Title:
Extraction and Reconstruction of Articulated Robots from Point Clouds of Manufacturing Plants

Authors:
Kota Kawasaki, k.kawasaki.uec@gmail.com, The University of Electro-Communications
Kakeru Takeda, k.takeda@uec.ac.jp, The University of Electro-Communications
Hiroshi Masuda, h.masuda@uec.ac.jp, The University of Electro-Communications

Keywords:
Point-Clouds, Shape Reconstruction, Industrial Robots, Terrestrial Laser Scanners, Point Processing

DOI: 10.14733/cadconfP.2024.131-135

Introduction:

Recent advances in laser scanners have made it possible to obtain 3D geometric information about the current state of manufacturing plants. By using a Terrestrial Laser Scanner (TLS), 3D data of a manufacturing plant can be acquired as point clouds of hundreds of millions to billions of points. Point clouds are useful for constructing a faithful virtual environment and simulating production lines in the virtual environment. In particular, offline robot teaching based on a virtual environment is very effective in assembly factories where many robots work together. However, point clouds include various objects, such as floors, machine tools, safety fences, and robots. In order to use point clouds for the simulation of robot operation, it is necessary to detect and segment such objects from point clouds. It is especially important to classify moving objects and fixed objects; moving objects include robots, robot guns, wire harnesses, etc., and fixed objects include workbenches, stands, toolboxes, floors, fences, etc.

In robot motion simulation, if robots can be identified from the point cloud, the virtual environment can be created as a point cloud excluding moving objects, and collision checks between the 3D models of robots and the virtual environment can be performed. Furthermore, points of attached parts and wire harnesses can be obtained as neighbor points of robot points. Since these points are often not included in the 3D models of robots, a precise collision check can be performed by adding these points to 3D models of robot links and simulation robot motion.

Given a 3D model of an object to be extracted from a point cloud, the iterative closest point (ICP) algorithm [1] and the principal component analysis (PCA) [3] can be used to identify regions in the point cloud that fit the 3D model. However, these methods are not applicable when the shape of the object varies significantly with its posture, as in the case of a robot with articulated arms.

In this paper, we propose a method for extracting articulated robots in arbitrary postures from point clouds of manufacturing plants. 3D models of robots are available from the manufacturer in most cases, and the connection relations of the robot links can be described using the unified robot description format (URDF). These data are commonly used in robot simulations such as ROS (Robot Operating System). In our method, the CAD model and the URDF data of each robot are used as prior knowledge. By identifying robot links in the point cloud, objects that move with the robot, such as robot guns, wire harnesses, and other accessories, are detected as neighborhood points of the robot links.

Description of Robot Structure Using the Unified Robot Description Format (URDF):

The URDF is an XML format for representing robot structure, and it is widely used in robot simulations [5]. Our method considers vertically articulated robots, as shown in Fig. 1(a), which are generally used in manufacturing plants.

A typical vertical articulated arm robot has a serial link mechanism consisting of links connected in series, as shown in Fig. 1(b). Links are connected mainly using revolute joints. Since a closed-loop link mechanism cannot be represented by URDF, we represent the link mechanism using URDFs of the “main chain” and “sub chains”. In this paper, the connected links that determine the robot's posture are referred to as the "main links”. The robot may also have a branched link mechanism for rigidity and loading, as shown in Fig. 1(c). Such links move in conjunction with the main links and are not directly specified for the robot operation. In this paper, we call those additional links as “sub links”.

The robot assembly model shown in Fig. 1(a) can be created by combining 3D models of links. The URDF maintains 3D models of links and their joint data. Joint data include the relationships between the parent and child links, and the rotation axis and rotation range of each link. In our method, the 3D model of each link is converted into a point cloud by randomly sampling points on the 3D model for fitting the 3D model to a point cloud. In this paper, point clouds generated from 3D models are referred to as CAD point clouds. On the other hand, points obtained using a TLS are called a TLS point cloud. Fig. 1(d) shows an assembly model consisting of CAD point clouds.

