Automatic Pose Generation for Robotic 3-D Scanning of Mechanical Parts

Inhwan Dennis Lee, Ji Hyun Seo, Young Min Kim, Jonghyun Choi, Soonhung Han, and Byounghyun Yoo

***Abstract*—In this article, we present an automatic pose generation approach for robotic 3-D scanning systems designed for the reverse engineering of mechanical parts. 3-D scanning of an object is usually a highly complicated process that involves aligning multiple measurements from different viewpoints, fusing the merged data, and finally reconstructing the underlying surface. The reconstruction is often performed in isolation to the acquisition step. In contrast, the measured data often exhibit holes owing to limited viewpoints or noises in measurements, which can be observed after the postprocessing. Previous approaches either use the predefined scan paths without knowledge of the actual surface, or calculate the scan path based on the approximated smooth surface enclosing the incomplete scanned measurement. The need for an accurate 3-D scanning of mechanical parts often arises for reverse engineering purposes, as their shape often exhibits sharp corners or continuous shape of primitives (planes, cylinders, etc.), which the existing pipelines do not consider. We suggest a 3-D scan pipeline specifically designed to generate optimal scan paths of mechanical parts from incomplete data with limited fields of view. Our pipeline detects primitives from the scanned partial data, and utilizes the detection to approximate the underlying shape of the ground truth object; thus, it suggests the next-best-view of the scanner from the scarce measurements. In this regard, the suggested method is the first working pipeline specifically designed for scanning mechanical parts; furthermore, it outperforms previous approaches in terms of an efficient scan path generation.**

***Index Terms*—Automatic 3-D scanning, next best view, pose generation, primitive detection.**

Manuscript received August 17, 2019; accepted February 27, 2020. Date of publication May 7, 2020; date of current version August 5, 2020. This work was supported in part by the Korea Institute of Science and Technology (KIST) under the Institutional Program Grant 2V07140 and 2E30270 and in part by the National Research Council of Science and Technology (NST) by the Korean government (MSIT) under Grant CMP-16-01-KIST. This paper was recommended for publication by Associate Editor J. O'Kane and Editor F. Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Byounghyun Yoo.)*

Inhwan Dennis Lee and Soonhung Han are with the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: inhwanlee@kaist.ac.kr; shhan@kaist.ac.kr).

Ji Hyun Seo and Byounghyun Yoo are with the Korea Institute of Science and Technology, Seoul 02792, South Korea (e-mail: jihyun.seo@wrl.kist.re.kr; yoo@byoo.net).

Young Min Kim is with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: youngmin.kim@snu.ac.kr).

Jonghyun Choi is with the AONEware, Seoul 07064, South Korea (e-mail: aoneware@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2020.2980161

I. Introduction

With the recent advance of 3-D scanning technology, there are multiple solutions available for acquiring a high-quality scan of ordinary objects. Even with the best scanner available, the object should be scanned in multiple viewpoints and then merged into the unified coordinate frame for further processing [1]. The alignment usually requires online tracking or manual assignment of correspondences followed by a variant of the iterative closest point algorithm [2] to refine the alignment. Such processes, especially for high-resolution scans, cannot be conducted in real-time, and the insufficiency in the scanned data can often only be observed after intensive manual labor. The typical practice is to iterate between the scanning and the quality evaluation after postprocessing until the result is satisfactory. The persistent iteration is the major bottleneck that prohibits ordinary users from utilizing the 3-D scanning technology for purposes such as 3-D printing. The optimal scan position depends on not only the sensor specification (optimal measurement range, viewing direction, etc.) but also the shape, which can usually only be known to the system after the scanning process is completed. In addition, there are regions that cannot be measured by scanners, such as cavities with narrow openings or specular materials.

Prior works have attempted to circumvent the problem and allow automatic scanning of an object by attaching a robot to the scanner, as shown in Fig. 1. The primary challenge is that we do not know the shape of the object beforehand, and the optimal scan pose or completeness can be defined when the shape is known. The commercially available solutions usually use the predefined path around the object, anticipating that the densely sampled viewing positions would cover the desired area to be scanned. Some past approaches [3], [4] created a rough shape of the object by first scanning with the predefined path and using the estimated shape to calculate the area, which requires more measurements. For general shapes, the use of the boundary of measurements [5], [6], or smoothness and sampling density of the surface [7] can have reasonable performance.

The aforementioned prior works have focused on visual completeness and showed the performance on figures of a character or miniatures of a building. However, in this article, we focus on the engineering perspective and on acquiring manufacturing features that can be considered for the assembly of physical parts. An example of a common use-case of practical application is scanning a damaged real-world part for 3-D printing for maintenance of the product, where the product is discontinued

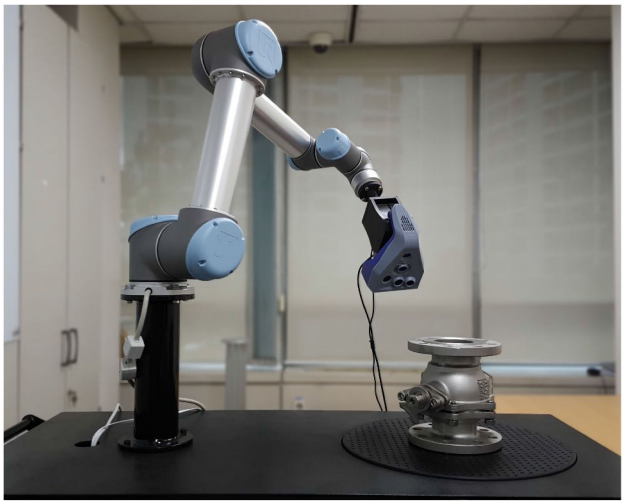


Fig. 1. Real-world setup for robotic 3-D scanning. By moving the robot arm and rotating the turntable based on the given pose from the next-best-view (NBV), the 3-D scanner views the intended surface of the unknown model.

or hard to acquire. However, the first burden for 3-D printing is acquiring a high-quality model with the robotic 3-D scanner.

We present an automatic pose generation approach for robotic 3-D scanning specifically designed for scanning mechanical parts that can be approximated as a combination of primitive shapes. The shape family often deserves highly accurate reverse engineering for fault detection, inspection, or manufacturing applications. The existing autoscanning approaches that assume smooth surfaces are not suitable for such models with high curvature regions (often represented as B-rep in computer-aided design (CAD) modeling) and holes. The common practice is the measurement of the desired specification or dimensions of interest accurately with specialized devices or probes by trained human engineers, which cannot scale to a significant number of models.

In contrast, our approach enables the automatic scan pose generation of the robotic 3-D scanner for part models with high-curvature regions and concavity. We can infer unknown shapes from scarce measurements based on the regularity within the acquired data and primitive fitting, and use the information to design the next-best-view (NBV) of the scanner. Existing primitive detection algorithms designed for reverse engineering assume data that cover the entire shape even with uniformly sampled points, and the detection can be erroneous with partial data. We designed an algorithm to detect primitives reliably with nonuniform partial data and approximate unknown shapes. Our algorithm is implemented on a physical platform with a 3-D scanner attached to the robot arm and a turntable. We demonstrate with both virtual and real data that our approach is superior to existing approaches for automatically scanning CAD models.

The major contributions of our suggested pipeline can be summarized as follows.

1) We suggest the first automatic 3-D scan path generation approach designed for mechanical parts using primitive detection.
2) Our primitive detection algorithm robustly processes incomplete scan data, which includes partial and nonuniform samples. It can further estimate a region with acceptable error, where the previous reverse engineering methods have limits and restrictions.
3) We demonstrate a user-friendly physical solution for automatic 3-D scanning by calibrating multiple hardware and providing a Web-based interface that modulates communication between software modules.

II. Related Work

A. Shape Priors and Primitives for Reverse Engineering

Our approach is based on the observation that CAD models can be represented as combinations of primitive shapes. The idea of using parametric representations for reverse engineering has a rather long history [8]. Based on the knowledge of CAD design and representation steps, the earlier works started with boundary representations [9], [10], or parametric patches [11]–[13]. Simple primitives such as spheres, cylinders, or planes can also be incorporated into the process [14]. Recently, Du *et al.* [15] progressed one step further and suggested a way to reverse-engineer constructive solid geometry (CSG) trees from detected primitives. However, even with the uniformly sampled data covering the entire surface of the object, the reverse engineering process can be erroneous at subtle boundaries between different primitives or as the complexity increases.

More recently, geometric priors other than simple primitives have been utilized for effective reverse engineering. For example, a residential area can be represented with billboard trees, buildings, and ground planes with a predefined relationship [16] or an indoor environment can be covered with an assembly of planar patches [17]. The semantic priors considerably reduce possible ambiguities between a few candidate combinations by considering the relationships between detected elements. Such a relationship can be deduced from physics in mechanical assemblies of parts [18] or learned from the shape database considering the geometry and context [19].

The reasoning of a relationship can also help detect the primitives [20], [21]. In this regard, primitive detection is practiced in two stages. After the initial primitives are approximated, the global relationship can be deduced to refine and regularize the detection. The inherent ambiguity of parametric representation is considerably reduced with the global regularity criteria. Le and Duan [22] demonstrated that finding the dominant direction can suffice to regularize the primitive parameters reliably.

In our setting, we utilize primitive detection for partial, nonuniform scan data to infer the unknown shapes and determine the NBV for the measurement. Some works have attempted to fill the missing portion with primitive assistance [23], but they are limited to the cases with relatively small holes with predefined continuous boundaries. In contrast, we adapt the existing primitive detection to be robust to highly challenging data as more data become available by extracting the dominant direction and refining the parameters.

B. Scan Path Generation

The 3-D measurements represent the distance to the surface of the object from the viewpoint of the 3-D sensor. The quality of measurements depends not only on the accuracy of the sensor but also largely on the viewpoint and the resulting occlusions. Even with the best available scanner and even after multiple measurements from carefully designed viewpoints, there are missing parts and holes on the object where the measurement could not reach. We approach the problem of designing the best viewpoint of the 3-D sensors to obtain the quality of reconstruction or reverse engineering that is as high as possible when the measurements are postprocessed.

The most common practice is manual scanning or using a predefined path [3]. The quality of reconstruction is inspected manually after each stage of postprocessing and the desired region is captured again. Approaches to assist the user during the scanning stage have also been proposed [24].

In terms of automated scanning, the scan path is automatically generated given rough or exact geometry of the target object. The region has been reconstructed using automated aerial robots with a camera in an explore-then-exploit manner, which captures the rough geometry using a predefined path of the camera, and then generates a path for detailed coverage of the region [25], [26]. Similarly, Karaszewski *et al.* [4] suggested a pipeline composed of two stages, where the rough estimate of the shape is first acquired, and the details are captured during the second stage based on the initial estimation. The existing CAD model is also used to generate an optimized path of the sensor (2-D and 3-D) for the full coverage of the object [27], [28]. In the aforementioned works, scanning views are only planned for the coverage of known geometry, where every point on the surface has an equal level of importance without any priority.

The major challenge is that we often do not know the underlying shape before scanning, and we need to make a decision based on partial information. In such a case, it is important to analyze the currently available sensor data and plan the NBV with priority measured for the maximum amount of information. Early works assume a setup with an unknown object on a turntable and exhaustively calculate occlusion and sampling density within the predefined viewing volume [29], [30]. However, the path has only one degree of freedom, and the amount of computation cannot scale to a more general setting. For aerial 3-D reconstruction, Peng and Isler [31] combined the NBV approach with the explore-then-exploit method, which scans rough geometry of the region and plans the next scanning views not only focused on simple coverage but also with priority given to each view and its corresponding target surface through the analysis of existing data. For indoor scene scanning, Xu *et al.* [32] suggested a scene acquisition technique for object segmentation based on robot navigation and pushing candidate objects to improve the segmentation. Their scan path depends on the original rough reconstruction of the whole scene and was further refined for individual objects using the Poisson-field-based metric [7]. For single objects, the next view is chosen based on the surface boundaries of the previous scans [5], [6], or the quality of Poisson reconstruction [7]. A boundary-based method induces views to fill holes but does not consider geometric detail in the scanned region. Conversely, the quality of Poisson reconstruction indicates the level of geometric detail in the previously scanned region and also the evidence of data at the isosurface of Poisson field.

None of the previous approaches for scan automation was designed to utilize geometric priors of mechanical parts or CAD models; they assume a continuous [5], [6] smooth surface [4], [7]. Hu *et al.* [33] suggested a method to incorporate semantic labeling instead of primitive labels, to assist registration and reconstruction during the scanning stage. However, in contrast to our work, this work is focused on registration and reconstruction, and the problem of exhaustive scan path is still in the control of the user without any systematic suggestions. Our approach makes the decision of completeness based on not only density and continuity, but also primitive detection and inference of shapes. As an additional advantage of the approach, our suggested pipeline is robust against deep holes, sharp feature lines, and continuous surfaces. Although it is not our major goal, we can also extract interesting features or segmentation for reverse engineering, such as major planes, perpendicular holes, and radius of cylinders, which can be used to search the database or process a large family of 3-D CAD models effectively.

III. Overview

The overview of the system is depicted in Fig. 2. We assume that there is an object to be scanned by an ordinary 3-D scanner whose measurement is obtained by defining the viewing position and direction as a 4×4 transformation matrix $\mathcal{T}^t$. The acquired data are in the form of depth frame D^t, which is a 2-D grid of dense 3-D points. As presented in the previous approaches of automatic scanning [4]–[7], the object is scanned through an iterative process: the system processes the accumulated frames up to the current time step t, $\cup_{t=1,\ldots,t_0} \mathcal{T}^t(D^t)$. It outputs the NBV $\mathcal{T}^{t_0+1}$ to be scanned based on the analysis of the incomplete measurements. We suggest an optimized NBV assuming that the object to be scanned is a typical mechanical part, which can be represented by a combination of primitives. The accumulated measurement is analyzed by detecting prominent primitives that comprise the object. The process is iterated until the system decides that no more improvement is possible by taking more measurements, and all the necessary features are obtained during the scanning session.

