CIRP Annals - Manufacturing Technology 64 (2015) 165–168

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: http://ees.elsevier.com/cirp/default.asp

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] Department of Manufacturing Science and Technology, Budapest University of Technology and Economics, Budapest, Hungary

ARTICLE INFO

Keywords:
Computer aided design (CAD)
Object recognition
Point cloud

ABSTRACT

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.

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]. In 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 complex geometries [2], by using technologies of laser scanning and industrial computed tomography [7]. Research is driven by similar motivation also in building information modeling (BIM) where objects like walls, floors, ceilings and openings are to be recognized as far as possible without manual intervention [8–10]. 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

* Corresponding author at: Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary. Tel.: +36 12796299; fax: +36 14667503.
E-mail address: vancza@sztaki.mta.hu (J. Váncza).

http://dx.doi.org/10.1016/j.cirp.2015.04.026
0007-8506/

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 (catalog 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.

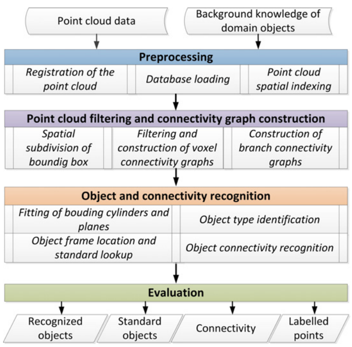

Fig. 1. Workflow of the object recognition process.

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 that 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. Fig. 2 presents the *VCG* and *BCG* of a sample region used as working example throughout this paper: this region of 2 × 2 × 1 m 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. Section 5 describes how these graphs are generated from the initial point cloud, while Section 6 deals with the recognition of elementary objects.

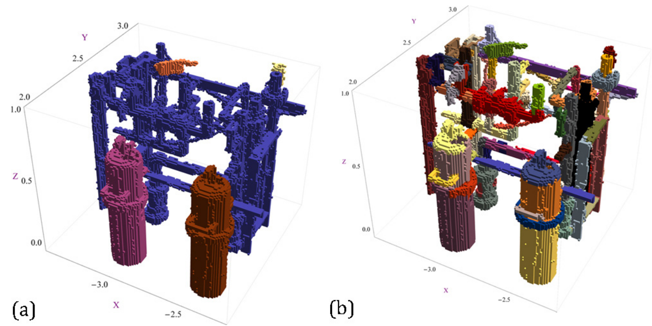

Fig. 2. *VCG* (a) and *BCG* (b) built over the data of the sample region.

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* exceeds 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 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 Fig. 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