mismatch between model and reality [4], and often no models are available at all. By making use of up-to-date laser scanning technology, huge amount of data can be collected which refer to the surface of objects [5]. This way, one can also build quasi-volumetric models of industrial equipment [6]. However, creating an appropriate *structural model* out of this measurement data is a tedious, mostly manual and time consuming process. For a complex object such as a power plant, model construction from *point clouds* may take several months. Hence, there is a need of a computer-aided *reverse engineering* process that supports and accelerates this activity [5].

This research has a precursor work that was aimed at matching the existing CAD model of a complex engineering object to the point cloud measured on its actual surface [4]. Now, the primary *goal* is to develop generic technologies for constructing explicit structural volumetric models of such objects from big, noisy and unstructured sets of data. The current work is much more concerned with recognizing the typical components and their hidden *topology* than reconstructing the surface features of measured objects. A compact, semantically rich geometric model of the object at hand is sought that complies with the background knowledge of the problem domain.

Such a semantic model is a prerequisite of making inferences in MRO activities [3]. Recently, in production engineering there have been developed semi-automatic methods for identifying the structure of assemblies containing also complex geometries [2], by using technologies of laser scanning and industrial computed tomography [7]. Research is driven by similar motivation also in building information modelling (BIM) where objects like walls, floors, ceilings and openings are to be recognized as far as possible without manual intervention [8][9][10]. Recognition is typically concentrating on the *surface* of objects, via polygonal meshes and parametric surface models fitted to the point cloud. Hence, state-of-the-art methods generate models of complex objects in terms of structured surface meshes [11]. In contrast, the main novelty of the method presented here is that it looks for and exploits the topology of a complex engineering object that is underlying its representative point cloud.

2. Problem statement

The developed object recognition method rests on a few generic *assumptions*. First, the point cloud—even if data is taken from a number of different scanner positions—is registered. It is also supposed—and in some stages of the recognition workflow also exploited—that the complex object is assembled from linear extruded elementary objects such as pipes, beams, pillars, or even from walls and cuboid objects. However, the availability of the CAD model of the object is not assumed.

The *inputs* for the recognition process are (1) a 3D registered point cloud of the measured complex object, (2) prior knowledge of the types of its elementary components, and optionally, (3) additional information on the exact geometries of the potential elements (catalogue of standard beams) may be available, too.

The *results* of recognition should be a compact representation of the measured object consisting of (1) its identified elementary objects, (2) the actual geometric parameters of these components, as well as (3) their connectivity relations. Furthermore, (4) each point of the cloud has to be indexed either with the components found or marked as unidentified.

Because of the industrial motivation, the main performance *criteria* are twofold: (1) reducing the overall processing (manual and computational) time by increasing the level of *automation*, as well as (2) achieving as high as possible recognition *accuracy*, even in face of partial or noisy data. Note that the evaluation of results in the target domains requires also a historical perspective and human introspection.

3. Workflow of object recognition from point cloud data

The problem statement implies a number of *challenges*. The point cloud data which is typically in the proprietary format of a particular scanning system should be transformed to a uniform representation. Because of the sheer size of the data (in the order of 1000 million points, hundreds of GB), efficient storage and query call for special indexing and database management solutions. The point cloud is collected from the results of a series of on-site measurements: due to occlusion, shadowing, and inaccessibility, it is inevitably partial and noisy. Even without clutter and occlusion, the area may contain objects that do not really meet the linearity assumption. Furthermore, objects assembled of linear elements may be without any characteristic direction, like a meandering system of bended pipes. Finally, background knowledge of the actual domain should be represented in a way that is, on one hand, amenable for automatic computations, and, on the other hand, meaningful for the users of the object recognition method.

A *workflow* has been developed for solving the problem, with specific regard to the above challenges. Fig. 1 presents this workflow, while the subsequent sections describe in short the key principles and ideas of the processing stages. Technical details of preprocessing are not elaborated here.