We start from an initial incomplete measurement of the object obtained using the scanner by capturing from predefined views. There are many possible choices, and in our system, we suggest either (1) 24 equally located views around 360° of the objects or (2) 5 views looking at the right, left, front, back, and top sides of the bounding box of an object. We allow the user to choose between the two options. Given the incomplete scan, we iteratively optimize the NBV based on the approximation of the underlying surface. The partial, nonuniform measurements are first preprocessed using the conventional Poisson surface reconstruction [34] to create a smooth and uniform sampling. From the samples, we can detect primitives robustly near the measurements [22]. We subsequently infer unknown regions by

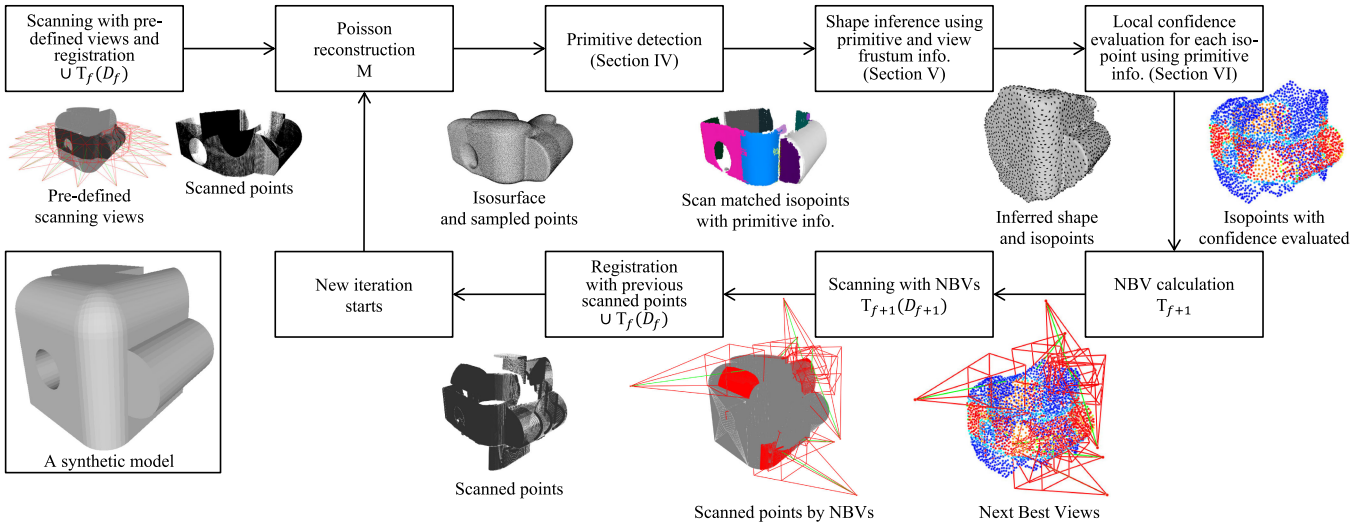


Fig. 2. Overview of the automatic 3-D scanning pipeline. For each iteration, a frame of measurement is transformed ($\mathcal{T}^f(D^f)$) and merged into a unified coordinate system. The merged measurement is further processed with detected primitives in addition to the ordinary Poisson reconstruction (Section IV). Based on the detected primitives and view frustum information, we infer a new shape, which is modified from the initial shape approximation $\mathcal{M}^f$ (Section V). Subsequently, local confidence for each isopoint on a new inferred shape is evaluated using primitive information and NBVs are calculated based on the local confidence map (Section VI) until the system decides that the scan is complete.

using detected primitives, the history of viewing frustum of the scanner, and recreate the Poisson surface approximation. For each point on the approximated surface, we assess confidence by determining its conformity to the primitive, in addition to point density, and representation of actual geometric detail [7]. Subsequently, we calculate the NBV to observe the overlap of confident and less-confident surfaces together to register the additional information anchored by the confident region reliably.

The rest of this article is organized as follows. Sections IV–VI, respectively, describe each of the major phases of the system: primitive detection, shape inference, and NBV calculation. For the actual use of the pipeline, various aspects of the real-world implementation are presented in Section VII. Then, in Section VIII, evaluations of the performance of the proposed algorithm in comparison to those of state-of-the-art algorithms are given. Section IX concludes this article.

IV. Primitive Detection for Partial Scans

Given the accumulated scans, $\cup_{t=1,\ldots,t_0}\mathcal{T}^t(D^t) = \mathbf{S} = \{s_i\} \subset \mathbb{R}^3$, of an object, the system detects primitives and returns the parameters of the primitives $\mathbf{P} = \{p_i\}$ with the segmentation of the points based on the affiliation to the primitives, $\phi : \mathbf{S} \to \mathbf{P}$. Previous primitive detection algorithms assume uniform, complete samples of the underlying model, and even in these cases, the detection can be erroneous. Our algorithm modifies the existing algorithm (Section IV-B) with additional preprocessing (Section IV-A) and postprocessing (Section IV-C) of the data. The preprocessing converts the measurements $\mathbf{S}$ into uniformly sampled points of an approximated surface $\mathbf{Q}$, and the postprocessing adjusts the parameters by considering whether the samples are from the actual measurement or the inferred shapes. Fig. 3 depicts the overall process of our primitive detection algorithm, where we highlight our robust adaptation with shaded boxes with bold fonts.

A. Preprocessing: Poisson Surface Reconstruction and Classification

The preprocessing converts the partial and irregular scan measurements $\mathbf{S}$ into uniform samples $\mathbf{Q}$ to apply primitive detection algorithms reliably. The data are converted in four stages.

In the first stage, based on the accumulated scanned points $\mathbf{S}$, an approximate surface mesh $\mathcal{M}$ is created using Poisson surface reconstruction. The byproduct of the Poisson surface reconstruction is Poisson scalar field Γ, which we also use for quality assessment of the reformed shape in Section VI. Fig. 3(a) shows an example of scanned points (left) overlaid with the reconstructed mesh (right) filling the unknown area under smoothness assumption. It is the first estimated form of the unknown model and will be reformed by applying the properties of the primitive features after the detection.

Second, the reconstructed mesh $\mathcal{M}$ is classified into *scanned* $\mathcal{M}_s$ and *nonscanned* $\mathcal{M}_n$ regions by comparing the distance between the scanned points $\mathbf{S}$ and each face, as shown in Fig. 3(b). The scanned surface $\mathcal{M}_s$ is the approximation from the actual measurements, whereas the nonscanned surface $\mathcal{M}_n$ is an artificial generation filled under smoothness assumption from the partial data.

The third stage uniformly samples the points $\mathbf{Q} = \{(q_j, \mathbf{n}_j)\} \in \mathbb{R}^6$ on the isosurface $\mathcal{M}$ of the Poisson scalar field. Following the classification result, the sampled points are also categorized as either *scanned*, $\mathbf{Q}_s$, or *nonscanned*, $\mathbf{Q}_n$. The primitive detection is performed only on the actual scanned region $\mathbf{Q}_s$, followed by the extension of the detection result to *nonscanned* $\mathbf{Q}_n$ for the primitive region growing in Section V.

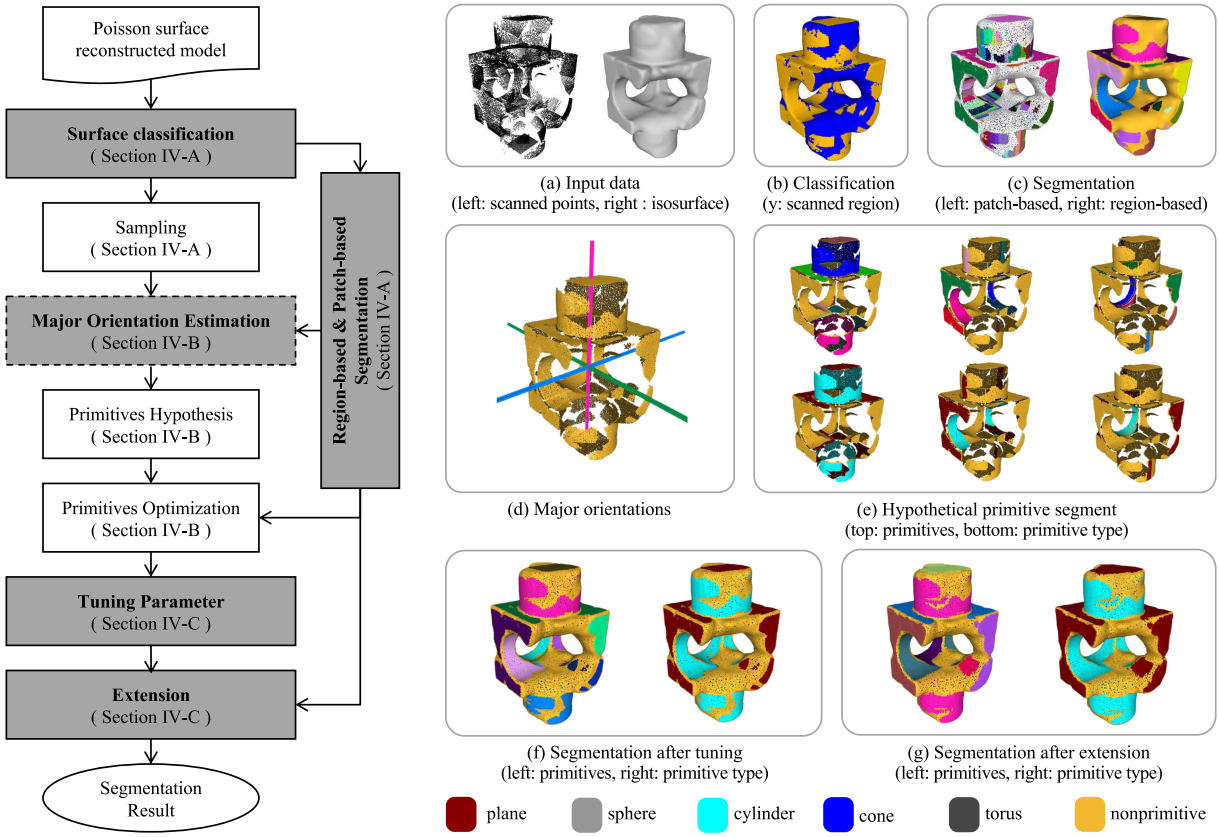


Fig. 3. Pipeline of primitive detection algorithm with intermediate results of a sample model. (a) Input data, scanned points (left), and isosurface (right) from Poisson surface reconstruction. (b) Surface classification separating a triangle surface into scanned and nonscanned regions [Color: yellow (*scanned*), blue (*nonscanned*)]. (c) Patch-based (left) and region-based (right) segmentation on isosurface. (d) Estimated major orientation. (e) Proposed hypothesis of primitives for each estimated orientation. The three columns show the proposed primitives based on different orientations in two ways: random colors per primitives (top) and color-coded per primitive type (bottom). (f) Optimized primitives from the proposed hypothesis (e). (g) Extension of the primitive-labeled region on the overall surface.

Finally, we partition $\mathbf{Q}$ into groups of small planar patches $\tilde{\mathbf{Q}}$ to create oversegmentation of the samples $\mathbf{Q}$, and also maintain the *scanned* or *nonscanned* labels per patch. The pieces of small continuous planar patches are useful to reduce the computation costs and create a useful approximation of the surface. It is used to optimize the fitting in Section IV-B, and also for postprocessing in Section IV-C. In addition, we further create a larger region segment $\tilde{\mathbf{P}}$ as the candidate entity for primitive estimation, by grouping neighboring planar patches with similar normals. The region segmentation separates the mesh by sharp area, mainly at the edges, which generally aligns with primitives or groups of primitives. The introduction of the region segments further stabilizes and accelerates the primitive detection step during the initial unknown stages.

After preprocessing, the samples and patches in the scanned group ($\mathbf{Q}_s$, $\tilde{\mathbf{Q}}_s$) will be used for the conventional primitive fitting and nonscanned region ($\mathbf{Q}_n$, $\tilde{\mathbf{Q}}_n$) to expand the detected primitives.

B. Primitive Detection for Reverse Engineering

In this subsection, we briefly describe the core functionality of recognizing and extracting primitive $\mathbf{P}$ from 3-D data based on estimated major orientations [22] as the detection is stabilized by utilizing the major directions and enforcing necessary regularization accordingly. The readers interested in further details should refer to the original publication [22].

The input to the algorithm is the uniform samples of scanned points $\mathbf{Q}_s$, and the output is the detected primitives with the corresponding primitive parameters $\mathbf{P}$ and the labels of sampled points $\phi : \mathbf{Q} \rightarrow \mathbf{P}$. First, the major orientation is voted by binning the probable plane normals and cylindrical axes. The normals with the most number of votes are selected as the candidates of major orientation. An example of a major orientation estimated from the sample mesh is shown in Fig. 3(d). For each candidate orientation, points are divided into planar and nonplanar regions, depending on whether the normal of the point is closely orthogonal to the orientation. In the planar region, the plane is searched where a certain number of points with similar normal values are gathered. In the nonplanar region, the detected circle slices, as in the earlier step, are used to form the profile curve of the axis along the major orientation, which can be simplified to parts of the cylinder, cone, sphere, or torus. In Fig. 3(e), three columns are the primitives detected from three different major orientations showing a random color for primitives (top) and a fixed color per type of primitives

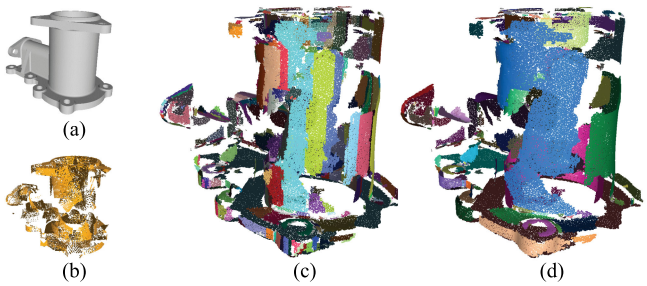


Fig. 4. Region-based segmentation on the scanned region of the model. (a) Tested CAD model. (b) Partial scanned points. (c) Patch-based oversegmentation. (d) Region-based segmentation. For each patch in (c) or region in (d), random colors are used.

(bottom). After the primitives are estimated from the selected major orientation, the parameters are refined and the resulting labels are locally adjusted.

