Computational Visual Media
https://doi.org/10.1007/s41095-022-0296-2 Vol. 9, No. x, month year, xx–xx
Research Article

Bin-scanning: Segmentation of X-Ray CT Volume of Binned Parts using Morse Skeleton Graph of Distance Transform

Yuta Yamauchi, Tatsuya Yatagawa, Yutaka Ohtake, and Hiromasa Suzuki(✉)

© The Author(s)

Abstract Industrial X-ray CT scanners have enabled nondestructive evaluation of industrial products, including even inside of their bodies, due to the transmissive nature of X-rays. In light of the effectiveness, this paper introduces a new approach to accelerate the inspection of many same mechanical parts in a bin by X-ray CT scanning. The input to this problem is a volumetric image (i.e., CT volume) of many parts, which is obtained by a single CT scan. We need to segment the parts in the volume to inspect each of them, but random postures and dense contacts of the parts prohibit the part segmentation using a traditional template matching. To address this problem, we convert both the scanned volumetric images of the template and the binned parts to simpler graph structures and solve a graph matching problem to segment the parts. We perform the distance transform to convert the CT volume to a distance field. Then, a graph is constructed based on the Morse theory, where its nodes are located at extremum points of the distance field. The experimental evaluation demonstrates that our fully automatic approach can detect the parts even for a heap of 50 parts in CT volumes. Moreover, the overall computation can be done in approximately 30 minutes for a large CT volume with about $2000 \times 2000 \times 1000$ voxels.

Keywords X-ray computed tomography, Volume segmentation, Graph matching, Nondestructive inspection

1 Introduction

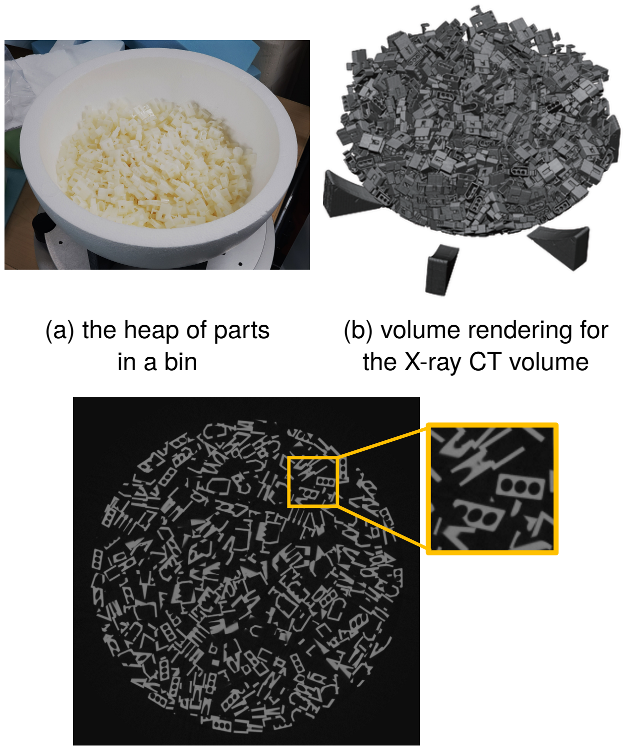
X-ray computed tomography (CT) is an effective tool for nondestructive inspection of industrial products, which often consist of a considerable number of parts. The X-ray CT scan can detect defects in manufactured parts, such as cracks, cavities, and inclusions, owing to its capability of imaging even inside of objects in a nondestructive manner.

As X-ray CT devices have prevailed, the demand for inspecting many mass-produced parts in their production process has arisen. However, current off-the-shelf X-ray CT scanners often take time and monetary costs for scanning a large number of products in a production line. In contrast, X-ray CT scanners can visualize the objects entirely in their field of view. A possible solution for the above problem is to scan many parts together in a single CT scan. Indeed, the same parts manufactured by injection molding are arranged in a sorting tray and scanned by the X-ray CT. However, preparing a tray and arranging the parts is still costly and cumbersome.

Scanning a heap of identical parts stored randomly in a bin (see Fig. 1(a)) will be a more efficient solution. We refer to this process as *bin-scanning* in this paper. As shown in Fig. 1(b) and (c), the bin-scanning obtains an X-ray CT volume that contains all the parts in the bin by a single CT scan. Then, the part regions in the CT volume are segmented for inspecting the parts individually. This procedure is a template matching problem in the sense that the segmented regions correspond to approximately identical objects. Traditional template matching has used the chamfer distance [1] and zero-mean normalized cross-correlation (ZNCC) [2] to measure the similarity to a template. However, these approaches for image-level template matching inherently suffer from the huge search space of its posture, although a large number of studies [3] have been conducted to speed up the exploration of the template.

Matching two sets of sparse keypoints has been a potential approach to fast template matching. Typically, a geometric feature around each keypoint is computed in advance, and these features are employed as a guide for template matching. In the case that each keypoint is associated with a spatial position, we can solve the matching problem using well-studied rigid and non-rigid registration algorithms [4–6].

Address: School of Engineering, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan.
E-mail: {yamauchi,tatsy,ohtake,suzuki}@den.t.u-tokyo.ac.jp.
Manuscript received: 2022-xx-xx; accepted: 2022-xx-xx



(a) the heap of parts in a bin

(b) volume rendering for the X-ray CT volume

(c) a slice of volume and the zoom up for a part

Fig. 1 Our paper discusses *bin-scanning*, which refers to the process of scanning X-ray CT images of randomly stored identical parts in a bin, as shown in (a). The volumetric image of a CT volume is shown in (b), and a slice is shown in (c).

Unfortunately, such approaches that only rely on the spacial arrangement of points might cause unpredictable mismatching when some features of the keypoints are similar. When a set of keypoints also has a graph structure, the template matching can be formulated as a subgraph matching problem. Early studies have considered a problem in finding subgraph isomorphism (i.e., the subgraph with exactly the same graph structure) [7, 8]. Since matching subgraphs with inexactly the same structures is an NP-hard combinatorial optimization problem, other approaches proposed later have solved approximated and relaxed graph matching problems [9, 10]. Recently, the graph and subgraph matching problem can be solved even using deep learning, where several studies [11–13] have reported the high potential of deep learning techniques for more robust template matching. Unfortunately, the subgraph matching has rarely ever been applied to the problems in 3D objects because the graph structure will be intractably complicated, e.g., when the graphs are constructed by connecting vertices sampled on object surfaces.