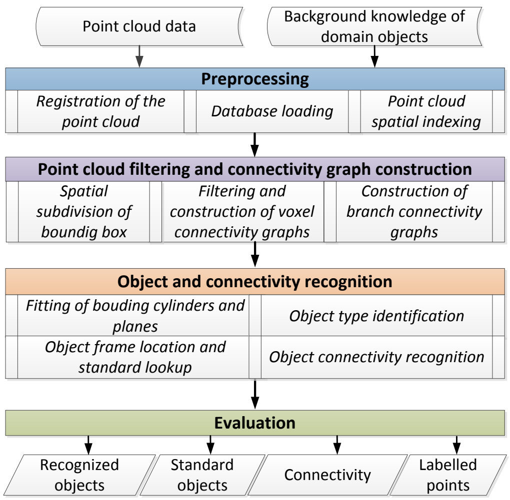

Figure 1. Workflow of the object recognition process.

4. Principles and representations

The representations used for transforming the 3D point cloud into a structured model of a complex object are based on some general engineering principles. Firstly, *aggregation* is used when collecting points of the cloud into a discrete, uniformly sized 3D grid structure and working with these *voxels* (or their centroids) instead of points in some calculations. Furthermore, voxels with low point density are *filtered* out. Often, it is still impossible to process data of the complete investigated area at one time. Hence, the area is *decomposed* into *regions* of manageable size. The recognition process can run in each region simultaneously. However, in order to retain connectivity information, there is a slight overlap between the adjacent regions; common voxels on the borders are processed in each respective region. Next, it is assumed that topological relations between elementary objects can be originated in the connectivity of their corresponding voxels. Hence, voxels in close proximity are represented in a *voxel connectivity graph* (*VCG*) where nodes denote voxels and an edge stands for any two voxels which are adjoining in space. Any region under study is typically represented by a *VCG* of disjoint subgraphs.

The final principle exploits that complex objects are built of basically *linear* components. Hence, each *VCG* has also a more refined alternative model where so-called branches and their connectivity are represented. In such a *branch connectivity graph* (*BCG*) the nodes stand for branches composed of specific connected subsets of adjacent voxels of a *VCG*, while edges represent connections between branches. Figure 2 below presents the *VCG* and *BCG* of a sample region used as working example throughout this paper: this region of 2x2x1m includes over 8 million points. With a voxel size of 1 cm the corresponding *VCG* has c.a. 80 000 voxels. In Fig. 2a, voxel structures in different colors stand for connected subsets of the *VCG*, while in Fig. 2b color coding distinguishes various branches of the corresponding *BCG*. The *BCG* provides a more articulated representation of the measurement data and hints already at the presence of typical object types. Sect. 5 describes how these graphs are generated from the initial point cloud, while Sect. 6 deals with the recognition of elementary objects.

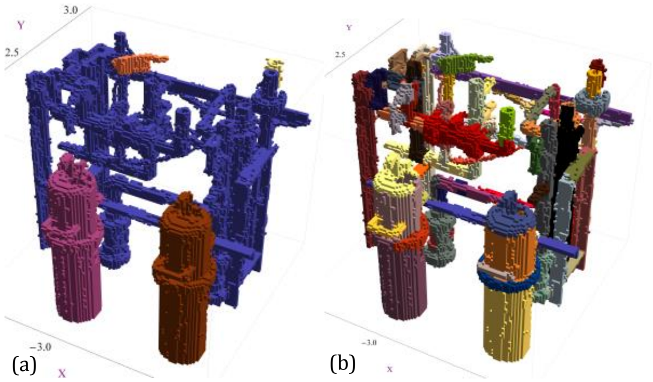

Figure 2. *VCG* (a) and *BCG* (b) built over the data of the sample region.

5. Point cloud filtering and connectivity graph construction

5.1. Filtering and VCG composition

Filtering and *VCG* composition are aimed at (1) removing the noise from the input data and (2) determining the connected subsets of voxels that are good candidates for object recognition. The procedure composes a *VCG*, where (1) the amount of points in each voxel is over a threshold, and (2) the number of voxels of any connected components in the *VCG* exceed a critical limit. Voxels (and included measurement points) not meeting the above conditions are removed. Since this may change connectivity, the procedure is repeated iteratively. By interleaving filtering and *VCG* construction, both *scattered* and *isolated* points are removed from further investigations. Hence, the procedure focuses the subsequent stages of the workflow on those areas of the space that are not only densely populated by points, but contain also candidates of large enough complex structures.