For a more robust major orientation estimation, we utilized the region segments $\tilde{\mathbf{P}}$ as suggested in Section IV-A. The estimation of major orientation depends on the distribution of the point cloud's normal, which also indicates that the smaller the size of the primitive to the total area, the smaller is the possibility that it can be determined. To avoid the limits in size of the primitive, we estimate the orientation from $\tilde{\mathbf{P}}$, which generally segments primitives or groups of primitives that share the same axis as shown in Fig. 4(d). The segmentation in the previous method, i.e. patch-based oversegmentation shown in Fig. 4(c), despite being applicable for the other steps of the algorithm, creates a small segment of the plane that will only show the plane normal of itself. However, the region-based segmentation bounds the $\mathbf{Q}$ in rough primitive unit and each region segment distribution of normals shows a clear pole or ring, which is not disturbed by the normal of other primitives. Estimating only the required orientation will not only lead to sophisticated detection but also reduce the unnecessary search process overall. Fig. 5 shows the estimation of major orientation from each region segment as per our suggested method and from the overall region as per the previous method. Compared with the previous method, it can be observed that our approach has avoided estimating incorrect candidates as major orientations and has extracted necessary information.

For each scan, the primitive detection process is repeated as the new approximated surface is produced by accumulating additional scan data. However, it is not necessary to start the entire process from scratch. The new scan data $\mathcal{T}^{t+1}(\mathcal{D}^{t+1})$ is usually incremental; it is generally used to refine the accuracy, and the appearance of a new primitive is rare. To reduce the computational burden, we examine reusable primitives from the previous detection results. Specifically, we compare the Poisson surface reconstruction of the previous iteration $\mathcal{M}^t$ with the new surface reconstruction $\mathcal{M}^{t+1}$ after the surface classification described above. For each *scanned*-region face in the previous mesh, we determine the closest face of the new mesh and verify the difference in terms of both distance and normal. The primitive label is copied if the movement in the distance or normal is below the selected threshold. After comparing all the faces, the primitives with a sufficient number of sampled points are selected as reusable ones. Subsequently, the region is directly processed via the postprocessing of the extension for the primitive region mentioned in Section IV-C. By extending the existing primitives before searching for new primitives, we have reduced the number of samples to test for finding new primitives, and consequently save time by avoiding unnecessary repetitions.

C. Postprocessing: Tuning Parameters and Extending Primitive Labels

After obtaining the primitives with the initial parameters $\mathbf{P}$, the primitive parameters are refined with $\mathbf{Q}_s$ and extended toward the nonscanned region, $\mathbf{Q}_n$. The refinement and extension are tested in terms of the distance $d(q_j, p_i) < \sigma$ and the normal $(n_j, n(p_i)) < \theta$.

For refining the primitive parameters, we use the segmented points from the scanned region $\{(q_j, n_j) \in \mathbf{Q}_s : \phi(q_j, n_j) = p_i\}$ with a strict threshold of distance and normal σ_s, θ_s. The primitives are rechecked with the refined parameter and those without a sufficient number of inliers (60% of the original segment) are filtered.

After the refinement, the result is extended into its neighborhood, including the nonscanned region in the planar surface patch unit $\tilde{\mathbf{Q}}_u$ extracted from the preprocessing. The segmentation result in Fig. 3(g) shows that the extension stage has covered the nonprimitive region including the nonscanned region. Starting from the immediate neighbors of the segments of detected primitives, the primitive labels are extended with a relatively generous thresholds σ_u and θ_u ($\sigma_u > \sigma_s, \theta_u > \theta_s$). This can include the nonprimitive $\mathbf{Q}_s$, which is distorted owing to insufficient scan data. Fig. 6 shows an example of the primitive region extension using planar patches. For a distant patch to be included in a primitive, patches that satisfy the condition, including itself, must link this patch to the primitive. Finally, as the expansion step is finished, the primitives $\mathbf{P}$ are passed to the next process, i.e., the shape inference.

V. Shape Inference Using Primitives

From the *scanned* $\mathbf{Q}_s$ with the detected primitives $\mathbf{P}$, we approximate the unknown parts of the target object to optimize the next scanning view $\mathcal{T}^{t+1}$. After performing the primitive detection described in Section IV, we have the current approximate surface reconstruction $\mathcal{M}$ derived from $\mathbf{S}$ and primitive labels for each point samples. At a high level, we generate a better approximate surface $\hat{\mathcal{M}}$ extending the detected primitive shapes of the scanned region near the boundary of $\mathbf{Q}_s$ occurred owing to the limited size of the viewing frustum. We define the boundary $\mathbf{B} \subset \mathbf{Q}_s$ of the scanned region and extend the shape of detected primitives only if the extension lies within the unknown region considering the history of the past viewing frustums. The amount of the shape inference is adjusted considering the size of the viewing frustum. The overall process of shape inference is explained in more detail in the following subsections and also depicted in Fig. 7.

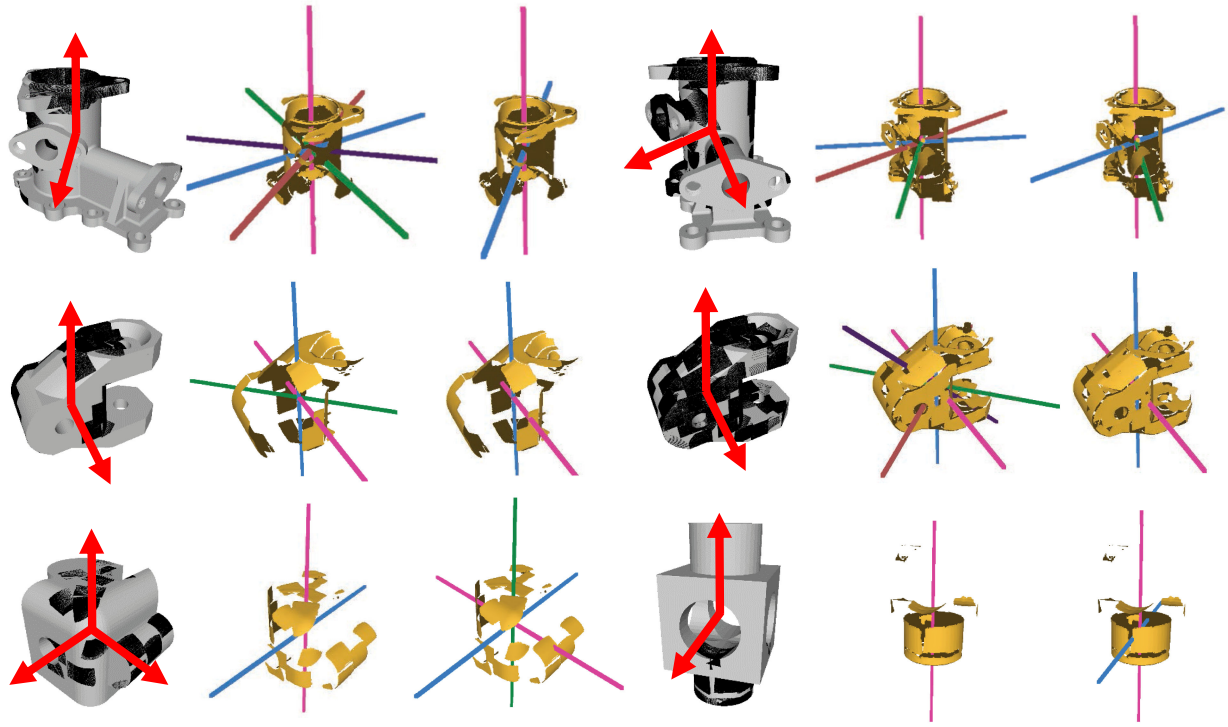


Fig. 5. Comparison of estimating major orientation using the previous method and our suggested approach. Samples of partial scan data in different models or at different iteration are tested. Each case presents predictable orientations from the scanned points of the target model (left), and the result of estimated orientations from the total region (middle) and each segmented region (right).

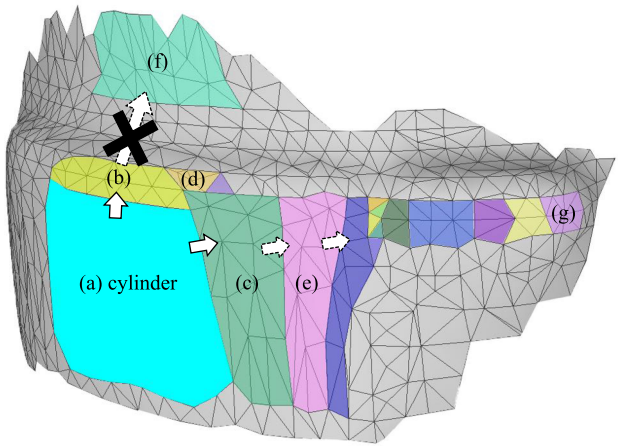


Fig. 6. Example of primitive extension on a surface mesh M. Primitive-labeled regions sequentially expand the area according to the face connection in the patch segment unit. Random colors are given on patches (b)–(f) that have there distance and normal close to the primitive (a) cylinder. However, if a patch with a permissible distance and normal error has a breakpoint in the connection by the steep surface in the primitive area, such as (f), it does not undergo extension in that direction.

A. Boundary Extraction and Outward Direction Calculation of the Scanned $\mathbf{Q}_s$

As the first step of the boundary extension, we extract boundary points $\mathbf{B}$ and define the vector v by defining the outward direction from each point b to be used for shape inference. The boundaries $\mathbf{B}$ of $\mathbf{Q}_s$ are extracted considering the sampling density of $\mathbf{Q}$ and the relative locations of $\mathbf{Q}_s$ and $\mathbf{Q}_n$. Specifically, we mark a subset $\mathbf{B} \subset \mathbf{Q}_s$ as boundaries if $\forall b \in \mathbf{B}$, $\exists q_n \in \mathbf{Q}_n$ such that $d(b, q_n) \leq r_b$, where r is a prefixed threshold considering the sampling density of $\mathbf{Q}$.

Subsequently, we calculate the outward direction v from each boundary point b by using neighboring q_s such that $d(b, q_s) \leq r_v$ and $q_s \notin \mathbf{B}$, where r_v is a prefixed threshold distance, which is 0.1 in our experiment. Given the sample points q_s, the outward direction is defined as the average of normalized vectors from q_s to the corresponding boundary point b. The outward vector from the scanned region to each boundary point is obtained as depicted in Fig. 7(k).

B. Boundary Classification Using the Viewing Frustum Information

The detected boundary points can be further classified into two categories: either from the sharp edges of the object itself or the measurements at the limits of the viewing frustum of the scanner. Our algorithm distinguishes the two categories and effectively extends the shape only from the boundary of the latter category. Consequently, it prevents an extension of primitive discontinued by object boundary, which happens inside the view frustum (see Fig. 8).

Specifically, the boundary points are classified into the object boundary $\mathbf{B}_o$ or the frustum boundary $\mathbf{B}_f$ depending on the distance to the boundary of the viewing frustum. During the scanning process, the history of scan positions and orientations is saved and used to decide whether the boundary point is inside the frustum. If the boundary of the scanned region happens to

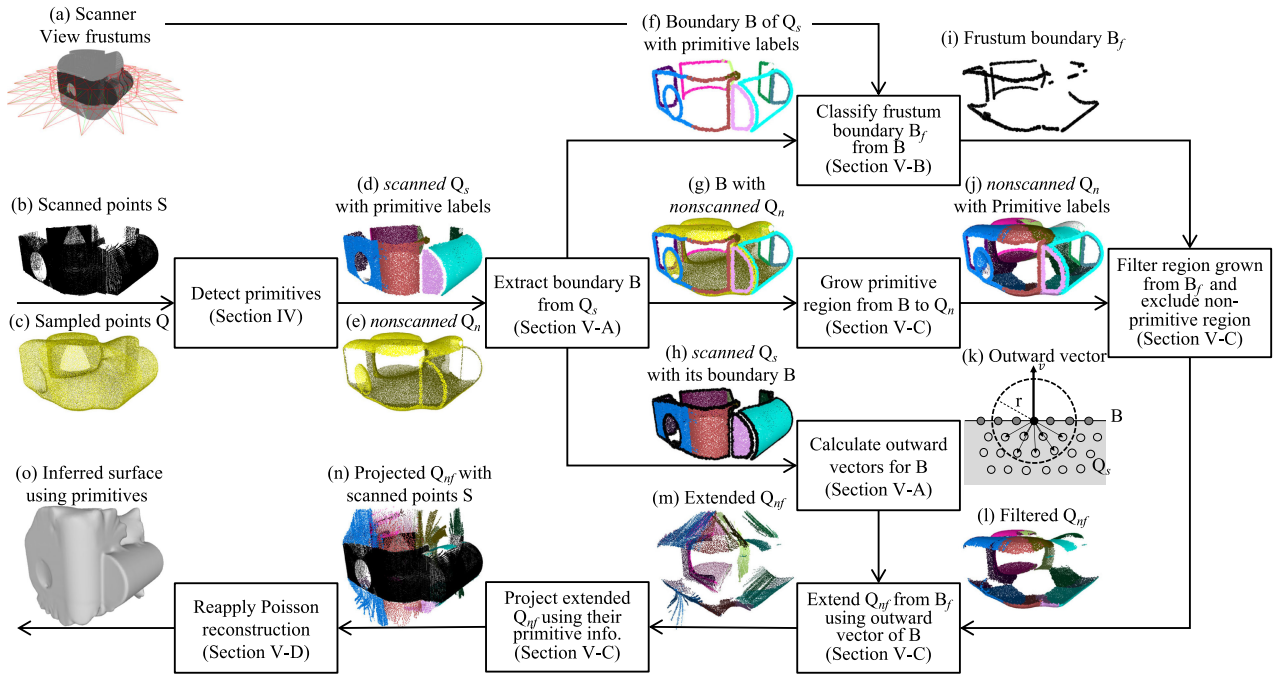


Fig. 7. Pipeline of shape inference based on detected primitive information. After the primitives are detected in Section IV, the frustum boundary $\mathbf{B}$ of $\mathbf{Q}_s$, which occurs by the limit of the scanner's viewing frustum, is determined. The primitives discontinued at $\mathbf{B}$ are extended by the modification and projection of $\mathbf{Q}_n$. A new surface approximation $\hat{\mathcal{M}}$ is acquired by reapplying Poisson reconstruction to the modified samples $\tilde{\mathbf{Q}}_{nf}$ by extending the detected primitives.