In this study, we solve the bin-scanning problem using a subgraph matching algorithm because it would be more efficient than simply matching CT sub-volumes using traditional similarity measures. The problem here is that the inputs of bin-scanning are not graphs but CT volumes of a template part and a heap of parts. To focus more on the global structure of the parts, we leverage the Morse theory [14], which is a mathematical tool to analyze the topological property of a manifold, to convert a CT volume into a graph structure. In bin-scanning, we can assume that graphs obtained for a template part and each of them included in the heap will be isomorphic because the shape of the template part is almost the same as those in the heap. Therefore, we can apply a simple matching algorithm for subgraph isomorphism [8] to match the graphs of template and heap.

Figure 2 illustrates the proposed method, which is roughly separated into two stages. The input for our system is two volumes of a template part and heaped parts. For brevity, we refer to them as the template volume and target volume, respectively. We convert each of the volumes into a graph, which we refer to as a Morse skeleton graph (MSG) in this paper. The MSG is constructed based on the Morse theory, and the nodes of the graph are located on the skeleton of the object. Then, a subgraph matching is performed to search the template in the target volume, where both the segmentation and localization of the template parts are achieved simultaneously.

Our method allows the parts in a heap to take random postures. Therefore, the sorting tray and manual parts arrangement are no longer necessary. Furthermore, once we finish scanning a target heap of parts, we can scan another heap of parts while processing the CT volume obtained by a prior scan. Compared to the standard scanning time, the computation time needed by our system is sufficiently short. Therefore, the bin-scanning with our system can significantly accelerate the in-line parts inspection process.

1.1 Problem statement

We assume our template matching problem is defined on the 3D Euclidean space $\mathbb{R}^3$. We denote a position with *bolded* letters (e.g., $\mathbf{x} \in \mathbb{R}^3$), a rigid transformation $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ also with *bolded* letters (e.g., $\mathbf{T} \in SO^3$), and a region (i.e., a 3D subspace of $\mathbb{R}^3$) by a *calface* letters (e.g., $\mathcal{P} \subset \mathbb{R}^3$).

The X-ray CT scan generates a volumetric image (i.e., CT volume) consisting of voxels with CT values which are roughly proportional to the physical density of the parts. When the parts are made of a single material, voxels of the parts have approximately the same values as shown in Fig. 1(c). Therefore, we can isolate the voxels of the parts by binarizing both the template and target volumes. More formally, we obtain a template part $\mathcal{P}_0$ and a target heap of

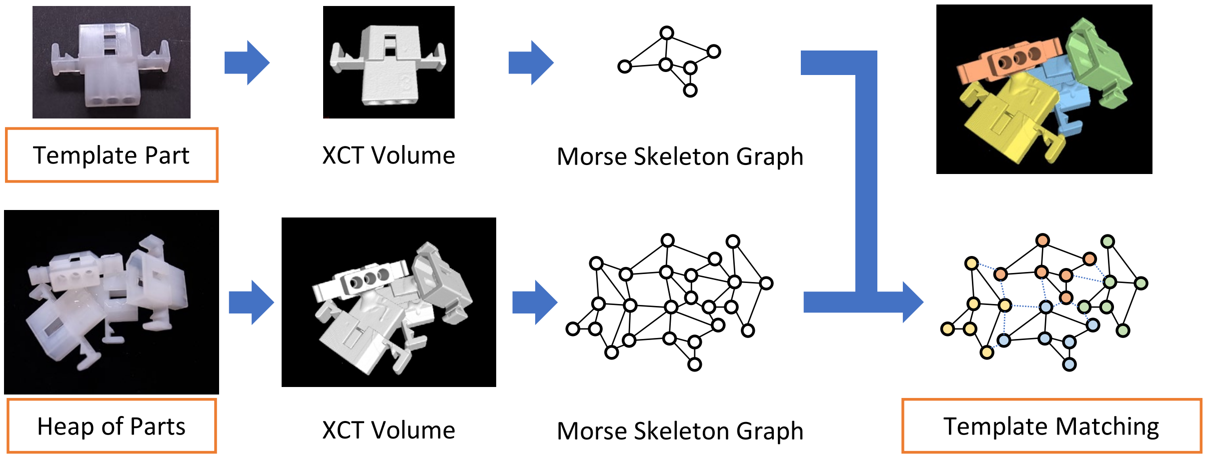


Fig. 2 An overview of our graph-based template matching system for bin-scanning. Our system processes two input volumes of a template part and its heap obtained by X-ray CT scans. The system identifies each part from the volume of heaped parts and calculates its posture as a rigid transformation matrix. To this end, each volume (i.e., those of a template part and its heap) is converted to a graph. We refer to the graph as a Morse Skeleton graph. Using a graph matching technique, we match the graph for a template with subgraphs in the entire graph for heaped parts. Note that the graph structures in this figure are for illustration only.

N parts $\mathcal{H} = \bigcup_{i=1}^{N} \mathcal{P}_i$ by the X-ray CT scan, whereas the number of parts N is unknown. The posture of each part $\mathcal{P}_i$ (i.e., its position and orientation) is represented by a rigid transformation $\mathbf{T}_i$, where

$$\mathcal{P}_i = \mathbf{T}_i \mathcal{P}_0 := \{\mathbf{T}_i(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_0\}. \tag{1}$$

The goal of our problem is to find $\mathbf{T}_1, \ldots, \mathbf{T}_N$ with given $\mathcal{P}_0$ and $\mathcal{H}$ that involves the heap of the parts $\{\mathcal{P}_i\}_{i=1}^{N}$. More precisely, $\mathcal{P}_i$ are closed subsets of $\mathbb{R}^3$, and a pair of parts do not intersect (i.e., $\forall i, j\ :\ \operatorname{Int}\mathcal{P}_i \cap \operatorname{Int}\mathcal{P}_j\ =\ \phi$, where $\operatorname{Int}\mathcal{P}$ denotes the interior of $\mathcal{P}$) because each part is the solid objects. In contrast, the edges and corners of two parts may be shared (i.e., $\exists i, j : \partial\mathcal{P}_i \cap \partial\mathcal{P}_j \neq \phi$, where $\partial\mathcal{P}$ denotes the boundary of $\mathcal{P}$).