5.2.Construction of branch connectivity graph

VCG construction generates disjoint connected components some of which are too complex and large for further processing (like the blue *VCG* component in Figure 2a). Hence, these constructs are *disassembled* with two goals in mind: (1) to cut the *VCG* into smaller connected subsets that could be passed as input for object recognition, and (2) to restore the connectivity of these components of a complex *VCG*. The underlying idea of disassembly is that in the relevant engineering domains such components are *extruded* objects like pipes, beams of various profiles, etc., which can be represented by connected voxel branches stretching in some characteristic direction. Planes are taken as extreme instances of such extruded linear objects.

The *BCG construction* method takes an external view and applies *projection* to find both quasi-linear structures as well as their connections through the following steps: (1) In a given direction linear arrangements of connected voxels—so-called voxel fibers—are sought with a length over a threshold. (2) Voxel fibers with adjacent voxels are unified into a branch. (3) Each branch is augmented with the isolated voxels in its immediate proximity. (4) The procedure is iteratively repeated for other directions. (5) After removing linear branches found this way, the remaining connected voxels are grouped. (6) Branches overlapping to a great extent are unified. (7) Finally, connectivity of branches that have common voxels is recorded. Fig. 3a provides a closer look at some unified voxel fibers, while Fig. 3b shows an example of augmenting a branch (red) with voxels (yellow). Note that the procedure is akin to *region growing* commonly used in the recognition of surface models [9].

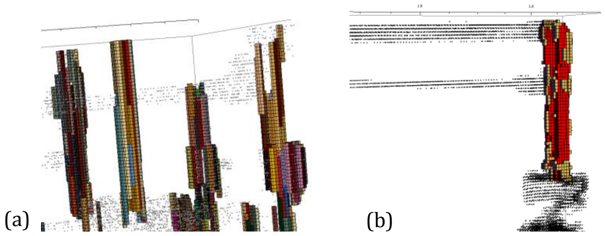

Figure 3. Unifying voxel fibers (a) and augmenting a branch (b).

Connectivity information of branches is checked and recorded by taking the intersection of the sets of voxels belonging to two branches. For instance, Fig. 4 presents the resulting graph of the *BCG* of the complete sample region shown in Fig. 2b.

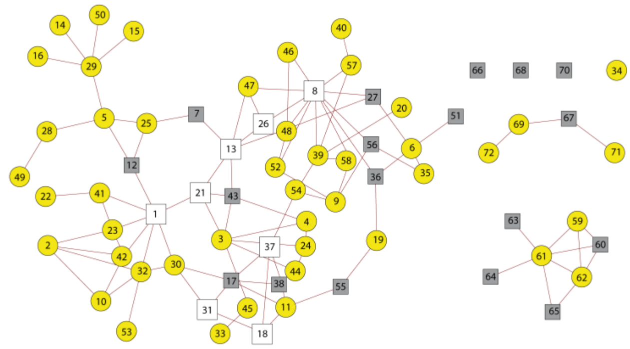

Figure 4. *BCG* of the sample data. For the identification of object types (rectangle: beam, circle: pipe, gray rectangle: unidentified) see Sect. 6.2.

6. Recognition of objects and their relations

The goal of *object recognition* is to identify and characterize the elementary objects that build up a complex assembly. It processes 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 subsequent steps points of the 3D cloud are again processed. Finally, connectivity relations between the recognized objects are taken directly from the *BCG.*

6.1. Fitting of bounding cylinders

This step is aimed at finding the characteristic *axis* of a branch. By definition, the *axis of the point set* is a direction vector such that minimizes the area of points projected along it. Intuitively, the axis minimizes the shadow of points on the projection plane. Practically, this area is calculated by counting the number of black pixels on a projected bitmap picture. Given the found *z* axis of a point set, a *bounding cylinder* is fitted to the points so as to obtain the start and end points of the axis, as well as a convex volumetric region that approximates the shape of the branch.

6.2. Recognition of object types

Object type recognition is performed via determining the *bounding planes* of a point set. Now, a number of planes parallel to the axis are fitted around the point set. Plane fitting is executed iteratively until new planes are found with support of a predefined, minimal number of points. Next, planes are 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 [10]. Hence, quasi-parallel lines running close together are clustered and are substituted by a single representative, and the resulting pattern of crossing lines is processed further.