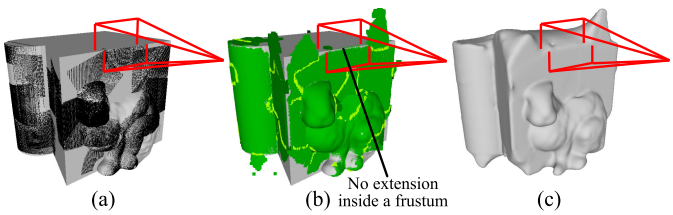


Fig. 8. Extension of primitive shape considering view frustum. From the scanned points in (a), we should extract frustum boundary points from which the region is extended. In (b), yellow points indicate the frustum boundary of the scanned region. It shows the result of extension in green together with the synthetic scanning object. Owing to the view frustum shown in red, the boundary inside the frustum is not used for the extension. The shape is inferred as (c).

be inside the view frustum (red in Fig. 8), it is excluded from the frustum boundary [yellow in Fig. 8(b)] and further expansion is prohibited.

C. Primitive-Based Scanned Region Extension Using $\mathbf{Q}_n$

Now, we have the boundary region $\mathbf{B}$ and the primitive detected from $\mathbf{Q}_s$, and we extend the detected primitives by the outward direction v. Instead of the initial shape approximation $\mathcal{M}$ obtained from the scanned points $\mathbf{S}$, we devise the nonscanned samples $\mathbf{Q}_n$ starting from the boundary $\mathbf{B}$ using the primitive information, and infer a modified shape $\hat{\mathcal{M}}$. Given the primitives detected from $\mathbf{Q}_s$, we project $\mathbf{Q}_n$ in Fig. 7(e) as depicted in Fig. 7(n) using the parameters of the detected primitives. A 2-D illustrative example of the extension process is shown in Fig. 9.

The amount of primitive projection is controlled by the number of samples $\mathbf{Q}_n$ nearby and the size of the viewing

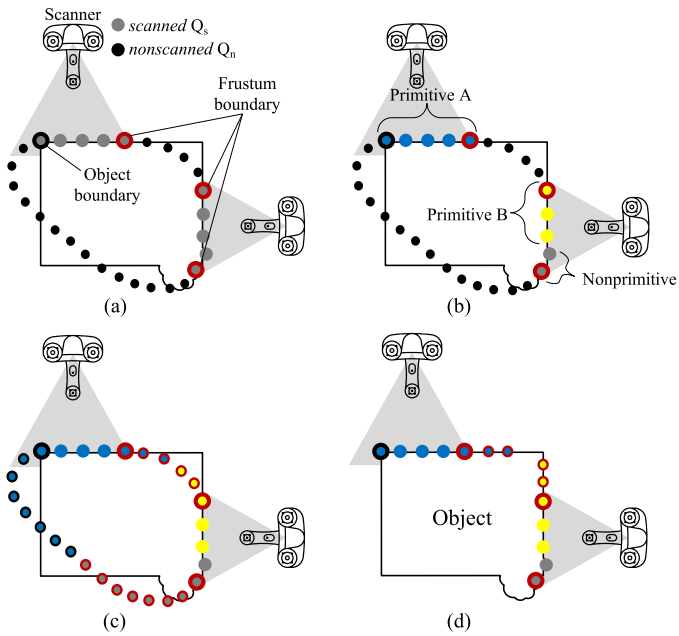


Fig. 9. 2-D illustrative example of the *scanned* $\mathbf{Q}_s$ extension using the projection of *nonscanned* $\mathbf{Q}_n$. (a) Extract boundary points of $\mathbf{Q}_s$ and classify them into frustum boundary and object boundary as described in Section V-B. (b) Detected primitives from $\mathbf{Q}_s$. (c) Primitive labels and frustum boundary or object boundary classification labels are attached to $\mathbf{Q}_n$ after the primitive region grows. (d) Extend primitives from frustum boundary points with primitive label A (blue filled points) or B (yellow filled points) excluding nonprimitives and the remaining points in $\mathbf{Q}_n$ are deleted.

frustum of the scanner. The points in $\mathbf{Q}_n$ are sampled from the approximated surface that smoothly connects gaps between the scan measurements $\mathbf{S}$, so that they fill up the gap between $\mathbf{Q}_s$

with an adaptive amount of points. When $q_n \in \mathbf{Q}_n$ falls within a narrow gap between two scanned regions, the extension should be small, whereas from a wider gap, the extended projection can be increased. However, there can be errors in the inference as the extension is based on continuity assumption without any physical measurements. We set the maximum size of projection to be less than the size of the scanner's viewing frustum to allow overlap between the new scan and the previous measurement and effectively compensate possible errors in predictions with the actual measurement.

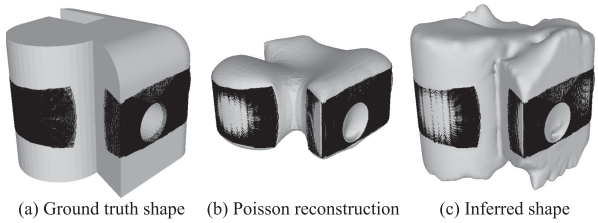
Given the maximum distance of region growth, we copy the primitive labels into $\mathbf{Q}_n$ according to the amount and extend the primitive shape. Specifically, we incrementally copy the neighboring primitive labels and frustum boundary classification result to $\mathbf{Q}_n$ starting from all the boundary points $\mathbf{B}$ with the detected primitive labels as shown in Fig. 7 (j). We start by adding $\mathbf{B}_f$ into the empty point set $\mathbf{G}$, which represents the region grown until now. Subsequently, we determine candidate points for the growth that are a subset of $\mathbf{Q}_n$ within a growing distance from $\mathbf{G}$. The candidate points are determined by if $\forall q_n \in \mathbf{Q}_n$, $\exists g \in \mathbf{G}$ such that $d(q_n, g) \leq r$, where r is the distance for each iteration of growth. After the candidate points are determined, the properties of the nearest point in the grown region $\mathbf{G}$ are copied to each candidate point. After all the candidate points are labeled with primitive and frustum boundary classification, they are added to $\mathbf{G}$ and the process is iterated again. The growth is continued until the accumulated growing distance becomes larger than a prefixed threshold distance, which is smaller than the size of the scanner's viewing frustum.

Now, we modify the position and normal of $\mathbf{Q}_n$ using their primitive labels and frustum boundary classification attached. We first filter points from $\mathbf{Q}_n$ that are grown from the frustum boundary by using the frustum boundary classification label on each point and also remove points with a nonprimitive label. As shown in Fig. 9(d), the nonscanned region grown from the object boundary or boundary point with a nonprimitive label in Fig. 9(c) is deleted. Therefore, the projection is only applied to those points grown from the frustum boundary points with a primitive label excluding nonprimitive ones. For the filtered points $\mathbf{Q}_{nf} \subset \mathbf{Q}_n$, we unfold the points using the outward direction v defined in Section V-A as depicted in Fig. 7(m) and project the unfolded points onto the virtual primitive surface by using the parameters of the primitive. The new positions effectively extend the primitives into the unknown region by the desired amount.

Specifically, for a point q_{nf} with the primitive label A, its reference boundary point b_r can be selected as the closest point among the boundary points $\mathbf{B}$ with the same primitive label A. Using b_r and its outward vector v_r, the position $P_{\tilde{q}_{nf}}$ of modified point $\tilde{q}_{nf}$ can be defined as

$$P_{\tilde{q}_{nf}} = \|q_{nf} - b_r\| v_r + (q_{nf} - b_r) - v_r^{(} q_{nf} - b_r) v_r \quad (1)$$

where $\|\cdot\|$ is the ℓ_2-norm. In (1), the term $(q_{nf} - b_r) - v_r(q_{nf} - b_r)v_r$ denotes a vector component of $(q_{nf} - b_r)$, which is orthogonal to v_r. The equation shows that the position of a modified point is first moved for a distance from its reference point in the corresponding outward direction and then the orthogonal component of $(q_{nf} - b_r)$ to the outward direction v_r is added to prevent concentration of modified points related to the same reference point in one direction. In short, we are effectively unfolding the boundary shape in the outward direction v while maintaining the distance from the original boundary point b. The unfolded points $\tilde{\mathbf{Q}}_{nf}$ are then projected onto the corresponding primitives that they belong to as depicted in Fig. 7(n).



(a) Ground truth shape (b) Poisson reconstruction (c) Inferred shape

Fig. 10. Visualization of original scanned points with ground truth model (a) and comparison between inferred shapes from Poisson reconstruction (b) and the proposed method (c).

D. Poisson Reconstruction Reapplied to the Extended Scanned Region

From the previous sections, a subset of $\mathbf{Q}_n$ is modified and projected to extend the detected primitives only from the frustum boundary $\mathbf{B}_f$. The extended primitive shape can then be merged with the original scanned points as depicted in Fig. 7(n). By reapplying Poisson reconstruction with the merged points, we obtain the new inferred shape $\hat{\mathcal{M}}$ of the target scanning object as shown in Fig. 10. Isopoints $\hat{\mathbf{Q}}$ are sampled from the new shape for the local surface assessment in the next stage.

VI. Quality Assessment of Inferred Shape and Next-Best-View Calculation

After we have the updated approximate surface with the detected primitives, we assign the score to represent the quality of the sample positions $\hat{\mathbf{Q}}$ of the inferred surface mesh. Subsequently, we locate the NBV of the scanner to capture the overlap between the confident and less confident regions. The quality assessment is described in Section VI-A, and the NBV calculation is presented in Section VI-B.

A. Quality Assessment

For each point $\hat{\mathbf{Q}}$ sampled from the surface approximation $\hat{\mathcal{M}}$, we calculate the confidence value $f \in [0, 1]$ that represents how sufficient the local scan measurement is. The inferred shape from Section V has the advantage of having a continuous primitive shape extended into the nonscanned region compared with the ordinary Poisson isosurface. We utilize the inferred primitives and provide higher priority (low confidence) to optimize the scanner's position to capture the unknown region. We still provide low confidence to the complex geometry of the sparsely scanned region but assign the region with lower priority than that of the nonscanned region. Consequently, our quality assessment is designed to scan the overall shape of the object first rather

than concentrating on the geometric complexity of the already scanned region.

We assess the scan quality of the neighborhood of each sample combining three terms. Two of them are the gradient confidence f_G and smoothness confidence f_S as suggested in [7]. The third criterion proposed in this study is related to the conformity of the local surface to a primitive shape f_P. We combine the three criteria to measure the sufficiency of the entire region such that we achieve the final surface representation either by reverse engineering using the detected primitives or by Poisson surface reconstruction.

First, the gradient confidence f_G, which depends on the local density of the scan measurements $\mathbf{S}$ [7], is calculated using Poisson scalar field Γ created in Section IV-A. Although $\hat{\mathbf{Q}}$ and $\hat{\mathcal{M}}$ are from the Poisson reconstruction of the modified samples $\tilde{\mathbf{Q}}_{nf}$, the gradient confidence of each isopoint $\hat{q}$ should be calculated using Poisson scalar field Γ obtained from the scanned points $\mathbf{S}$. The Poisson scalar field shows a high variation of scalar field at which the true scanned points reside, and hence, it reflects the existence of scanned points at each $\hat{q} \in \hat{\mathbf{Q}}$ accurately. The points $\hat{q} \in \hat{\mathbf{Q}}$ are further classified into two groups: the scanned region $\hat{\mathbf{Q}}_s$ whose gradient confidence is higher than f_{Gs}, and the nonscanned region $\hat{\mathbf{Q}}_n$ whose gradient confidence is lower than f_{Gs}. Instead of merely calculating the distance to the scanned points $\mathbf{S}$ as suggested in Section IV-A, we consider the density of the scanned points $\mathbf{S}$ with the Poisson scalar field Γ to decide whether the approximated shape can be trusted.

To calculate the confidence of the region with scanned evidence $\hat{\mathbf{Q}}_s$, we consider whether it belongs to a primitive region. If the distance between a point sample $\hat{q} \in \hat{\mathbf{Q}}_s$ and the closest detected primitive is smaller than the threshold distance $\mathbf{d}_P$, it is in a primitive region. The scanned points within a primitive are treated to be within a sufficiently scanned area, and hence, the highest confidence value of 1 is provided.

A portion of $\hat{\mathbf{Q}}_s$ that does not belong to a primitive region represents a typical Poisson surface reconstruction of the nonprimitive scanned area. Therefore, we apply the combination of gradient confidence f_G and smoothness confidence f_S in the same manner as in [7]. We normalize the confidence to be between f_{Gs} and 1, and consequently, it is always lower than that of the primitive region mentioned beforehand.

For the remaining isopoints $\hat{\mathbf{Q}}_n$, we apply the combination of gradient confidence f_G and smoothness confidence f_S again but it is normalized between 0 and f_{Gs}. As $f_G < f_{Gs}$ for $\hat{q} \in \hat{\mathbf{Q}}_n$, the confidence of the region is always lower than that of a region having scanned evidence.

The reason for discriminating confidence intervals for the scanned region and nonscanned region is to avoid assigning high confidence to the erroneous inference of the nonscanned region. Usually, the nonscanned region results in high f_S owing to its continuity in the normal direction from the nearby scanned points, and we might miss critical features in the nonscanned region owing to its higher confidence than that of the scanned region with low f_S. For example, we might scan parts of a plane in the object of interest, and fill the unobserved holes on the plane using the detected primitive. In this case, the confidence value normalized into the same interval results in a nonscanned region of the unobserved hole having higher confidence than that of the scanned region with rich geometric detail. By dividing the confidence of the two regions into different intervals, we guarantee the correct confidence that prioritizes adequate coverage of the underlying shape.

The overall process is shown in Fig. 11. A lower confidence indicates higher priority to be scanned by the NBVs. In summary, the final order of priority for scanning is as follows: nonscanned region with nonsmooth surface approximation, nonscanned region with smooth surface approximation, scanned region with no primitive detected and high geometric details, scanned region with no primitive detected and low geometric details, and scanned region with primitive detected. The region with f_G less than f_{Gs} is scanned with the highest priority, and subsequently, the nonprimitive region with the gradient confidence higher than f_{Gs} is scanned. Once the primitive is detected from a region with gradient confidence higher than f_{Gs}, it will not be a target for NBVs until the other regions are scanned and achieve high confidence close to 1.

B. NBV Calculation

Given the surface confidence f, we calculate the NBV jointly considering the scanner's optimal viewing range and the inferred shape approximation. On the one hand, we need to cover the region with low-quality assessment. On the other hand, we intend to maintain the overlap between the existing and new scans. As depicted in the example shown in Fig. 12(a), the inferred region that is distant from the scanned points is not always accurate. The risk of erroneous inference is higher as we move farther from the region with the scan evidence. In addition, the overlap of measurements is practically necessary to register the scans correctly into the global coordinate system.