Our method converts both template volume $\mathcal{P}_0$ and target volume $\mathcal{H}$ to lightweight graph structures to accelerate the comparison of two volumes (see Fig. 2). The conversion is based on the Morse theory and persistent homology, which allow us for flexible control of the number of nodes of a graph. Then, we match the graph structures between the template and target using an efficient subgraph matching algorithm [8]. In this way, the whole template matching process can be finished in approximately 30 minutes, even for a large CT volume (e.g., the one with 2000 × 2000 × 1000 voxels).

2 Related Work

The template matching for bin-scanning needs to solve two different problems simultaneously, i.e., segmentation to detect each part in a heap and localization to determine the positions and postures of the parts. This section introduces the related work on these topics, including recent studies based on deep learning. Since the Morse theory is leveraged in our template matching, we also introduce its applications to several geometric modeling tasks.

2.1 Image and volume segmentation

The most straightforward approach for segmenting specific regions in a grayscale image is binarization, which often relies on conventional thresholding [15–17] and combined with mathematical morphology, such as erosion, to detach the image domain into small connected regions [18]. Pixel clustering, such as the mean shift method [19], has been employed for segmentation as well [20, 21], although its high computational cost hinders its application to large volume data. The active contour models, such as Snakes [22] and Level Sets [23], are another popular choice for 3D CT volumes [24, 25], which optimize the contour of an object using edges extracted from the gray voxel values. However, it is difficult to obtain the contours of objects touching one another using the active contours. Graph cuts [26–28] can find a globally optimal segmentation result, thus avoiding the fragmentation of object regions due to noise. Unfortunately, its high computational cost makes processing large 3D volumes impractical. The watershed method [21, 29, 30] is also a common approach for segmentation, often used with the distance transform when applied to binary images. The main drawback is over-segmentation, which causes failure in the subsequent template matching process.

In contrast to these approaches above, our method is based on the distance transform and employs a similar approach to the watershed method, while it has two noticeable differences. First, we control the *persistence* of the watershed, based on the Morse theory, to prevent over-segmentation. Second, our method applies the subgraph matching to a graph built upon the results of the watershed method containing over-segmented regions.

2.2 Template matching based on geometric features

Template matching is often based on image and geometric features, simultaneously solving the segmentation and localization problems. Based on the vicinity of features, we can define the correspondences of elements, such as pixels, voxels, and 3D points, and match a template to other data. For volumetric images, feature vectors, e.g., the 3D scale-invariant feature transform (SIFT) [31], are computed by extending the feature for 2D image [32]. However, when applied to large 3D CT volumes, these approaches can suffer from high computational costs in exploring the correspondences of too many feature points, even accelerated by random sample consensus (RANSAC). In contrast, several recent approaches [33–35] have achieved efficient template matching of 3D sparse point clouds, which are obtained by hand-held optical scanners and LiDARs, based on Point Pair Features (PPF) [36] and normal vectors of the points. These approaches could be applied to our target (i.e., CT volume of the identical parts in a bin) by isosurface extraction [37] and point sampling on the surface. However, the mean recall of the state-of-the-art method [35] is still insufficient for industrial part inspection, where the detection requires almost 100 % accuracy.

2.3 Segmentation using deep neural networks

Needless to emphasize here, deep learning is one of the most standard approaches for image and volume segmentation, while various network architectures, such as fully-convolutional network [38], U-Net [39, 40], feature pyramidal network [41] have been proposed. However, due to rapid improvement in image resolution of X-ray CT devices, some models have already been equipped with 8K detectors. Thus, the large memory consumption of 3D convolutional neural networks (CNNs) hinders the straightforward application of the above networks to large CT volumes. The application of deep learning techniques is common in medical-purpose X-ray CT that aims to reduce dose, namely, the patients' X-ray exposure. A previous method reduced the computational complexity by applying 2D CNNs to images obtained by slicing the target volume along three axes [42], while some others concentrate the computational resource only on a specific part of the volume [43, 44], and use a region-specific priors [45, 46]. For industrial X-ray CT, a recent study [47] applied the flood filling network [48] to segment a large X-ray CT volume with 10000^3 voxels. Unfortunately, all these approaches aim to extract the regions belonging to a specific object category (e.g., lung) for a medical purpose and not to identify the regions for different substances of the same object. Furthermore, preparing a large training dataset for our purpose (i.e., segmenting heaped parts in a bin) will be extremely cumbersome.

2.4 Morse theory for geometric modeling

Morse theory is a powerful tool for topology analysis, allowing for extracting important geometric features. Forman [49] introduced the discretized version of the Morse theorem and its related algorithms, and Edelsbrunner et al. [50] followed him to show its application on linear 2-manifolds (e.g., a triangular mesh). As well as the geometry processing on triangular meshes [51], the Morse theory has also been applied to image processing [52, 53], visualization of cosmic objects [54], molecular analysis [55], and mesh quadrangulation [56, 57]. For further applications, we refer the readers to comprehensive surveys [58, 59].

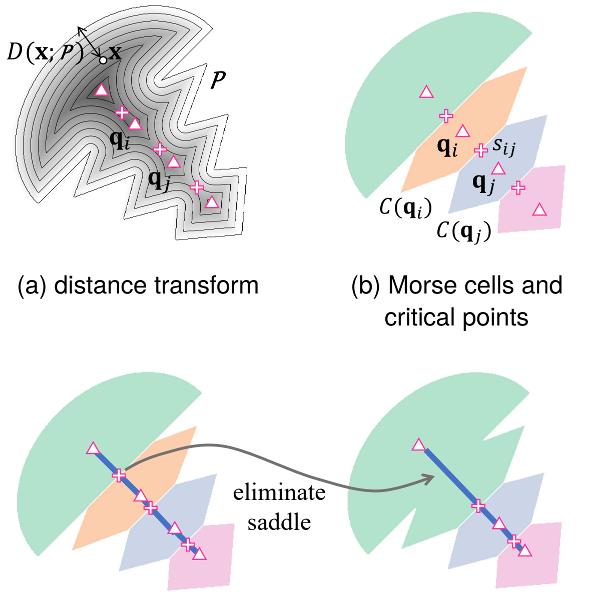
A segmentation method similar to that shown in this paper has been introduced by Nagai et al. [60]. Their method has introduced semantic segmentation based on the Morse theory and processed 3D X-ray CT volumes of industrial assemblies. However, segmentation of heaped parts in an X-ray CT volume is still a challenging problem because their method requires careful human intervention to achieve a good segmentation result, even for an assembly of a few parts.