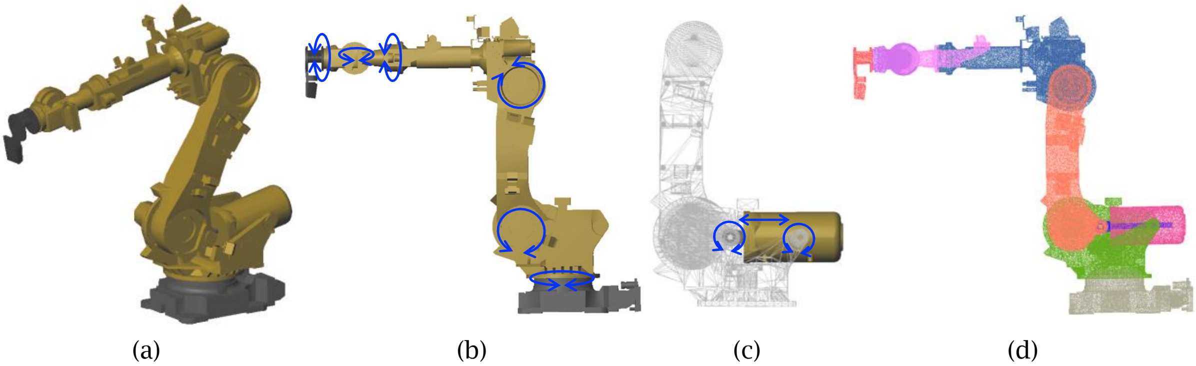

Fig. 1: Process of acquisition: (a)Robot model, (b) Main chain, (c) Sub chain, (d) CAD point clouds.

Detection of the Base Link:

Planar regions can be effectively detected from TLS point clouds by using the method proposed by Masuda et al. [4]. This method maps a TLS point cloud onto a 2D image using the azimuth and elevation angles of the laser beam, and planar regions are detected on the 2D image. Since most robots and machine tools are installed on the floor in manufacturing plants, objects on the floor can be separated by removing the floor points, which consist of large horizontal planes. The point cloud of each robot can be selected from segmented objects using common classification techniques such as deep learning. In this example, we selected robot points simply using the size of the bounding box of each separated object because there were no other objects of the same size as robots. Fig. 2(a) shows an example of robot points. In general, various methods, such as machine learning, can be used to identify the classes of separated objects, but we do not discuss classification methods further in this paper.

In our method, the base link is first detected from TLS point clouds. The base link is the root link of the main chain. To detect the base link, the installation plane of the robot is selected from planar regions, as shown in Fig. 2(b, c). Then, candidate points of the base link are searched upward from the reference plane. A k-nearest neighbor graph in Fig. 2(d) is created by using neighbor search, and Dijkstra's algorithm is applied at each point to calculate the shortest path from the installation plane. In Fig. 2(e), the color of each point indicates the distance of the shortest path. The shortest path distances are also calculated using the CAD point cloud of the base link in the same way. Then, TLS points with distances shorter than the maximum of the CAD point cloud are selected as candidate points of the base link, as shown in Fig. 2(f).

The CAD point cloud of the base link is fitted to the candidate points of the base link. For rough registration, an oriented bounding box (OBB) is created from each of the CAD point clouds and the TLS candidate points, as shown in Fig. 3(a, b). Then, the OBB of the CAD point cloud is moved and rotated at equal intervals so that the sum of the shortest distances between each point in the CAD point cloud to the TLS point cloud is minimized. As shown in Fig. 3(c), the calculated position is used as the rough registration result. Finally, the sparse ICP algorithm [2] is applied for precise registration, as shown in Fig. 3(d).

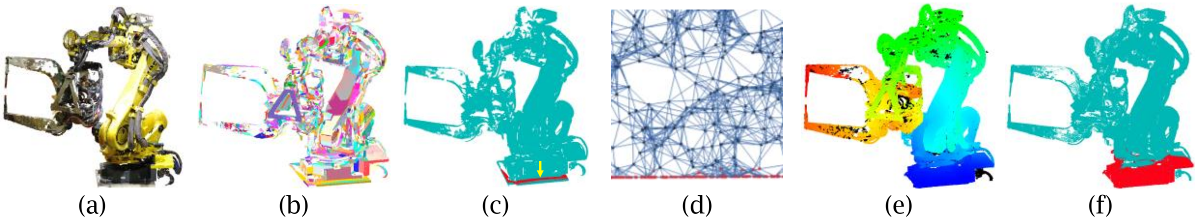

Fig. 2: Process of acquisition of the base link: (a)Point-Clouds, (b) Plane detection, (c) Selection of an installation plane, (d) K-nearest neighbor graph, (e) Distance from the installation plane, (f) Candidate points of base-link.