With these principles in mind, we adapt the viewing vector field [7] to generate the NBV. Given the optimal working distance range of a scanner $[\mathbf{d}_n, \mathbf{d}_f]$, the search space contains all points whose distance to the closest point on the inferred shape is smaller than or equal to $\mathbf{d}_f$. For all voxels $\mathbf{v}_i$ in the search space, we calculate the NBV score, which measures its suitability as a viewpoint in the next scan. The NBV score interprets how well a point in the search space $\mathbf{v}_i$ can scan a low-confidence region considering the scanner's optimal working distance and the viewing angle relative to the position and the normal of the inferred shape. Each voxel center $\mathbf{v}_i$ is matched with its target isopoint $\hat{q}_k$ that can be best scanned from its position. $\mathbf{v}_i$ has high NBV score when its target isopoint $\hat{q}_k$ has low confidence, its normal is facing the view direction, and the distance is close to the optimal working distance of the scanner.

To maintain overlap with the scanned region, we constrain the matched isopoint $\hat{q}_k$ for NBV calculation to be a subset of the total isopoints $\hat{\mathbf{Q}}$ that are close to the scanned points within a threshold distance. We used one-third of the height of the scanner viewing frustum in our experiment. As shown in Fig. 12(b), NBVs are targeting isopoints near the scanned region, and hence, new scan measurements are overlapped with the existing scan.

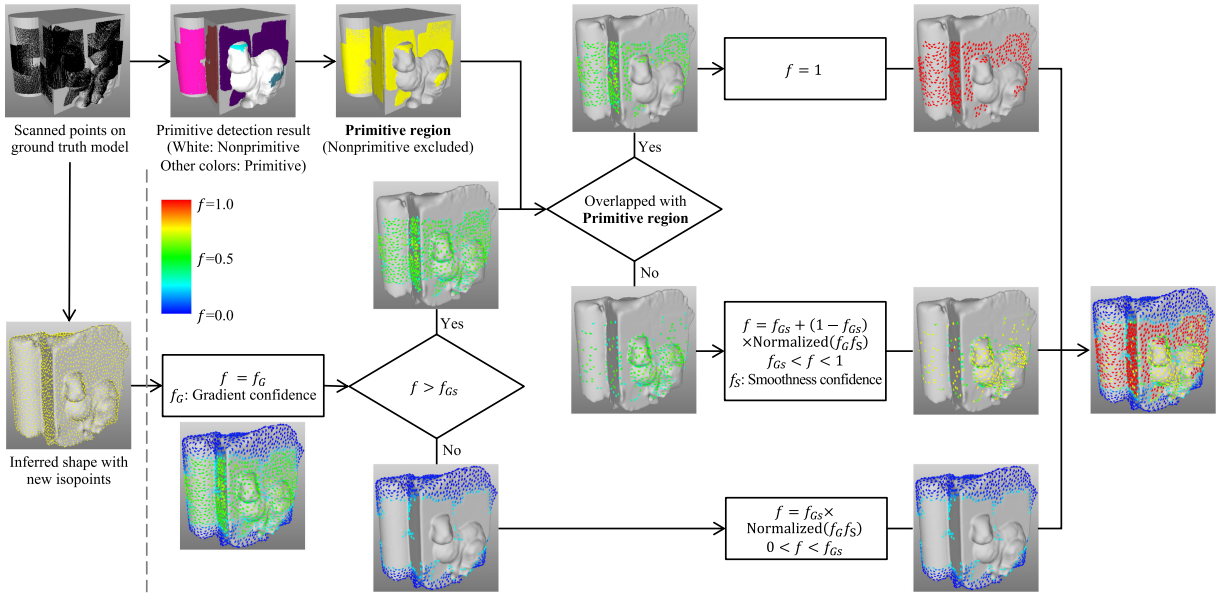


Fig. 11. Overview of quality assessment: Isopoints from the inferred shape are used for the assessment of local confidence. First, isopoints are categorized by scanned and nonscanned regions by using the threshold value of gradient confidence $\mathbf{f}_{Gs}$. For a scanned region, if it is matched with the primitive region, the confidence value of 1 is given. Scanned points $q_s \in \mathbf{Q}_s$ excluding nonprimitive labeled points are obtained as the primitive region. If the scanned region is not primitive, gradient and smoothness confidence are used together and a value higher than $\mathbf{f}_{Gs}$ is given. For a nonscanned region, a gradient confidence lower than $\mathbf{f}_{Gs}$ is given.

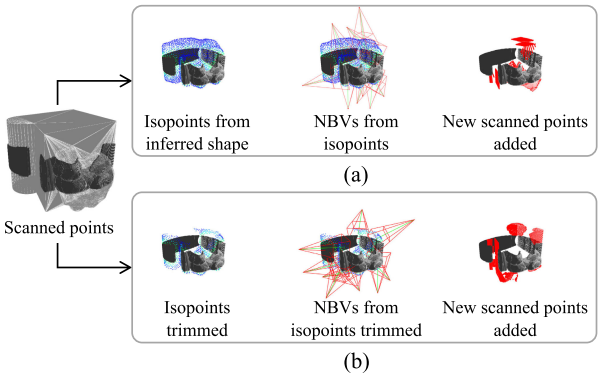


Fig. 12. Comparison between NBVs calculated from total isopoints (a) and trimmed isopoints that contain a subset close to scanned points (b). In (a), NBVs are targeting isopoints that are far from the scanned points and not accurate compared with the actual object. In (b), target isopoints of NBVs are selected conservatively only from a region close to the scanned points, and hence, the result of new scanned points covers many nonscanned regions.

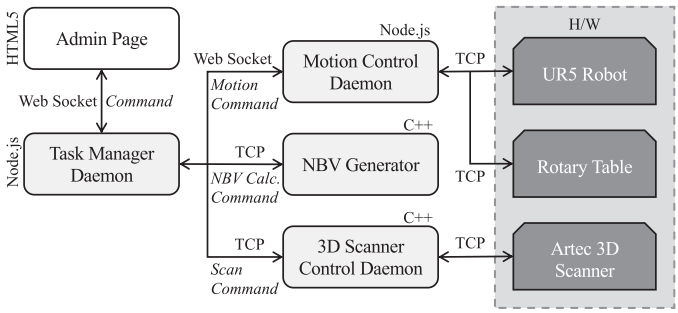


Fig. 13. Our autoscanning system is controlled by an HTML admin page, which controls the task manager daemon. In addition to the NBV generator that has been implemented with C++, two daemons control the scanner and the motion of the robot and rotary table, respectively. The control commands are sent either by web sockets or TCP commands.

VII. Real-World Implementation

The algorithm explained in the previous sections only considers the point cloud data and can be implemented with a virtual scanner. However, several aspects must be considered to use the system in the real world. In this section, we explain the conditions and further include calibration steps that must be considered to realize the automatic 3-D scanning process based on our algorithm. The overall system is depicted in Fig. 13.

A. Hardware Implementation

We used the Artec Space Spider 3-D scanner for scanning the real-world object. The suggested measurement range is 20–30 cm, and the viewing frustum is roughly the size of the name card, which is small compared with the size of parts we are considering (we used models within the bounding box whose dimensions are in the range of 20–40 cm). We designed the real-world setup to best represent the common use cases that we are targeting, and as the scanning volume for the individual frame is very small compared with the shape, it is crucial to have the optimal scanning path with a good estimation of the shape to obtain the best output model.

We place the scanner at the calculated NBV by combining the motions of a Universal Robots UR5 robot and a rotary table. The UR5 robot has a limited range of motion with many singularities and could by itself not reproduce all the motions generated by our NBV generator even though the robot has a reach of 850 mm. The rotary table can generate one-degree-of-freedom rotation

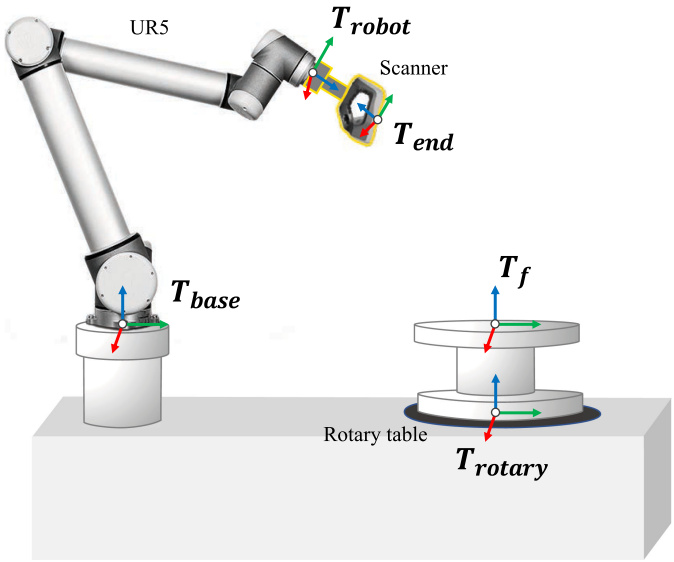


Fig. 14. Given NBV for the scanner position relative to the origin $\mathcal{T}_f$, the transformation is decomposed into the motions of robot T_{robot} and the rotary table T_{rotary} considering the offset between the robot hand and the scanner T_{end} and the offset introduced by the robot base T_{base}. The combined transformation is presented in (2).

around the origin, which considerably reduces the burden on UR5 and it could access most of the viewing direction.

We set the origin of the coordinate system at the center of the rotation of the rotary table. We calibrate the relative position of the end-effector of the robot and the origin of the depth frame by reverse engineering two fixed rigid transformations at the base and the end-effector of the robot considering possible tolerance of installation and precision of the robot, rotary table, and custom-made jig for the 3-D scanner. Specifically, we assume that there are two fixed rigid transformation matrices that should be multiplied:

$$\mathcal{T}_f = T_{\text{rotary}} \cdot T_{\text{base}} \cdot T_{\text{robot}} \cdot T_{\text{end}}. \tag{2}$$

In other words, the overall transformation that converts the measured points D_f into the global coordinate system $\mathcal{T}_f$ is represented as the sequential combination of transformation between the scanner's origin to the end-effector T_{end}, robot movement T_{robot}, the offset by the base of the robot T_{base}, and the rotary table T_{rotary}. The coordinate frames related to each of the transformation matrices are shown in Fig. 14. We produce $\mathcal{T}_f$ by controlling T_{rotary} and T_{robot}, and precalculate the fixed offset T_{base} and T_{end}.

In the real-world setup, we could not measure the bottom of the object. We assumed a scenario that generates two scans of the top and bottom of the object separately, which is the common approach for the scanning process. We limit the working volume to be above the rotary table accordingly and only produced NBV from above the rotary table. We also assumed that the user provides a tight bounding box of the object to provide a guide to the scanner during the early stage of the scanning. The bounding box is necessary to compensate for the limited scanning volume of the hardware and enable the scanner to find the object without collision.

B. Software Implementation

We seamlessly integrate the combination of hardware setup for an ordinary user to scan an object rapidly. Fig. 13 depicts the software integration of the control sequences for the given hardware. There is a web page written with HTML5 that provides a simple GUI to control and send the necessary commands through a web socket. The admin page is connected with a Javascript web socket module of the task manager daemon, which is connected with three modules of software. The first is the motion control daemon, which receives the motion command through a web socket and decomposes the NBV into the necessary motion command packets for the UR5 robot and the rotary table over TCP/IP. The second module of the software is a 3-D scanner control daemon, which sends necessary command packets to the Artec Space Spider 3-D scanner to scan the object when the motion is generated for the NBV. The third module of the software is our NBV generator written with C++. The 3-D scanner control daemon and the NBV generator are controlled by command packets from the task manager daemon. It observes the merged scan models and generates the new NBV according to our algorithm described in Section V.

C. Data Flow

With the help of the task manager daemon, the three modules described in Section VII-B seamlessly send information to each other. The NBV generator will send a set of NBVs in the form of the transformation matrix, and then the motion control daemon moves the scanner to the NBV. When the motion is complete, the 3-D scanner daemon allows the scanner to generate scan data at NBV. After the new set of scan data is generated, the transformation-scan pairs are sent to the NBV generator to calculate the next NBV. The process is iterated until the NBV generator decides that the scan is complete.

VIII. Results

We demonstrate the performance of our primitive detection and NBV calculation compared with those of the state-of-the-art algorithms. The pipelines are verified through virtual simulation and real experiments.

The virtual simulation is performed on a PC with an Intel Core i7-6700 K 4 GHz CPU and 16 GB RAM. Our primitive detection pipeline is implemented with MATLAB and the virtual scanning with optimized NBVs was implemented with C++ using OpenGL and Qt. For the test models, we use 12 machine parts models composed of various types of primitives as shown in Fig. 15. The virtual scanner's view frustum is set up following the specification of the Artec Space Spider scanner and the size of the model is set to mimic the actual size of the corresponding mechanical part or object. The initial scanning views and the number of NBVs calculated at each iteration are predefined and set depending on the size of the scanned object. The iteration of virtual scanning proceeds until the coverage value of the original digital model by the sampled points $\mathbf{Q}$ and by the scanned points $\mathbf{S}$ both exceed over 90%. In addition, it also stops when there is no improvement in the coverage in the series of iteration. The

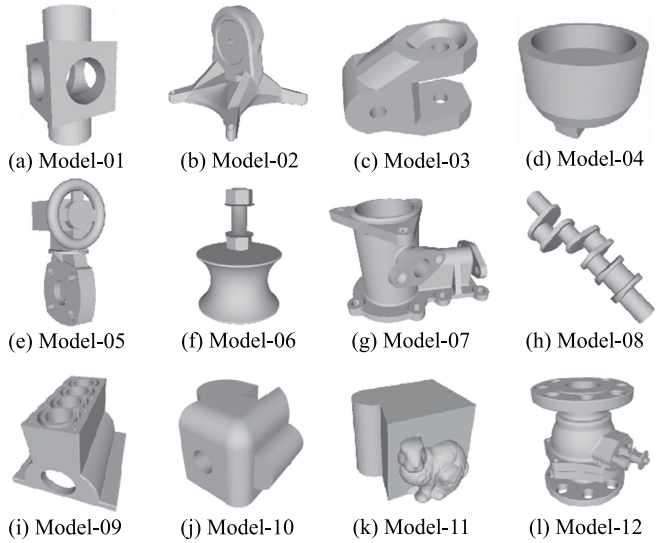


Fig. 15. Test models for virtual scanning. The models were provided from the following sources. (a) AIM@SHAPE Shape Repository, (b)–(d) Zortrax. (e) and (f) Dassault Systèmes. (g)–(i) GrabCAD. (j) and (k) Self-produced model. (l) Self-produced model based on actual mechanical part.

coverage of the digital model by the $\mathbf{S}$ is calculated by the ratio of a number of sampled points from the model for which there exists a point in $\mathbf{S}$ within a threshold distance to the number of total sampled points from the model. The coverage by the sampled points $\mathbf{Q}$ is calculated similarly.