3 Graph construction from a volumetric image

Our method based on subgraph matching starts with converting an input 3D CT volume into a graph structure using the Morse theory. Before we elaborate on our approach, this section introduces several background theories and how the graph structure is obtained with the volume. As an overview, we show the workflow in Fig. 3.

3.1 Distance transform and its properties

We can assume a 3D part is represented by a connected region $\mathcal{P} \subset \mathbb{R}^3$. The distance transform is a process to define a scalar field based on a metric in $\mathbb{R}^3$. For a part $\mathcal{P}$ (i.e., a connected



(a) distance transform

(b) Morse cells and critical points

(c) Pair cancellation to merge cells by saddle elimination

Fig. 3 Generation of Morse complex and Morse skeleton graph (MSG) for a simple screw shape. (a) We assume the input part $\mathcal{P}$ is a connected region in the Euclidean space and perform the distance transform to obtain a distance field, illustrated with isocurves in this figure. The saddles and local maxima of the distance field are indicated by plus "$+$" and triangle "$\triangle$" symbols. (b) Each cell of a Morse complex corresponds to a stable manifold centered by a local maximum, which is connected with another cell across a saddle. The MSG is a graph structure defined by the cells. (c) The complex, as well as the MSG, can be simplified based on persistence homology by consecutively removing a saddle and merging cells.

region), we can write the distance transform $D : \mathbb{R}^3 \to \mathbb{R}$ by introducing a simple Euclidean norm $\|\cdot\|$ as its metric.

$$D(\mathbf{x};\mathcal{P}) = \begin{cases} \min\limits_{\mathbf{y}\in\partial\mathcal{P}} \|\mathbf{x}-\mathbf{y}\| & \text{for} \quad \mathbf{x}\in\mathcal{P}, \\ 0 & \text{for} \quad \mathbf{x}\notin\mathcal{P}. \end{cases}$$

At a point $\mathbf{x}\in\mathcal{P}$, its value simply takes a minimum distance to the boundary $\partial\mathcal{P}$ and is zero for all $\mathbf{x}\notin\mathcal{P}$. Figure 3(a) shows a simple example of the distance transform. The distance transform for a region defined on a discrete 3D volume can be calculated by solving an eikonal equation $\nabla D(\mathbf{x}) = 1/f(\mathbf{x})$, which is efficiently solved by the fast marching method [61] with the computational complexity ordered by $\mathcal{O}(N\log N)$.

A nice property of the distance transform is its invariance under rigid transformation. Let $\mathbf{T}(\mathbf{x}) = \mathbf{R}\mathbf{x}+\mathbf{t}$ and $\mathbf{T}^{-1}(\mathbf{x}) = \mathbf{R}^{-1}\mathbf{x}-\mathbf{R}^{-1}\mathbf{t}$ be a rigid transformation in $\mathbb{R}^3$ and its inverse, where $\mathbf{R}\in\mathbb{R}^{3\times3}$ is a rotation matrix, and $\mathbf{t}\in\mathbb{R}^3$ is a translation vector. Using the notation $\mathbf{T}\mathcal{P}$ in Eq. (1) for a rigidly transformed region, we can write the invariance of the distance transform as in the following equation:

$$D(\mathbf{x};\mathcal{P}) = D(\mathbf{T}(x);\mathbf{T}\mathcal{P}). \tag{2}$$

Assume here that a number of parts $\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_N$ with the same shape are in the target volume, and $\mathcal{P}_0$ be the part in the template volume. Then, we can write each part $\mathcal{P}_i = \mathbf{T}_i\mathcal{P}_0$ using a rigid transformation $\mathbf{T}_i$. Considering the invariance of the distance transform in Eq. (2), we obtain

$$D(\mathbf{x};\mathcal{P}_i) = D(\mathbf{x};\mathbf{T}_i\mathcal{P}_0) = D(\mathbf{T}_i^{-1}(\mathbf{x});\mathcal{P}_0).$$

Furthermore, a distance transform of the heaped parts $\mathcal{H} = \bigcup_{i=1}^{N}\mathcal{P}_i$ is derived from the property of distance transformation taking zero outside of the region.

$$D(\mathbf{x};\mathcal{H}) = \sum_{i=1}^{N} D(\mathbf{x};\mathcal{P}_i) = \sum_{i=1}^{N} D(\mathbf{x};\mathbf{T}_i\mathcal{P}_0).$$

This property is important for template matching because it means the distance transform of the template part is equivalent when a part with the same shape is in a heap.

3.2 Basic Morse theory

The Morse theory was originally devised for smooth function on manifolds [14], and in computer graphics, this theory is often used to determine a topological structure on 2-manifold [57, 62]. In contrast to these studies, we extract a graph (i.e., the topological structure) on 3-manifold $\mathcal{M}$, which corresponds to $\mathcal{H}$ or $\mathcal{P}_i$ equipped with the simple Euclidean metric. Here, we assume that the 3-manifold $\mathcal{M}$ is parameterized by three coordinates (u_0,u_1,u_2).

Let $f:\mathcal{M}\to\mathbb{R}$ be a real-valued function on the manifold. Then a point is *critical* when its derivative with a coordinate $[\partial f/\partial u_i]$ approaches zero, otherwise it is *regular*. Furthermore, a critical point is *Morse* when the Hessian matrix $[\partial^2 f/\partial u_i\partial u_j]$ at the critical point is non-singular, otherwise it is *degenerate*. If and only if critical points on the manifold are Morse, we refer the function f to a Morse function. Then, we can define an integral line of the Morse function f, which is a maximal path on $\mathcal{M}$ whose gradient direction agrees with the gradient of f. Ascending (or stable) manifold and descending (or unstable) manifold are defined as clusters of integral lines having common origin and destination, respectively. The Morse complex is a partition of $\mathcal{M}$ into descending manifolds. In contrast, the Morse-Smale complex is a partition of $\mathcal{M}$ where each cell is composed of a set of integral lines with the same origin and destination. In our problem, a cell of the Morse complex computed for the distance transform corresponds to a graph node. Specifically, Each node lies on a local maximum of the distance field, while two adjacent nodes lie on opposite sides of a saddle.