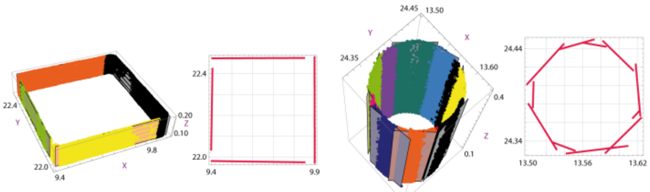

Figure 5. Object type recognition by projected planes.

First, *cylindrical objects* are looked for by visiting vertices of the projected polyline in a fixed order. If the difference between the angles at subsequent vertices remains below a threshold throughout this whole tour, then the object is considered to be cylindrical. Since the cylindrical objects' cross section may vary over the whole branch, cylinders are recognized in smaller slices that are cut along the axis of the branch (see Fig. 6). Otherwise, the branch is categorized as an extruded object with a non-cylindrical profile.

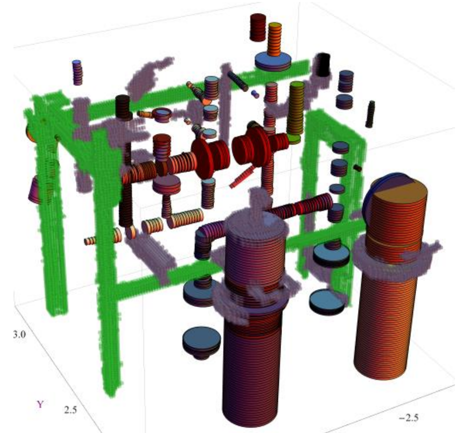

Figure 6. Recognition of cylindrical objects in the sample region.

Typical structural linear objects in the problem domain are also pillars and beams of various profiles. Their main feature is that they are *linearly extruded* 3D versions of some 2D linear contour. For identifying the particular object types, the contour lines of the projected planes are used, similarly to cylinder recognition.

Having objects with identified types, the earlier developed 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 segment of the point cloud (for details, see [4]). Object recognition is completed by determining the local reference frames of the elementary objects. Finally, if a catalogue of standard elements is available, various types of extruded objects are fitted to them.

6.3. Object connectivity recognition

In the last step, nodes of the *BCG* are labeled with the recognized elementary objects. Alternatively, if the object recognition process was unsuccessful, the respective branch is labelled as unidentified (see Fig. 4 earlier). Networks of connected objects can be obtained by inducing a subgraph of a given type of nodes. Fig. 7 highlights the *BCG* of two networks of connected pipes in the sample region, together with the original points and the recognized elementary objects. The final representation is rich enough for distinguishing complex structures, like a system of connected pipes of varying diameters.

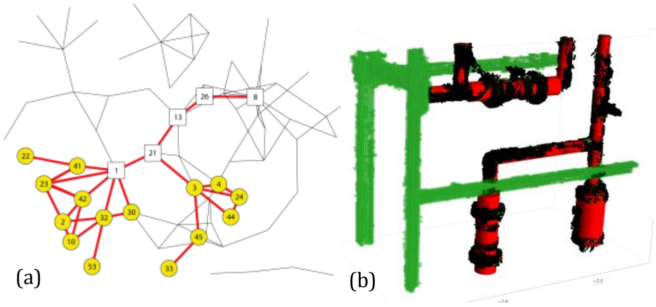

Figure 7. Graph (a) and object models (b) of two networks of recognized pipes, together with data points (excerpt from the sample region).

7. Implementation and experimental results

The implemented object recognition system operates by using an *MS SQL Server 2012* database that handles the input data. Voxelization has been done by applying the *Point Cloud Library* (PCL), and all processing and visualizations methods have been implemented in *Wolfram Mathematica v8.0*. The tests have been run on a virtual server machine with 6 core Intel Xeon X5650 2,67 GHz processor, 6 GB RAM. Experiments have been performed with several large-scale industrial datasets in different domains.