For the real-world experiment, the Artec Space Spider scanner is attached to a robot arm UR5 and a turntable to provide movement and rotation to capture the desired view as in Fig. 1. The NBV calculation is performed on a PC in the same environment as the virtual simulation. The initial scan views are calculated considering the user input of the size of the bounding box with the model placed on the turntable. The termination criterion of the experiment is when there is no improvement in confidence for the isopoints in the series of iteration.

A. Primitive Detection Compared With the State-of-the-Art Algorithms

Our primitive detection algorithm is compared against two of previous works, the RANSAC-based approach [23] and the robust algorithm considering dominant directions [22].

Fig. 16 shows the qualitative visualization of primitive detection in a partial scan by model. The primitive regions are represented in colors by the type of primitives. We demonstrate the performance on the uniform samples $\mathbf{Q}$ of Poisson surface reconstruction mesh $\mathcal{M}$ extracted from the scan data. Tests have been performed under the same condition, by detecting primitives at the same resolution of sampled points on the scanned region, which is the preprocessing step in our algorithm. Our algorithm outperforms the other two approaches and stably detects the correct types of primitives. The RANSAC-based approach is quite fast and can detect primitives from small parts. However, it can make a wrong prediction for the types of primitives through its local feature detection approach. The Le and Duan [22] approach globally identifies the primitive, but this depends on the ratio of the primitive to the total area. In contrast, our algorithm not only detects the primitives with high accuracy but also correctly expands their region including both scanned and nonscanned areas connecting isolated or distorted parts owing to incomplete scanning. Compared with the algorithm in Le and Duan's [22], the improvement in recognizing the primitives has been explained in Section IV, thus determining the detailed orientation of the model in the segmented region in addition to other supplement processes. The relative improvement is prominent especially during the earlier stage of the scanning, where the object shape is not known, and the correct scan-path generation is critical. We also sometimes determine fewer numbers of primitives, as we focus on the matched surface with the actual scanned points and apply a tight-error threshold when accepting the detected candidate primitives.

TABLE I
COMPUTATION TIME (S) OF PRIMITIVE DETECTION FOR INTERMEDIATE SCAN DATA USING THE SAMPLE MODELS SHOWN IN FIG. 16

#Model	Le & Duan	Ours
01	57.71	30.18
02	143.62	100.91
03	87.99	52.88
05	128.63	120.88
06	74.06	69.51
07	69.69	54.81
08	198.53	32.94
10	36.44	43.7

The primitive detection pipeline can be time-consuming especially considering running it for every addition of new scan data. The computation time of our algorithm does not falloff compared to Le and Duan [22], as shown in Table I, which presents the time taken for the sample models in Fig. 16. We only sample a subset of data to optimize the performance and also consider only the scanned region. Moreover, we reuse the primitives detected from the previous iterations to avoid repeated computation, which will reduce the computation time even more.

B. Comparison With State-of-the-Art NBV Technique Using Virtual Experiment

We compare our proposed method with the quality-driven, Poisson-guided autoscanning algorithm [7] which represents a state-of-the-art algorithm. The implementation of the Poisson-guided method is based on the code publicly shared by the author of [7] in GitHub. To observe the difference between our method and the Poisson-guided method, we implemented both algorithms using the same normalizing scale factor, the same Poisson reconstruction with identical parameters, and the same parameters for calculating gradient confidence and smoothness confidence. The difference is that our method has an additional shape inference stage that uses the primitive information extracted and an additional primitive-based confidence term. Therefore, we confirm that the difference in the result is not caused by the quality of implementation, as both are similar except for the shape inference stage added to our method.

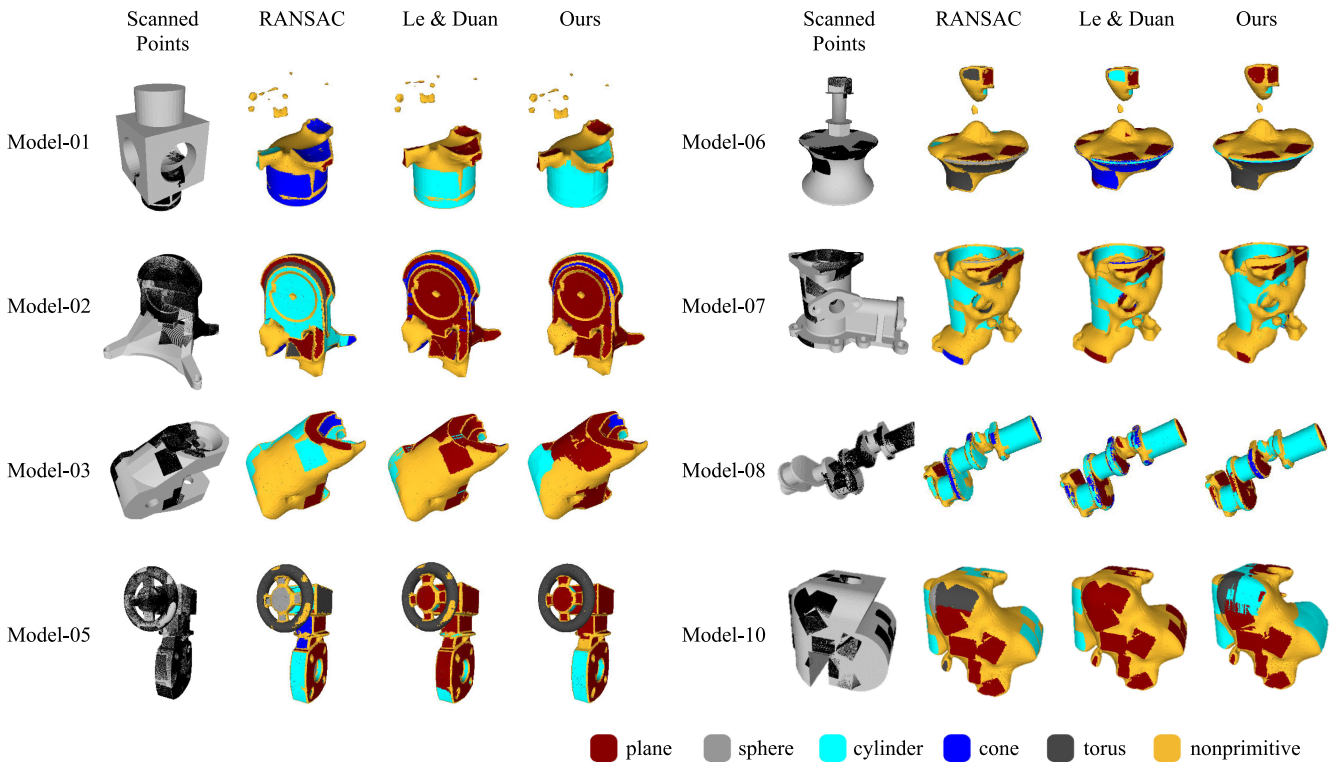


Fig. 16. Comparison of primitive detection algorithms on scanned data at midprocess of autoscanning. In each case, the state of scanning on the test model (left-most) is shown with the segmentations representing the types of primitives by colors. The algorithms are run on the same condition, sampled points $\mathbf{Q_s}$ from the reconstructed mesh $\mathbf{M}$, except that our algorithm also uses the artificial region $\mathbf{Q_n}$ for extending the primitive region.

First, a qualitative comparison between the two methods is shown in Fig. 17. In Poisson-guided autoscanning, the isosurface of Poisson field is used as the approximation of the surface. This leads to the inferred shape of unknown object being less similar to the ground truth model compared with that in our approach and affects the position and direction of NBV calculated from it. In the early iteration of our approach, a large nonscanned region connecting sparse scanned regions participates in the extension of primitives that are discontinued at the frustum boundary, and hence, the shape inferred is larger than the isosurface of the Poisson-guided method. As the iteration proceeds, the amount of extension is reduced owing to the reduction of *nonscanned* $\mathbf{Q}_n$ created by Poisson isosurface. This is reasonable because the scanned region completes the shape as the iteration proceeds, and the predicted amount of extension should be less.

In terms of NBV selection, NBVs were generated outside the bounding box of the object assuming they are provided by the user in a real scanning environment before the scanning process. NBVs are selected to target low-confidence isopoints that usually occur at the nonscanned region far from the scanned region owing to low f_G or at the region with the maximum difference in normal compared with its nearest scanned region owing to low f_S.

In the case of Model-11, iteration 1 in Fig. 17, NBVs from both methods were selected to view the points at the nonscanned region that are far from the scanned region with the constraint that the NBVs can only exist outside the bounding box of the object. In our approach, NBVs targeted the extended nonscanned surface that formed a continuous primitive from the scanned region and resulted in a large amount of new scanned points, shown in yellow. NBVs from the Poisson-guided method viewed the points on the nonscanned surface which is curved from the boundary of the scanned region. In the Poisson-guided method, the inferred surface ignored the continuity of the primitive shape and resulted in fewer new scanned points by generated NBVs.

The effect of quality assessment is shown in the case of Model-09 in Fig. 17. In our approach, NBVs were calculated focusing on covering the nonscanned region rather than concentrating on holes and the complex shape at the top, whereas the NBV calculation in the Poisson-guided method is concentrated on the complex shape at the top even when the nonscanned region remaining. This is due to the discrimination in confidence between the nonscanned region and scanned region in our method inducing faster understanding of the overall shape before going into detail. Once the nonscanned region is eliminated, our approach concentrates on the complex shape similar to the Poisson-guided method. It is verified in iteration 16 of Model-07 in Fig. 17 that our approach concentrated on small holes and curved edges after finishing scanning the overall shape, whereas the Poisson-guided method is still incomplete in scanning the overall shape.

For the quantitative comparison between the methods, virtual simulations of iteration are performed as shown in Fig. 18. The similarity between the target model and the reconstructed surface $\mathbf{M}$(left) is measured by the coverage of the model surface by the sampled points $\mathbf{Q}$. It represents how well the overall shape can be understood by the scanned points. Only a selective scanning

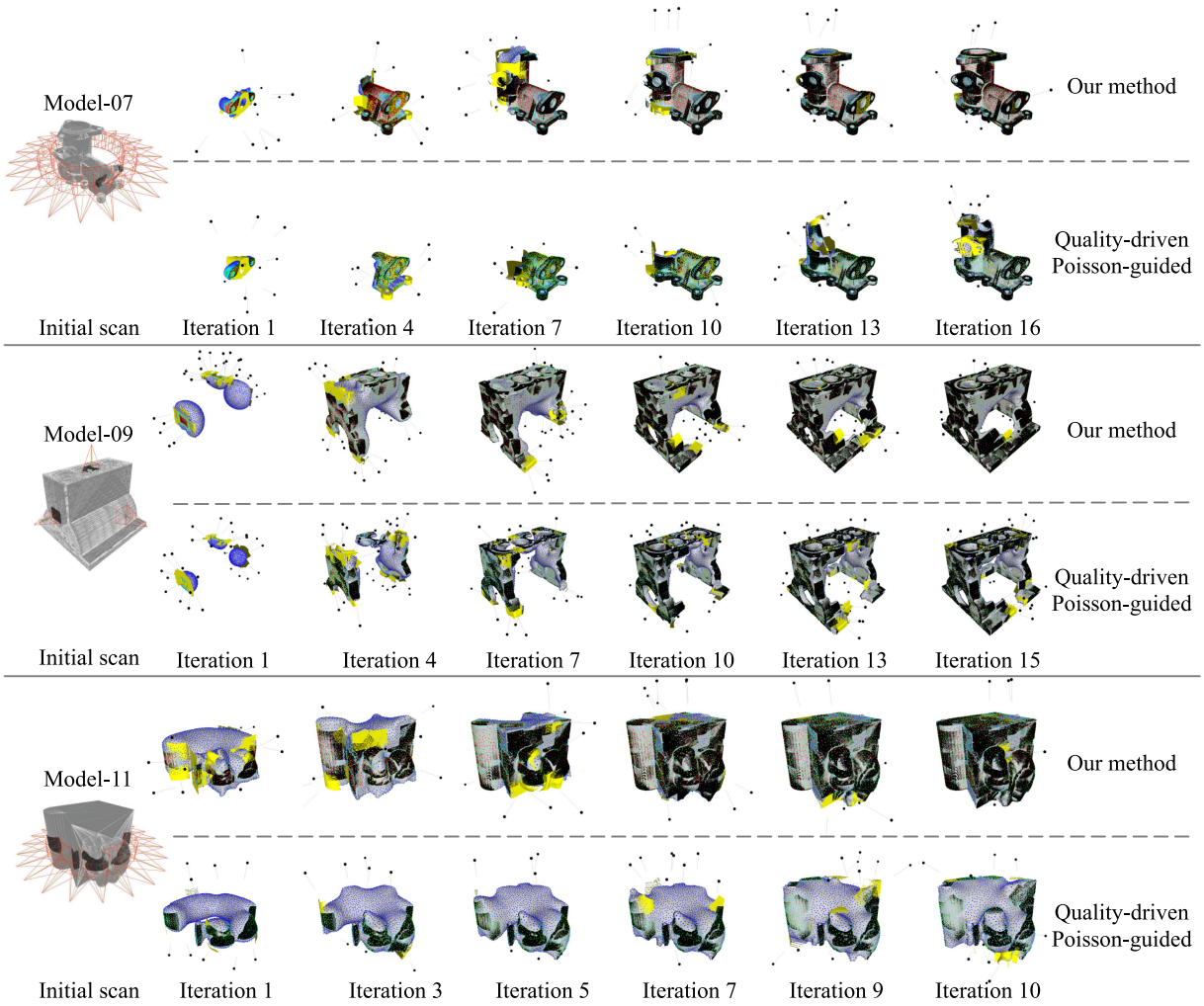


Fig. 17. Comparison of autoscanning progress using our approach and the quality-driven, Poisson-guided method [7] in virtual environment. The inferred surface model is in gray, and black points on the model are scanned points $\mathbf{S}$. Colored dots on the surface denote the quality of scanning at the location. Yellow points are points newly scanned by NBVs shown in bold black points with a blue line, which shows the direction of the view.

of edges or the boundary of the object is required to improve this measure. In contrast, the coverage by the scanned points $\mathbf{S}$ represents simple coverage by scanned points, which can be improved by scanning with the least overlap and scanning everywhere in balance.