3.3 Simplification on Morse complex

An important property of the Morse complex is that the number of cells can be reduced by an operation called *pair cancellation* [58, 63]. The pair cancellation is based on the importance of a pair of critical points, which is often defined by the difference in the values of Morse function values at two critical points. The importance is also known as a *persistence*, and the persistence p of two critical points u and v is formally defined as $p = |f(u) - f(v)|$ where f is a Morse function.

In a 3D space, saddles of a function can be classified into 1-saddle and 2-saddle, depending on how ascending and descending directions distribute around the saddle. The cancellation on a 3D Morse-Smale complex can only be defined for a pair of local minimum and 1-saddle and that of local maximum and 2-saddle, eliminating both the critical points in the pair and revising the topology (or a gradient field) to redefine the graph structure. Informally, on a 3D Morse complex (not Morse-Smale), the cancellation between a local minimum and a 1-saddle corresponds to the elimination of edge, while that between a 2-saddle and a local maximum corresponds to the elimination of a face [63]. In this way, we can control the complexity of the Morse complex by increasing and decreasing the minimum persistence allowed. We simplify the graph structure defined by the Morse cells and their adjacencies, as shown in Fig. 3(c).

4 Part Segmentation by Graph Matching

To reduce the computational complexity of the template matching problem between target and template volumes, we solve the problem using a computationally simpler subgraph matching algorithm. We represent each of the template and target volumes as a graph with a few nodes and edges using the pair cancellation described above. Once the graph structures are obtained, we search subgraphs of each individual part in the target heap using a graph matching algorithm.

4.1 Morse skeleton graph

To define a graph structure on an input CT volume, we first extract an object region Ω and its boundary isosurface $\partial\Omega$ using the marching cubes method [37]. We assume that Ω is a close 3-manifold with a boundary. Second, we compute the distance transform $D(\mathbf{x}; \Omega)$ from the boundary $\partial\Omega$ by the fast marching method [61]. Strictly, the distance field on a discrete grid (e.g., pixels in an image and voxels in a volume) is not Morse because it is not C^2-continuous at its local maximum. On the other hand, when we assume this distance field is a discrete height map, we can determine critical points by checking whether neighboring voxels have a smaller or larger function value than the voxel of interest.

In our problem, the critical points appear only on either local maximum or 2-saddle due to the non-decreasing property of the distance transform. To partition Ω into descending manifolds, we trace an integral line starting from every voxel $\mathbf{x} \in \Omega$ until it reaches a local maximum, obtaining a Morse complex for Ω as shown in Fig. 3(b). We denote each cell of the complex as $\mathcal{C}(\mathbf{q}_i) \subset \mathcal{M}$ where $\mathbf{q}_1, \ldots, \mathbf{q}_M$ are M local maximum points of $D(\mathbf{x}; \Omega)$. In contrast, 2-saddles exist between two cells. We determine the 2-saddles by searching a voxel over the boundary region of two cells (i.e., $\partial\mathcal{C}(\mathbf{q}_i) \cap \partial\mathcal{C}(\mathbf{q}_j)$) that satisfies

$$\mathbf{s}_{ij} = \operatorname*{argmax}_{\mathbf{s} \in \partial\mathcal{C}(\mathbf{q}_i) \cap \partial\mathcal{C}(\mathbf{q}_j)} D(\mathbf{s}; \Omega).$$

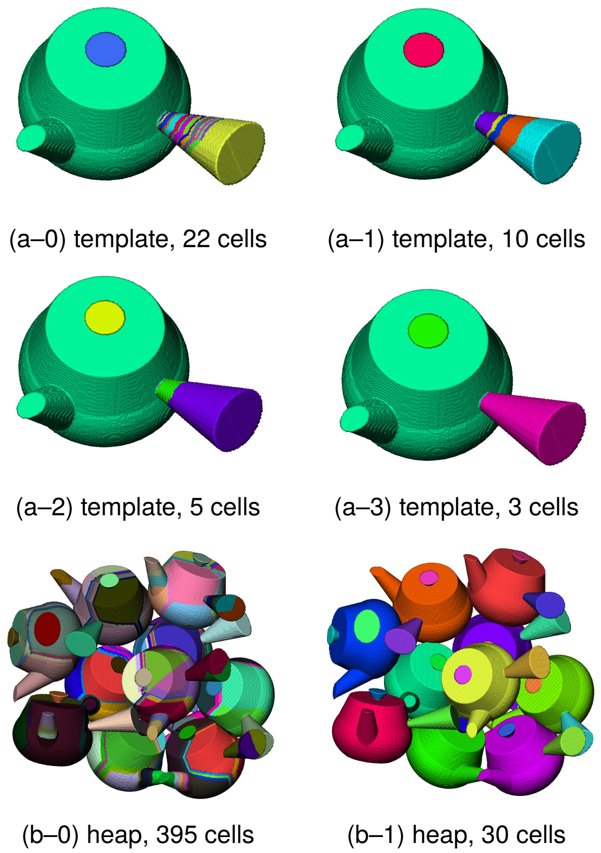
The MSG is defined by the nodes corresponding to the Morse cells and the edges corresponding to the 2-saddles between two cells (see Fig. 3(c), left). As these critical points are located on a medial axis (i.e., a skeleton of Ω), we refer to the graph structure as a Morse *skeleton* graph.