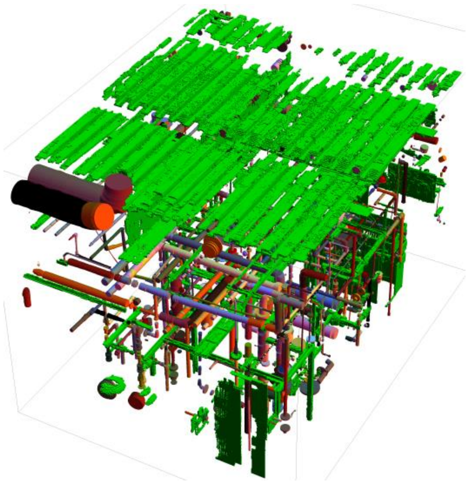

Figure 8. Recognized cylindrical objects of the water equipment plant.

Here results are summarized for a test case of a pure water equipment plant that was originally presented in an earlier related work that was aimed at matching an existing CAD model to a point cloud [4]. Data over 200 million points has been decomposed into 45 regions. Fig. 8 depicts the cylindrical objects found, while main statistics of the recognition process are summarized in Table 1. Two kinds of computational times are given: the *Serial processing* column contains processing times provided calculations are done in each region one after the other, while timing data in the *Parallel processing* column are given with the assumption that after loading the complete dataset object recognition in each of the 45 regions can run in parallel. Hence, these cells contain the worst-case data per each phase regarding all regions. The prior method that matched the CAD model against the point cloud revealed that 5% of the CAD elements was surely not part of the as-is plant [4], and using only the point cloud the current experiments recognized 78% of the components of the CAD model, together with their connectivity.

Table 1. Statistics of the recognition process and results of the test plant.

3D point cloud data	*Serial processing*	*Parallel processing*
Number of points [-]	213 455 636	
Bounding box [m x m x m]	{{-4.25,0.75},{-0.75,5.25},{0,4.5}}	
Database preproc. time[min]	185	
Database reading time [s]	1517 (≈25 min)	89
Filtering time [s]	968 (≈16 min)	58
BCG construction time [s]	3700 (≈62 min)	406
Number of branches [-]	1807	
Accuracy vs. CAD model [%]	78 + 5	
Object recognition time [min]	1459 (≈24 hour)	141
Overall processing time [min]	**1562 (≈26 hour)**	**148**

8. Conclusions and future work

This research resulted in an object recognition workflow that comprises of the following main stages: (1) data preprocessing, (2) data filtering and connectivity graph construction, and the (3) recognition of elementary objects. The representations based on voxel and branch connectivity graphs were powerful enough for capturing topological relations of elementary objects that build up a complex engineered construct. Thanks to advanced graph matching algorithms [12], complexity of handling very large datasets could also be controlled, and the internal structure of measured objects behind pure point cloud data could be restored. The objects could be recognized even from noisy and partial data. Contrary to traditional technologies, the novel method is able to recognize connected systems of generic extruded objects. Tests run on large plant datasets have shown the practical applicability and scalability of the method whose performance can further be improved by exploiting the potential in parallel processing.

Acknowledgement

The Hungarian authors thank for the support of the NFÜ grant No. ED_13-2-2013-0002 and the OTKA grant No. 113038.

References

[1] Nonaka Y, Enomoto A, Fujii N, Kolibabka J, Rasch J, Schulte S, Engelhardt M, Kaneko J, Ichijo T, Shibuta K (2014) Fast Path Finding System with GPGPU Computing for Replacement Tasks in Plant Maintenance. *Procedia CIRP* 22:23-32.
[2] 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.
[3] Farnsworth M, Tomiyama T (2014) Capturing, Classification and Concept Generation for Automated Maintenance Tasks. *CIRP Annals - Manufacturing Technology* 63(1):149-152.
[4] 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.
[5] 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.
[6] Németh I, Püspöki J, Haraszkó 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.
[7] De Chiffre L, Carmignato S, Kruth J-P, Schmitt R, Weckenmann A (2014) Industrial Applications of Computed Tomography. *CIRP Annals - Manufacturing Technology* 63(2):655–677.
[8] 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.
[9] 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.
[10] 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.
[11] Curkovic M, Vucina D (2014) 3D Shape Acquisition and Integral Compact Representation Using Optical Scanning and Enhanced Shape Parameterization. *Advanced Engineering Informatics* 28(2):111-126.
[12] Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) *Data Structures for Disjoint Sets.* In *Introduction to Algorithms*, pp. 498–524. MIT Press, 2nd ed.