From the results obtained for Model-02, Model-04, Model-06, Model-10, and Model-11, it can be observed that the proposed approach outperformed the Poisson-guided method in the case of an object with a bulk shape without small concave features. The Poisson-guided method performs reconstructions only based on scanned points and information regarding their normals. It is effective at estimating the shape that encloses the scanned part of the object by connecting the scanned region smoothly by only considering the normal of the points at the boundary; however, the result is not sophisticated enough to enable the extrapolation of a discontinued primitive as in our method. As illustrated in Fig. 10, primitive information regarding the scanned regions is extrapolated to estimate, as a continuous primitive shape, the adjacent area that is discontinued by the frustum boundary. Intelligent extrapolation to the nonscanned region leads to a better inference of the scanned object and the NBVs that are calculated to scan the unknown part of the object, especially for bulky shapes that consist of a continuous primitive surface.

For objects with small holes or slits such as Model-03, Model-07, Model-09, and Model-12, our method did not exhibit substantial difference from the Poisson-guided method. In the case of Model-07, the overall shape is scanned more rapidly than in the Poisson-guided method, but because of small holes that needed to be scanned in detail subsequently, the effect of early scanning is compensated and the coverage at the later iteration becomes similar in both methods. The quality assessment focused on the nonscanned region makes inefficient NBVs that focus on the region that cannot be scanned from the viewpoint outside the bounding box of the object. This also makes slower progress at the end of the graph.

For objects with holes and slits such as Model-09, both methods do not show good performance owing to the large surface area relative to volume. A shape with deep holes and slits increases the total area of the surface and it weakens the advantage of our method in the graph. Even though the overall shape is scanned rapidly, the improved coverage is low because the outer surface comprises a small portion of the total surface.

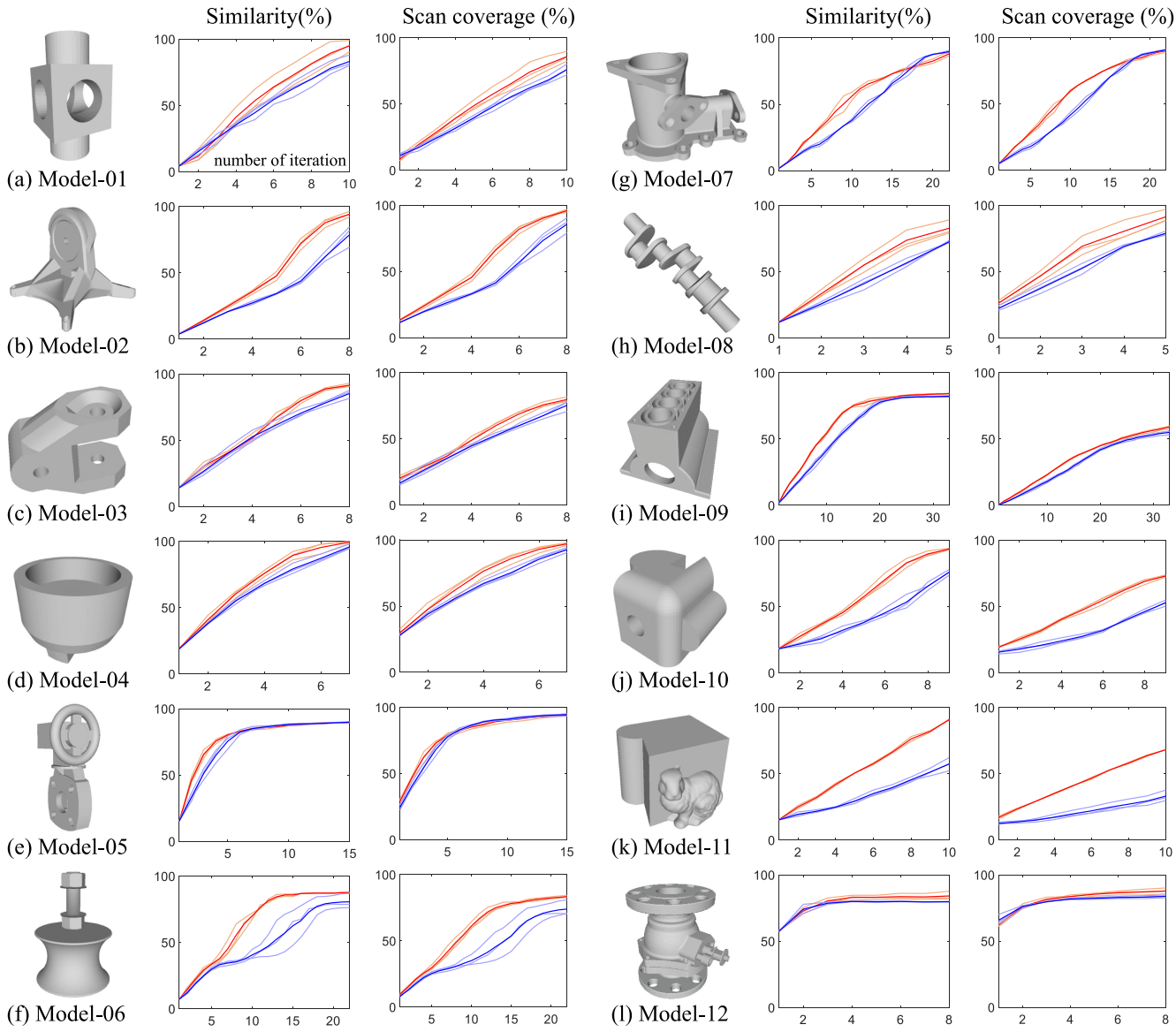


Fig. 18. Comparison of quantitative change in similarity and scan coverage between the suggested method and the quality-driven, Poisson-guided method [7]. For each model, the graph shows the similarity of the reconstructed surface $\mathbf{M}$(left) to the target model and the scan coverage of the target model during autoscanning. The similarity is measured by the coverage of the ground truth model's surface by the sampled points $\mathbf{Q}$, and the scan coverage is measured by the coverage of the ground truth model's surface by the scanned points $\mathbf{S}$. For each model, the experiment was performed three times, drawing the mean values in vivid lines along with the results of each experiment in faint color (red: our method; blue: Poisson-guided method).

Moreover, the surface inside the slit is difficult to reach by the scanner after the overall shape scanning, and it results in gradual improvement after its upper limit is reached.

The proposed method includes Poisson reconstruction, primitive detection, frustum boundary classification, primitive-based extension, isopoints trimming, and confidence evaluation, which consist of gradient, smoothness, and primitive-based confidence calculation. Conversely, the Poisson-guided method contains only Poisson reconstruction and gradient and smoothness confidence evaluation among its components. The proposed method and the Poisson-guided method are compared with respect to time consumption in Table II. The proposed method requires more time for primitive-based shape inference, which is not rigorously optimized.

To observe the contribution of each stage of the primitive-based shape inference in Table II to the scanning progress by NBVs, comparisons among methods, including the proposed method, the method without frustum boundary classification, the method without isopoints trimming, the method without any of them, and the Poisson-guided method, are used. The progress of the surface coverage of the ground truth model by sampled points $\mathbf{Q}$ and scanned points $\mathbf{S}$ is shown in Fig. 19. By comparing the methods with and without each component, it was found that the simultaneous consideration of the frustum boundary and isopoints trimming demonstrates faster convergence to the maximum coverage by the sampled points $\mathbf{Q}$. In contrast, when either the frustum boundary classification or the isopoints trimming is separately applied, the progress of the coverage by sampled

TABLE II
COMPUTATION TIME (S) OF EACH STAGE OF THE FIRST ITERATION OF NBV CALCULATION AFTER THE INITIAL SCANNING FOR MODEL-10 SHOWN IN FIG. 15

Stage	Poisson-guided	Ours
Poisson reconstruction	2.9	3.41
Primitive detection	n/a	11.33
Frustum boundary classification	n/a	0.08
Primitive based extension	n/a	13.17
Isopoints trimming	n/a	0.02
Gradient Confidence eval.	2.80	0.91
Smoothness Confidence eval.	0.08	0.07
Primitive Confidence eval.	n/a	0.09
Total	5.77	29.09

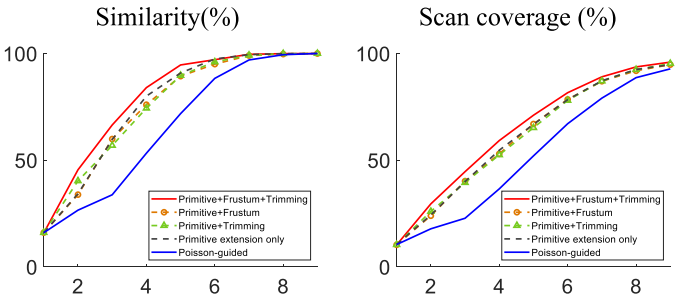


Fig. 19. Comparison of the methods using different combinations of the components in the proposed method and the Poisson-guided method applied to the Model-11 in Fig. 15. The size of the model is 300 mm, which is different from the size of 800 mm used in the experiment in Fig. 18.

points $\mathbf{Q}$ is deteriorated or not improved substantially compared to the case where neither of them is applied. Therefore, the results confirm that view frustum consideration and isopoints trimming should be applied together to improve the scanning progress by NBVs. As shown in Table II, the time consumption of the two stages is insignificant.

C. Comparison With State-of-the-Art NBV Techniques Using Real-World Experiment

Real-world scanning differs from virtual scanning in that there always exists some amount of noise in the scan data depending on the material of the object or the specification of the scanner. In addition, there are restrictions on the movement of scanner in order to exclude the risk of collision with the surrounding environment. For real-world scanning, we used Model-10 and Model-12, which are also subjected to virtual scanning. Fig. 20 shows the key operations for searching NBV mid-progress when scanning Model-10. The Model-10 is 3-D printed in size of $30 \times 25 \times 22$ cm. Without providing the shape information except for its bounding box for avoiding a collision, our algorithm detects the primitives constituting the shape based on the accumulated scan data and implements the confidence map that generates NBVs to scan the unknown region. Fig. 21 qualitatively compares the progress in early, mid, and later stages.

D. Limitation

Despite these improvements and the robustness of our method, there are certain limitations that need to be overcome.

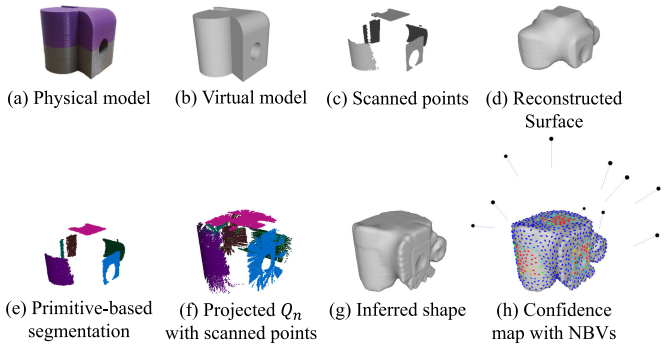


Fig. 20. Visualization of intermediate progress of autoscanning in real-world Model-10. (c) Scanned points. (d) Poisson reconstructed mesh based on scanned points (c). (e) Primitive detection on $\mathbf{Q_s}$. (f) Projection of $\mathbf{Q_n}$ using primitive information. (g) Inferred shape. (h) Confidence map with top 10 NBVs.

The objective of our system is the real-time implementation of primitive detection. Consequently, computation time is a critical issue. However, the supplementary processes that enable precise primitive detection with incomplete scan data reveal the considerable computing time as the scan data continues to accumulate and the size of the area is enlarged. Accordingly, the reuse of the primitives detected in previous iterations has been offered as a solution. Reuse of the precalculated primitive detection data can reduce repeated computation and computation time. However, the possibility of confusion arises when detection is performed from a partial region of one primitive. All primitives can be described as being composed of small plane patches, and the primitives of rotation can be accepted in lieu of each other in some cases. Such confusions can arise as there is no limit to the size of the primitives that comprise the model, and the current primitive detection method does not consider this problem.

Second, the limitation caused by the Poisson reconstruction remains. The holes or nonuniformity of scanned points can yield unwanted shapes, and the quality of the reconstruction depends on the octree depth. However, the limitation caused by the Poisson reconstruction can be resolved by resampling the points of uniform density from the Poisson surface and extracting the scanned region, which is the portion of the surface where the original scanned points exist with a sufficient level of density. To maintain the resolution of the reconstruction considering the scale, the depth of the octree was fixed and we only adjusted the size of the model with a normalizing scale. Robustness is obtained by using a credible part of the Poisson surface, where the reconstructed surface follows the shape of the scanned points relatively well compared to the region with holes or low and nonuniform density. However, the Poisson surface reconstruction is still vulnerable to inaccurate normalization of the scanned points estimated during the real-time scanning process.