However, the Morse complex computed for an X-ray CT volume can include a huge number of cells because the CT volume involves noise in practice. Therefore, the MSG corresponding to the complex also has a considerable number of nodes. To reduce the computational complexity in subsequent subgraph matching, we simplify the MSG using a pair cancellation described previously. As we discussed in Section 3.3, the cancellation of a pair of 2-saddle and local maximum corresponds to the elimination of a face (i.e., a cell or a portion of voxels in our case). Then, we cancel a series of saddle points in ascending order of the persistence until the minimum persistence of saddles reaches a predefined threshold τ. For the Morse complex computed for a distance field, we employ the persistence defined as follows:

$$p(\mathbf{s}_{ij}) = \min\{D(\mathbf{q}_i; \Omega), D(\mathbf{q}_j; \Omega)\} - D(\mathbf{s}_{ij}; \Omega).$$

Considering the basic property of persistence, elimination of a saddle point and the merge of two local maxima to a single one does not affect the persistence values of other saddles in theory. After a local maximum is removed, the persistence values of the saddles next to the removed local maximum may be updated. The pair cancellation, which is performed for the 2-saddles in the order of decreasing persistence, can be implemented using a priority queue. We hereinafter denote the MSG for an object Ω as $\mathcal{G}_\Omega = \mathcal{G}(\mathcal{V}_\Omega, \mathcal{E}_\Omega)$ where $\mathcal{V}_\Omega$ and $\mathcal{E}_\Omega$ are nodes and edges of the graph.

A simplification of the 3D MSG of a teapot example is shown in Fig. 4. To synthesize the data of heaped parts, we used standard rigid body simulation to determine their


(a–0) template, 22 cells

(a–1) template, 10 cells

(a–2) template, 5 cells

(a–3) template, 3 cells

(b–0) heap, 395 cells

(b–1) heap, 30 cells

Fig. 4 Simplification of the Morse skeleton graph. The initial number of 22 cells for the template volume in (a–0) is decreased gradually to 3 by increasing the threshold τ, as shown in (a–1), (a–2), and (a–3). Equivalently, the initial number of 395 cells shown in (b–0) for the volume of heaped parts is decreased to 30 cells in (b–1) by increasing τ.

postures. Then, CT simulation obtained the volumes for a single template part and heaped parts. The size of the volume for both the data was 800 × 800 × 600 voxels. Figure 4(a–0) shows the initial MSG of the template volume, including 22 cells. While the persistence threshold τ is increased, MSG is simplified to consist of fewer cells, as shown in Figs. 4(a–1), 4(a–2), and 4(a–3). The simplification of the MSG is also performed for the volume of heaped parts, decreasing 395 cells in the original MSG to only 30 cells, as shown in Figs. 5(b–0) and 5(b–1). During the simplification, the threshold τ should not be too large to obtain a few cells because at least three cells are required to obtain a rigid transformation matrix between the template and a part in the heap. In contrast, it should not be too small because the small τ obtains many cells. The MSG construction without the simplification is sensitive to noise signal in the original volume, and insufficient simplification makes the MSGs different for the template and heaped parts even if they represent the same parts with different postures. We will discuss later how to determine the threshold τ in Section 5.

4.2 Part separation and localization via subgraph matching

In a graph for a set of heaped parts $\mathcal{G}_{\mathcal{H}}$, the graph for each part (i.e., $\mathcal{G}_{\mathcal{P}_1}, \ldots, \mathcal{G}_{\mathcal{P}_N}$) may be connected to one another, and some edges connect cells belonging to the graphs of different parts (see Fig. 2). Therefore, we employ subgraph matching to detach the graphs of individual parts. In bin-scanning, we can assume that the graph for a template part has the same structure as that of each part included in the heap. Therefore, our method employs a simple algorithm to match subgraph isomorphism [7, 8].

Specifically, our subgraph matching is based on the VF2 algorithm [8] that combines a simple backtracking algorithm by depth-first search [7] with search pruning using node and edge attributes. Using the terminology in subgraph matching, the MSG for the template volume $\mathcal{G}_{\mathcal{P}_0}$ corresponds to a query graph, and that for the target volume $\mathcal{G}_{\mathcal{H}}$ corresponds to a data graph. In the VF2 algorithm, we sequentially match nodes of the query graph to those of the data graph in an order defined by the adjacency of nodes. Since each node of the MSG is associated with a Morse cell, which can be embedded into a 3D space, we employ three criteria for robust node matching. Assume a case where we try to match Mth node of the query graph to another one in the data graph after the nodes $\mathbf{q}_0^i \in \mathcal{V}_{\mathcal{P}_0}$ and $\mathbf{q}^i \in \mathcal{V}_{\mathcal{P}}$ for $i = 1, \ldots, m-1$ have already been matched. Then, we first check the size of cells $\mathcal{C}(\mathbf{q}_0^m)$ and $\mathcal{C}(\mathbf{q}^m)$ are sufficiently similar in size, where $\mathcal{C}(\mathbf{q})$ is the cell associated with $\mathbf{q}$. With a predefined threshold τ_V, this criterion is judged by

$$d_r(|\mathcal{C}(\mathbf{q}_0^m)|, |\mathcal{C}(\mathbf{q}^m)|) < \tau_V,$$

where $d_r(x, y)$ is a relative difference defined as $d_r(x, y) = |x - y| / \min(x, y)$, and $|\mathcal{C}|$ is the size of cell $\mathcal{C}$ (i.e., the number of voxels). Second, we check both $\mathbf{q}_0^m$ and $\mathbf{q}^m$ are at the same distance from the surface position nearest to each of them. This criterion is defined using the distance field and another threshold τ_D.

$$d_r(D(\mathbf{q}; \mathcal{H}), D(\mathbf{q}_0; \mathcal{P}_0)) < \tau_D.$$

Third, we compare the positions, i.e., $\mathbf{q}_0^1, \ldots, \mathbf{q}_0^m$, including the new one, can geometrically matches in shape with $\mathbf{q}^1, \ldots, \mathbf{q}^m$. To this end, we compare the distances from a new node $\mathbf{q}_0^m$ to other nodes $\mathbf{q}_0^1, \ldots, \mathbf{q}_0^{m-1}$ with those from $\mathbf{q}^m$ to $\mathbf{q}^1, \ldots, \mathbf{q}^{m-1}$. The criterion is formally written as

$$d_r\left(\left\|\mathbf{q}_0^i - \mathbf{q}_0^m\right\|, \left\|\mathbf{q}^i - \mathbf{q}^m\right\|\right) < \tau_R, \quad i = 1, \ldots, m-1.$$

By always checking the distance from a new one to the other nodes that have already matched, we can properly