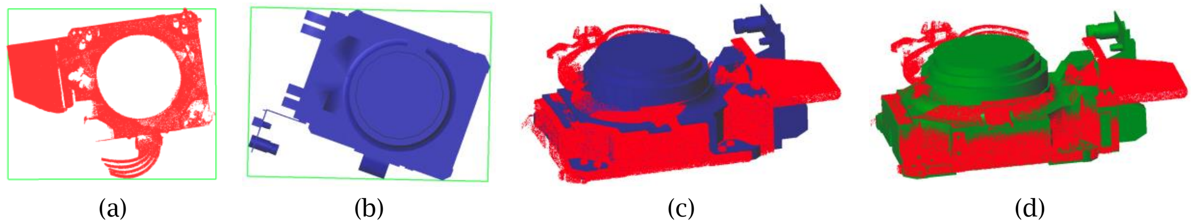

Fig. 3: Process of base-link registration: (a) OBB of base-link TLS points, (b) OBB of the base-link CAD points, (c) Result of rough registration, (d) Result of sparse ICP registration.

Calculation of Main Links:

In our method, the CAD point cloud of each link is fitted to the TLS point clouds sequentially, starting from the base link. Then, the rotation angle of each link is calculated by following child links from the base link. Initially, the rotation angle of each link is determined roughly; then the exact rotation angle is calculated by constrained registration using the relationships between links.

To roughly determine the rotation angle at each joint, the CAD point cloud is rotated at equal intervals around the axis defined in the URDF. The rotation angle is determined so that the sum of the shortest distances from each point in the CAD point cloud to the TLS point cloud is minimized. When the link shape is rotationally symmetric and the axis of rotation coincides with the axis of the link body, the links do not change when rotated. In such cases, the rotation angle is estimated using the child link.

Then, the rotation angles are calculated precisely using constrained registration. When fitting the CAD point cloud of a link to a TLS point cloud, the axis of rotation of the link must coincide with the axis of rotation of its parent link. To satisfy this constraint, we introduce two reference points on the axis. Let **a** and **b** be the reference points of the child link and $\mathbf{a_0}$ and $\mathbf{b_0}$ be the reference points of the parent link, as shown in Fig. 4. To align the axes of the two links, rotation matrix **R** and a translation vector **t** are calculated so that the following equation is minimized:

$$E = \sum_{i=1}^{N} |\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i|^2 + \lambda(|\mathbf{R}\mathbf{a} + \mathbf{t} - \mathbf{a_0}|^2 + |\mathbf{R}\mathbf{b} + \mathbf{t} - \mathbf{b_0}|^2) \tag{4.1}$$

where $\mathbf{q}_i$ is the corresponding point in the TLS point cloud that is nearest to each point $\mathbf{p}_i$ in the CAD point cloud, and λ is a constant. **R** and **t** that minimize Eq. (4.1) are calculated by iteratively updating the corresponding points.