In cases of partial occlusion, primitive-based extensions should not occur at the boundary created by the occlusion. This is due to the view frustum that detects the boundary within the view frustum. In that case, the partially occluded region will be inferred as a connected surface and the NBVs in the next iteration would target the region to scan the occluded space. However,

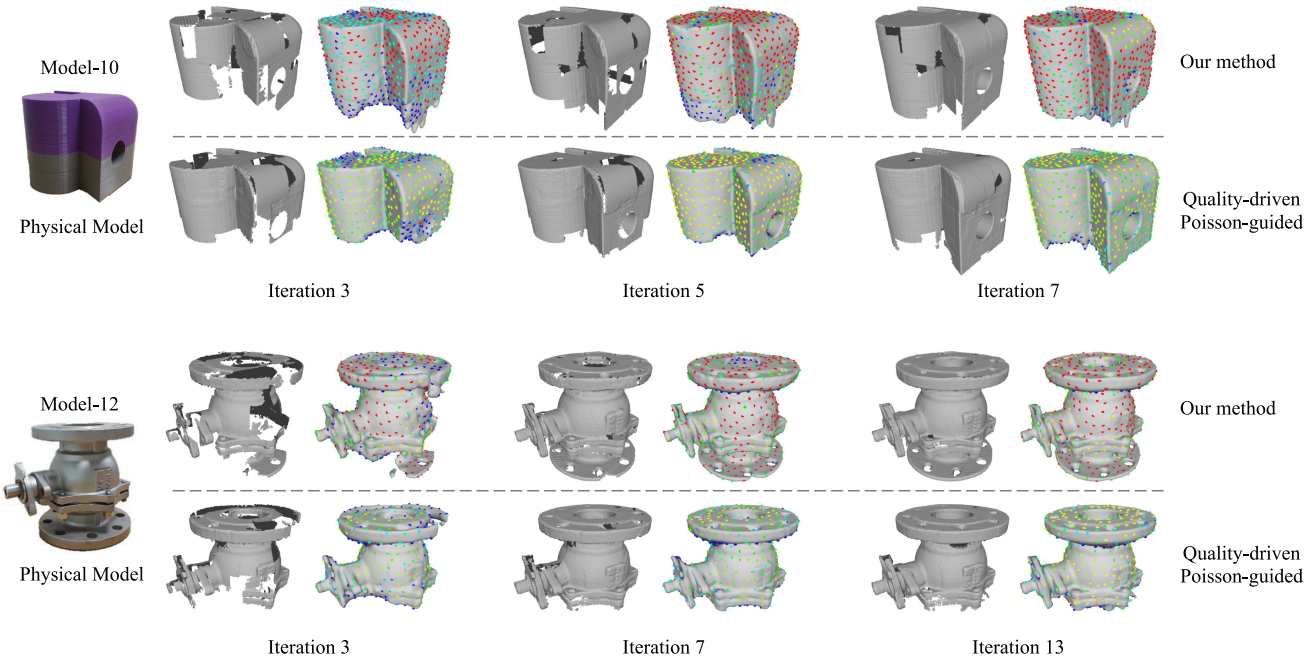


Fig. 21. Comparison of autoscanning progress using our suggested method and the Poisson-guided method [7] in real-world Model-10 and Model-12 showing scanned points (left) and isosurface with confidence map (right).

there is the case wherein a false classification of the frustum boundary occurs when the boundary of the view frustum is overlapped by the boundary of the object in which the occlusion occurs. This happens because the boundary of the scanned region is classified as a frustum boundary when it is near the boundary of the view frustum space. In that case, the false extension of the occluded boundary could disrupt the shape inference results. The trimming of isopoints reduces the inaccuracies that occur in such situations. However, additional NBVs would be needed to verify whether the extrapolation is correct.

The proposed method is time-consuming due to primitive detection and the processing of points for the extension using the detected primitives. However, the experimental results also arise from the compatibility issue between the detection of primitives in MATLAB, shape inference in a C++ application, and multiple file compilations performed to visualize the result.

IX. Conclusion

In this article, we presented an automatic pose generation approach for robotic 3-D scanning systems designed for reverse engineering mechanical parts that consist of primitive shapes. The process was time-consuming and repetitive but our approach automated the process by using three novel techniques: primitive detection, which can be applied on partial scanned points, inference of shape using the detected primitives, and quality assessment, which prioritizes every point on the inferred shape for NBVs. To determine the NBVs, one needs to infer the shape of the object first from partial information and the novelty of our approach was to use primitive information extracted from scanned points, which we can use in accordance with the measurement itself. Subsequently, one needs to determine which part of the inferred shape should be scanned first to help understand the object best. Our approach achieved this by proposing a method for local confidence assessment for each point on an inferred shape considering the coverage of the scanned points, geometric detail of the surface, and primitive conformity.

We also implemented the approach in an actual setup that can be readily used by a casual user without any prior experience with 3-D scanning setups. We demonstrated the robustness and performance gain achieved with our suggested algorithm compared with the state-of-the-art autoscanning process. By considering the detected primitives in the challenging setup, our suggested pipeline can facilitate a widespread usage of 3-D scanning and 3-D printing.

In the future, we would like further to develop the efficiency and reliability of our primitive detection algorithm. Despite efforts to reduce the computation time, the process remains time-consuming as the size of the dataset increases, which led us to reuse the previous results as a solution. However, the primitive detection can be inaccurate on some partial region of one primitive. Therefore, the criteria for accuracy in primitive detection and a strategy for reconsideration, to achieve optimal determination as new data is acquired, should be established.

Additionally, shape inference using primitives can be improved by inferring only the neighborhood of scanned points instead of creating an entire watertight surface that encloses the points. The currently used algorithm coupled with Poisson reconstruction inevitably creates watertight surfaces and extracts a portion of the surface adjacent to the scanned region as a target geometry for NBV calculations. This confirms that a

virtual surface distant from the original scanned points is not required.

Shape inference can be improved by using topological relationships among primitives or by adapting the energy function in [23] to our method. This would decrease the computation time consumed by the processing of the sampled points from the reconstructed surface. Conversely, Poisson reconstruction is replaceable with other methods of surface reconstruction, should they have better performance in creating accurate watertight polygons out of oriented 3-D points.

Optimization of processes, such as implementing detection and shape inference in the same environment, is also required to reduce computation time.

References

[1] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 303–312. [Online]. Available: http://doi.acm.org/10.1145/237170.237269

[2] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992. [Online]. Available: http://dx.doi.org/10.1109/34.121791

[3] S. Son, H. Park, and K. H. Lee, "Automated laser scanning system for reverse engineering and inspection," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 8, pp. 889–897, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890695502000305

[4] M. Karaszewski, M. Stepień, and R. Sitnik, "Two-stage automated measurement process for high-resolution 3D digitization of unknown objects," *Appl. Opt.*, vol. 55, pp. 8162–8170, Oct. 2016.

[5] S. Khalfaoui, R. Seulin, Y. Fougerolle, and D. Fofi, "An efficient method for fully automatic 3D digitization of unknown objects," *Comput. Industry*, vol. 64, no. 9, pp. 1152–1160, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361513000778

[6] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *J. Real-Time Image Process.*, vol. 10, no. 4, pp. 611–631, Dec. 2015. doi:10.1007/s11554-013-0386-6.

[7] S. Wu *et al.*, "Quality-driven poisson-guided autoscanning," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 203:1–203:12, Nov. 2014. doi: 10.1145/2661229.2661242.

[8] F. Buonamici, M. Carfagni, R. Furferi, L. Governi, A. Lapini, and Y. Volpe, "Reverse engineering modeling methods and tools: A survey," *Comput. Aided Des. Appl.*, pp. 1–22, 11 2017.

[9] A. D. Sappa, A. Restrepo-specht and M. Devy, "Range image registration by using an edge-based representation," in *Proc. 9th Int. Symp. Intell. Robot. Syst.*, 2001, pp. 167–176.

[10] G. Shen, T. Sakkalis, and N. Patrikalakis, "Analysis of boundary representation model rectification," in *Proc. 6th ACM Symp. Solid Model. Appl.*, 2001, pp. 149–158. [Online]. Available: http://doi.acm.org/10.1145/376957.376975

[11] M. A. Garcia and L. Basanez, "Fast extraction of surface primitives from range images," in *Proc. 13th Int. Conf. Pattern Recognit.*, Aug. 1996, vol. 3, pp. 568–572.

[12] A. W. Fitzgibbon, D. W. Eggert, and R. B. Fisher, "High-level CAD model acquisition from range images," *Comput. Aided Des.*, vol. 29, no. 4, pp. 321–330, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010448596000590

[13] D.-M. Yan, W. Wang, Y. Liu, and Z. Yang, "Variational mesh segmentation via quadric surface fitting," *Comput. Aided Des.*, pp. 1072–1082, 2012. [Online]. Available: https://www:microsoft:com/en-us/research/publication/variational-mesh-segmentation-via-quadric-surface-fitting/

[14] R. Bénière, G. Gesquière, F. Le Breton, W. Puech, and G. Subsol, "A comprehensive process of reverse engineering from 3D meshes to CAD models," *Comput. Aided Des.*, vol. 45, no. 11, pp. 1382–1393, Nov. 2013. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-00857347

[15] T. Du *et al.*, "InverseCSG: Automatic conversion of 3D models to CSG trees," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 213:1–213:16, 2018.

[16] H. Lin *et al.*, "Semantic decomposition and reconstruction of residential scenes from Lidar data," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 66:1–66:10, Jul. 2013. doi: 10.1145/2461912.2461969.

[17] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra, "RAPter: Rebuilding man-made scenes with regular arrangements of planes," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 103:1–103:12, 2015.

[18] M. Lin, T. Shao, Y. Zheng, N. J. Mitra, and K. Zhou, "Recovering functional mechanical assemblies from raw scans," *IEEE Trans. Visualization Comput. Graph.*, vol. 24, no. 3, pp. 1354–1367, Mar. 2018.

[19] H. Laga, M. Mortara, and M. Spagnuolo, "Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes," *ACM Trans. Graph.*, vol. 32, no. 5, pp. 150:1–150:16, Oct. 2013. doi: 10.1145/2516971.2516975.

[20] Y. Li, X. Wu, Y. Chrysanthou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "GlobFit: Consistently fitting primitives by discovering global relations," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 52:1–52:12, 2011.

[21] I. Kovács, T. Várady, and P. Salvi, "Applying geometric constraints for perfecting CAD models in reverse engineering," *Graph. Models*, vol. 82, pp. 44–57, Nov. 2015. doi: 10.1016/j.gmod.2015.06.002

[22] T. Le and Y. Duan, "A primitive-based 3D segmentation algorithm for mechanical CAD models," *Comput. Aided Geom. Des.*, vol. 52, pp. 231–246, Mar. 2017. doi: 10.1016/j.cagd.2017.02.009

[23] R. Schnabel, P. Degener, and R. Klein, "Completion and reconstruction with primitive shapes," *Comput. Graph. Forum (Proc. Eurographics)*, vol. 28, no. 2, pp. 503–512, Mar. 2009.

[24] Y. M. Kim, N. J. Mitra, Q.-X. Huang, and L. J. Guibas, "Guided real-time scanning of indoor objects." *Comput. Graph. Forum*, vol. 32, no. 7, pp. 177–186, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/cgf/cgf32.html#KimMHG13

[25] B. Hepp, M. Nießner, and O. Hilliges, "Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Trans. Graph.*, vol. 38, no. 1, pp. 1–17, 2018.

[26] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark," in *Proc. SIGGRAPH Asia 2018 Tech. Papers*, 2018, p. 183.

[27] W. Jing, J. Polden, P. Y. Tao, W. Lin, and K. Shimada, "View planning for 3D shape reconstruction of buildings with unmanned aerial vehicles," in *Proc. IEEE 14th Int. Conf. Control, Autom., Robot. Vision*, 2016, pp. 1–6.

[28] Q. Wu, J. Lu, W. Zou, and D. Xu, "Path planning for surface inspection on a robot-based scanning system," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, 2015, pp. 2284–2289.

[29] C. Connolly, "The determination of next best views," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1985, vol. 2, pp. 432–435.

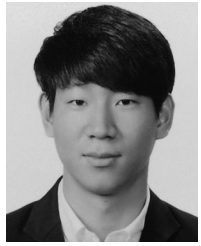
[30] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1016–1030, Oct. 1999.

[31] C. Peng and V. Isler, "Adaptive view planning for aerial 3D reconstruction of complex scenes," 2018, *arXiv:1805.00506*.

[32] K. Xu *et al.*, "Autoscanning for coupled scene reconstruction and proactive object analysis," *ACM Trans. Graph.*, vol. 34, no. 6, p. 177, 2015.

[33] R. Hu *et al.*, "Semantic object reconstruction via casual handheld scanning," in *Proc. SIGGRAPH Asia Tech. Papers* (ser. SIGGRAPH Asia '18, New York, NY, USA). ACM, 2018, pp. 219:1–219:12. doi: 10.1145/3272127.3275024.

[34] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.* (ser. SGP '06, Aire-la-Ville, Switzerland). Switzerland: Eurographics Association, 2006, pp. 61–70. [Online]. Available: http://dl.acm.org/citation.cfm?id=1281957.1281965



Inhwan Dennis Lee received the B.Eng. degree in mechanical engineering from the University of California, Los Angeles, USA, in 2013, and the M.S. degree in ocean systems engineering in 2015 from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, where he is currently working toward the Ph.D. degree in mechanical engineering.

His current research interests include geometry analysis, computer vision in 3-D, and 3-D scanning.

Ji Hyun Seo received the M.S. degree in computer science from the Ewha Womans University, Seoul, South Korea, in 2018.

She is a Research Assistant with the Korea Institute of Science and Technology, Seoul, South Korea. Her current research interests include 3-D scanning, computer vision, and 3-D reconstruction.

Young Min Kim received the B.S. degree from the Seoul National University, Korea, in 2006, and the M.S. and Ph.D. degrees Stanford University, USA, in 2008 and 2013, respectively, all in electrical engineering.

She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Seoul National University (SNU), South Korea. Prior to that, she was a Senior Research Scientist with the Center for Imaging Media Research, Korea Institute of Science and Technology (KIST), Seoul, South Korea. Her research interests include computer vision and computer graphics, especially in the area involving 3-D information and practical applications.

Jonghyun Choi received the Ph.D. degree in mechanical engineering from the Yonsei University, Seoul, South Korea, in 2007.

He is currently the President of AONEware, developing computer-aided design automation and real-time visualization. His research interests include real-time rendering in VR/AR, and human–computer interface, robotics, and mechanism design.

Soonhung Han received the B.S. and M.S. degrees in naval architecture and marine engineering from the Seoul National University, South Korea, in 1977 and 1979, respectively, the M.S. degree in naval architecture and shipbuilding from the University of Newcastle upon Tyne, U.K., in 1985, and the Ph.D. degree in naval architecture and marine engineering from the University of Michigan, MI, USA, in 1990.

He is a Professor with the Department of Mechanical Engineering, KAIST, South Korea. He is leading the Intelligent CAD Laboratory, KAIST and the STEP community of Korea. His research interests include STEP, geometric modeling kernel, and VR application in design.

Byounghyun Yoo received his B.S. degree from the Yonsei University, Seoul, South Korea, in 1997, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), in 1999 and 2006, respectively, all in mechanical engineering.

He is currently a Senior Research Scientist with the Center for Imaging Media Research, Korea Institute of Science and Technology (KIST), Seoul. Prior to that, he was a Research Scientist with the Center for Environmental Sensing and Modeling (CENSAM), Singapore-MIT Alliance for Research and Technology (SMART), Singapore, and a Research Fellow at Naval Postgraduate School, USA. His research interests include 3-D Web, modeling and simulation, virtual reality, human–computer interaction, and social manufacturing.