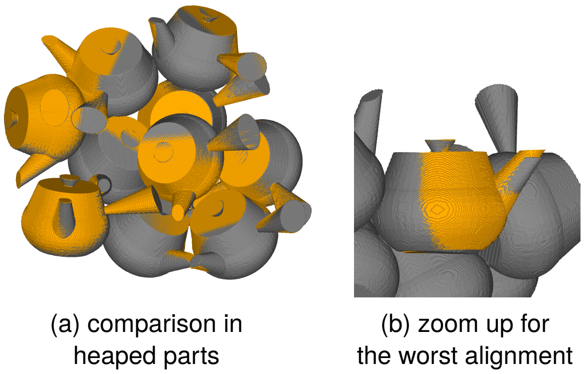

(a) comparison in heaped parts
(b) zoom up for the worst alignment

Fig. 5 Results of rigid transformations by subgraph matching. (a) The teapots transformed by our method (colored by yellow) are overlaid with the ground truth (colored by gray). (b) The position of the largest misalignment.

check the distances of all the pairs of nodes in a graph are sufficiently close to those of the other graph. For the same purpose, we might alternatively use a similar criterion to check the geometric equivalence of two graphs by aligning the graphs. However, it will be time-consuming because a typical solution to obtain a rigid transformation to align two sets of points needs the singular value decomposition (SVD) [64]. In contrast, the third criterion above is significantly faster than performing the SVD every time when a new pair of nodes is matched.

After all nodes in $\mathcal{G}_{\mathcal{P}_0}$ are matched to the nodes in $\mathcal{G}_{\mathcal{H}}$ with a correct topological order, we remove the matched part from $\mathcal{G}_{\mathcal{H}}$ to avoid duplicated matching. This matching procedure runs until no more subgraph remains in $\mathcal{G}_{\mathcal{H}}$ that can be matched to $\mathcal{G}_{\mathcal{P}_0}$.

After subgraph matching is completed, we can calculate a rigid transformation $\mathbf{T}_i$ of the part $\mathcal{P}_i$ in a bin using the positions of the nodes of matched subgraphs. The rigid transformation is obtained as a solution for minimizing the sum of squared distances between the matched graph nodes corresponding to the local maximum points of the distance transform. We can efficiently solve this minimization problem using the SVD [64].

Let us validate the subgraph matching followed by computing rigid transformation matrices. For the previous example with ten teapots, we calculate the rigid transformation matrices and rearrange the template teapot using the computed matrices. The rearranged heap of teapots (colored by yellow) and the input heap of teapots (colored by gray) are compared in Fig. 5(a). As shown in the figure, the postures of the teapots look almost equivalent to those in the original heap, and the amount of misalignment looks small even with the worse one in Fig. 5(b). Since the postures of teapots are calculated by rigid body simulation, the rigid transformation matrices for the input heap are known. After comparing the transformation matrices with those computed by our method, the average misalignment is as small as 3.48 voxels in translation and 1.83° in rotation.

5 Experiments

In order to evaluate the proposed algorithm, we conducted evaluation experiments on the three sets of binned parts shown to the left of Fig. 6, which consist of (a) 50 plastic screws, (b) 10 plastic cases, and (c) 500 plastic connectors, respectively. We implemented our system using C++ and tested on a computer equipped with Intel Core i9-9980XE CPU (3.0 GHz, 18 cores) and 128 GB of RAM. The computation times for these examples on this computer are shown at the bottom of their photos. As shown, overall computations have been finished within approximately 30 minutes, even for a large volume with $2000 \times 2000 \times 1000$ pixels.

5.1 System parameters

Before elaborating on the results, let us discuss how the parameters used in our method affect them. For the three parameters used in subgraph matching, we experimentally set $\tau_V = 0.5$, $\tau_D = 0.5$, and $\tau_R = 1.0$ and used the same values in all the experiments below because these values do not sensitively affect the results. In contrast, τ (i.e., a threshold to determine the minimum persistence of an MSG graph) somewhat affects the results. Figure 7 shows a chart illustrating the relationship between the number of cells in a graph of the template part and the minimum persistence τ, which is computed for the second example in Fig. 6(b). In this chart, the MSG simplification proceeds from right to left, where the minimum persistence monotonically increases from right to left while the number of cells decreases. Interestingly, the change in minimum persistence has a large gap between 0.2 and 0.6, where the number of cells decreases from 7 to 6. Such a large gap appears when the cells are merged overly to have uneven shapes. Therefore, we set the threshold $\tau = 0.4$ in the middle of the gap.

Table 1 shows how many positive matches (i.e., including both true and false positives) and true positive matches are detected with different τ. The numbers in this table are computed for the example in Fig. 6(b) of the plastic case, and the number of parts in a CT volume is ten. Although the choice of τ can affect the detection of parts, $\tau = 0.4$ in the middle of the large gap performs to correctly detect all the parts.