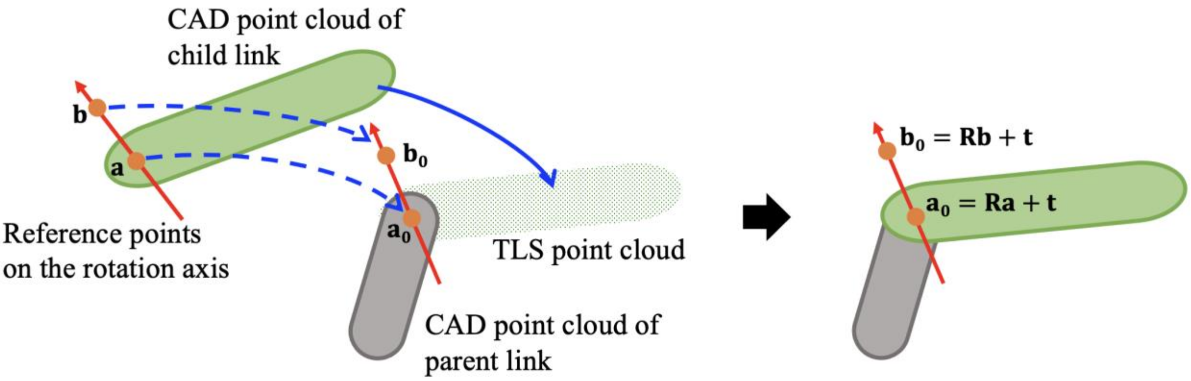
CAD point cloud of child link
b
a
b_0
a_0
Reference points on the rotation axis
TLS point cloud
CAD point cloud of parent link
$b_0 = Rb + t$
$a_0 = Ra + t$

Fig. 4: Registration constraints related to revolute joints.

Calculation of Sub Links:
Once the rotation angles of the main links are determined, the parameters of the sub links are calculated using the positions and postures of the main links. Fig. 5(a)-(c) shows a link mechanism commonly used in industrial robots. In these examples, since the positions of the links at both ends are determined by the fitted links, the positions and postures of the sub links can be computed as inverse kinematics solutions. In some cases, the solution is not uniquely determined. For example, the link in Fig. 5(d) has two solutions that satisfy the constraints of the main links. In such a case, the possible solution is selected by considering the rotation range of the link and the interference with other links.

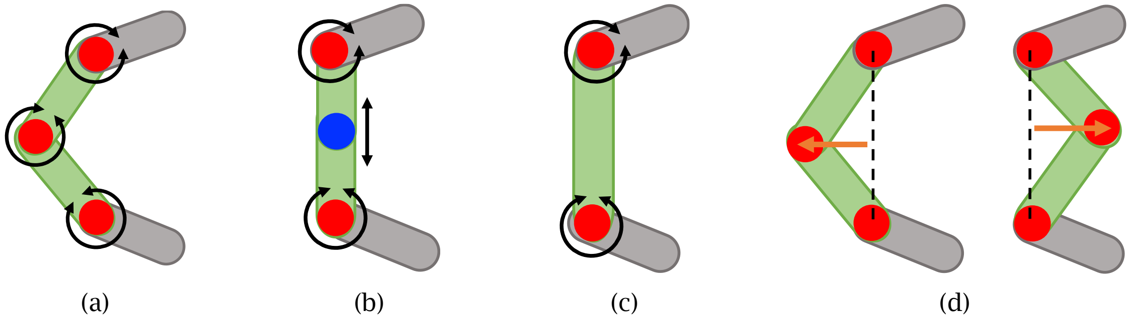

Fig. 5: Link mechanism for sub-chain: (a) Revolute-revolute-revolute linkage, (b) Revolute-prismatic-revolute linkage, (c) Revolute-revolute linkage, (d) Inverse kinematics solutions of revolute-revolute-revolute linkage.

Extraction of point cloud for each link:
Assembly robot arm models matching the TLS point clouds can be obtained by link fitting, considering the serial link mechanism. By comparing the TLS point cloud with the fitted CAD point cloud, the TLS point cloud of each link can be extracted. The remaining points that do not correspond to any link are considered robot guns, robot attachments, or wire harnesses. Since they move with the robot links, they can be regarded as part of the links in the robot motion simulation. Such enhanced 3D models can be used for accurate collision detection in robot motion simulation.

Experimental Results:
The proposed method was evaluated using point clouds of a Fanuc industrial robot as shown in Fig. 2(a). This robot is equipped with an end-effector for welding. The URDF consists of a main chain with seven rotary joints and two sub-chains with links shown in Fig. 5(b).

First, the 3D model of each link was fitted to the TLS point cloud. The results are shown in Fig. 6(a). By applying constraint registration, precise fitting results were obtained even for the links close to the end-effector and away from the base link. The auxiliary links could also be positioned in appropriate positions by computing the parameters of the sub chains.

Next, we extracted point clouds of each link using the fitted CAD models. Fig. 6(b) shows point clouds of links in different colors. Fig. 6(c) also shows assembled 3D models of the links. As shown in Fig. 6(b, c), the point clouds corresponding to the links were extracted by using fitted CAD models, and robot attachments and wire harnesses could be successfully separated from the point clouds.

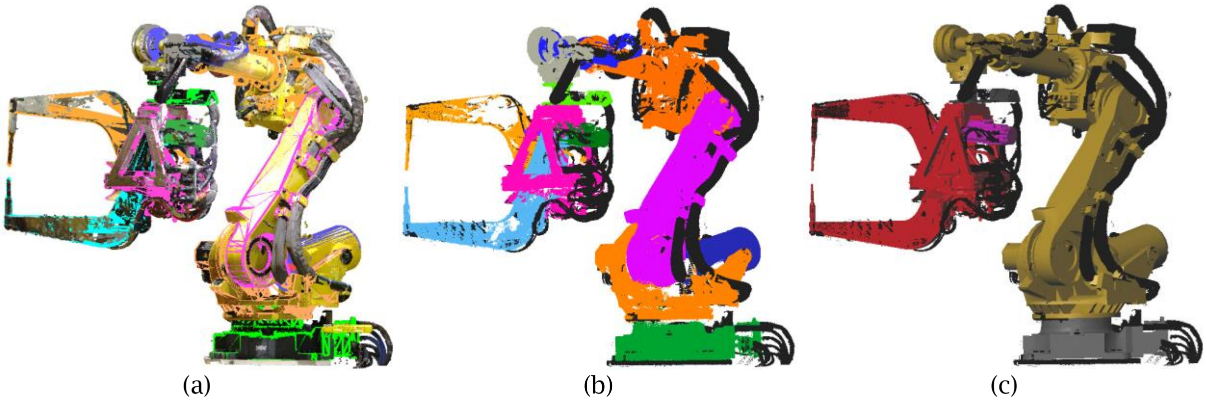

Fig. 6: Experimental Results: (a) Link fitting to TLS point clouds, (b) Extraction of a point cloud for each link, (c) Aligned CAD models.

Conclusion:

In this paper, we proposed a method for calculating the position and posture of links for articulated robots from the point clouds. The experimental results showed that our method could successfully extract point clouds of robot links with branching links.

In future work, we would like to develop methods for fitting detailed link mechanisms of various types of end-effectors as well as a wide variety of robots.

Kota Kawasaki, https://orcid.org/0000-0001-5188-158X
Kakeru Takeda, https://orcid.org/0009-0003-1295-9796
Hiroshi Masuda, https://orcid.org/0000-0001-9521-6418

References:

[1] Arun, K. S.; Huang, T. S.; Blostein, S. D.: Least-Squares Fitting of Two 3-D Point Sets, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987, PAMI-9(5), 698-700. https://doi.org/10.1109/TPAMI.1987.4767965

[2] Bouaziz S.; Tagliasacchi A.; Pauly M.: Sparse Iterative Closest Point, Computer Graphics Forum, 32(5), 2013, 113-123. https://doi.org/10.1111/cgf.12178

[3] Ip, C. Y.; Gupta, S. K.: Retrieving Matching CAD Models by Using Partial 3D Point Clouds, Computer-Aided Design and Applications, 4(5), 2007, 629-638. https://doi.org/10.1080/16864360.2007.10738497

[4] Masuda, H.; Niwa, T.; Tanaka, I.; Matsuoka, R.: Reconstruction of Polygonal Faces from Large-Scale Point-Clouds of Engineering Plants, Computer-Aided Design and Applications, 12(5), 2015, 555-563. https://doi.org/10.1080/16864360.2015.1014733

[5] Tola, D.; Corke, P.: Understanding URDF: A Survey Based on User Experience, International Conference on Automation Science and Engineering (CASE), IEEE, August 2023, 1-7. https://doi.org/10.1109/CASE56687.2023.10260660