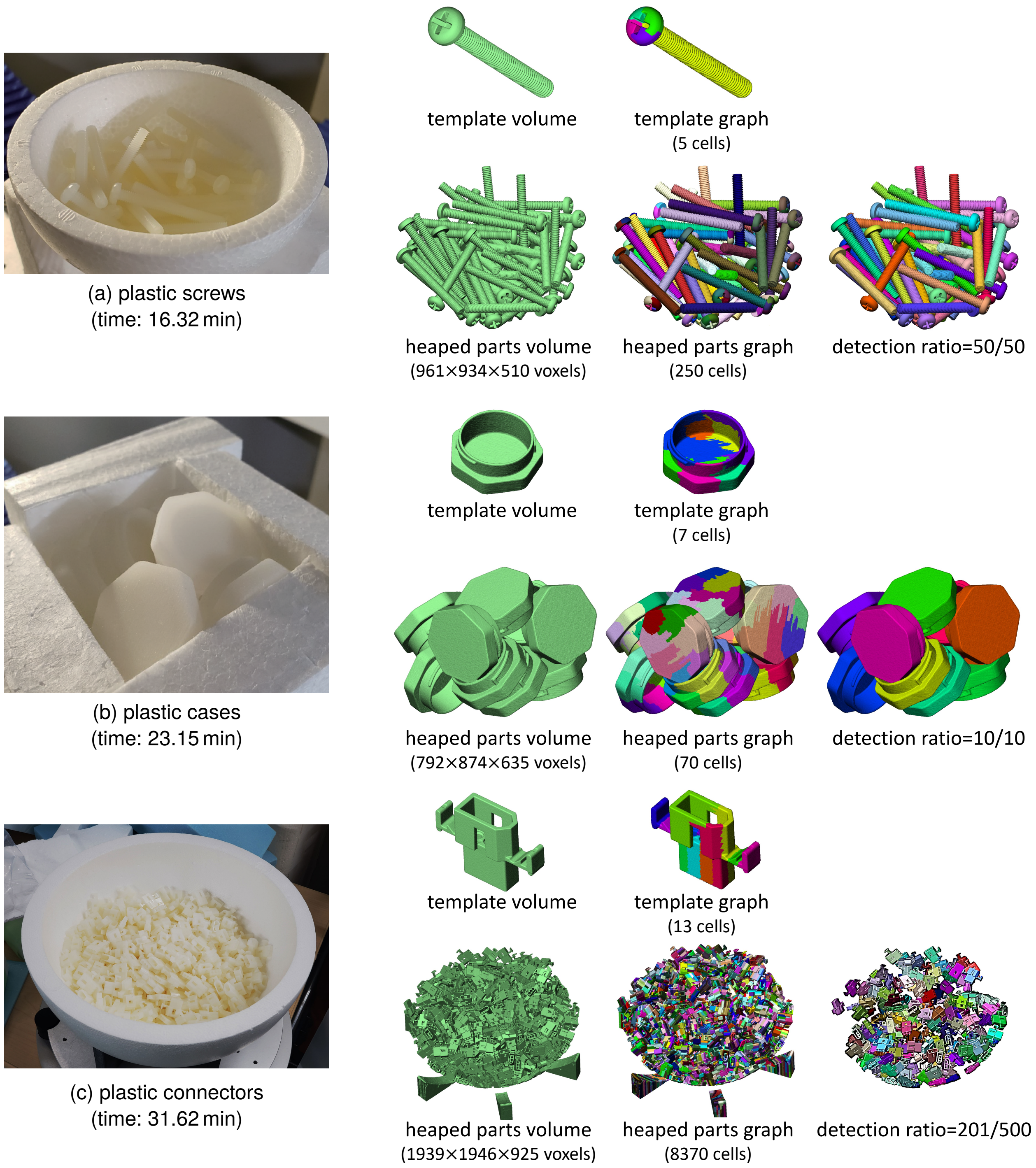


Fig. 6 Experimental results for three different parts, i.e., (a) plastic screws, (b) plastic cases, and (c) plastic connectors. The computation time to process each of them is shown at the bottom of its photo. In the right of this figure, template volume and its corresponding MSG is shown at the top, and heaped parts volume, its corresponding MSG, and detected parts are shown at the bottom. Different cells are colorized in different colors in the images of MSGs, and different parts are colorized with different colors in the images of detected parts.

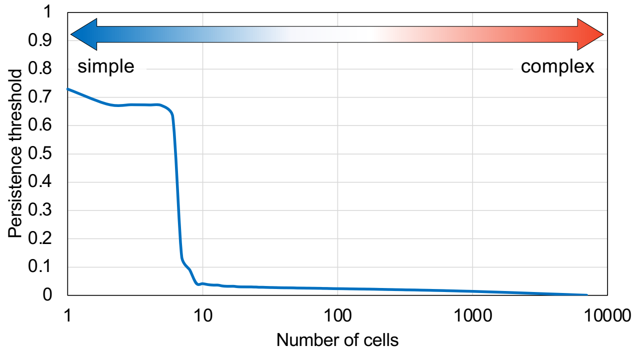


Fig. 7 A line chart of the minimum persistence with respect to the number of cells in the template of an example (b) of plastic cases. The MSG simplification proceeds from right to left, monotonically increasing the minimum persistence while the number of cells decreases. The minimum persistence τ is set to the value in the middle of a large gap, which will be 0.4 in this figure.

Table 1 Change in the detection ratio when using different thresholding parameter τ to control the number of cells. The numbers shown in this table are computed with the example (b) plastic cases in Fig. 6, which consists of 10 cases in total.

τ	#cells of template	#detected matches	#correct matches
0.6735	4	4/10	2/10
0.671	5	3/10	0/10
0.65	6	1/10	1/10
0.4	**7**	**10/10**	**10/10**
0.1	8	8/10	5/10
0.05	9	7/10	2/10
0.0415	10	7/10	0/10

5.2 Performance evaluation

Figure 6 shows the experimental results for three example parts. In each figure, the top row shows a template part and its corresponding MSG, and the bottom row shows a heap of parts, its corresponding MSG, and detected parts by our system. The nodes of an MSG are colorized differently in the figures of the MSGs, while the detected parts are colorized differently in the figures of detected parts.

As shown in Fig. 6(a) and 6(b), our system detects all 50 screws and 10 cases correctly despite the rotational symmetry of the parts. Although the shapes of MSG cells in Fig. 6(b) might be rather irregular, the shapes do not significantly affect our subgraph matching that considers only the locations of critical points and the size of the cell.

Furthermore, our system can obtain a rigid transformation matrix for each part. We applied the inverse transformation to each detected screw in Fig. 6(a) and aligned the screws in evenly spaced cuboids. This figure shows that the detection of screws is successful. Specifically, the cells belonging to the same parts are merged as we intended during the simplification of the MSG, and the shapes of detected parts are equivalent to that of the template part. Furthermore, the orientations of the screws are equivalent despite their thin and long shape with a small head.

On the other hand, we also find that the positions of the screws are somewhat misaligned. This is due to the misalignment of positions of local maxima detected with a discrete distance field. The isosurface extracted from an X-ray CT volume can deviate due to various artifacts [65], and the deviation can displace the positions of local maxima. Although the slight misalignments can be overcome by an additional alignment process (e.g., by using ICP), the standard deviations of the gravity centers of screws are only 4 % of its length without such postprocessing.

For a more challenging example in Fig. 6(c), we could detect only 201 of 500 connectors, which contradicts the last results for screws and cases shown above. The reason for the low detection ratio is small cavities, as shown in Fig. 9(a), generated by a fragile manufacturing process of injection molding. Such cavities cause inconsistency of distance fields between the template part and each part in the heap, thus obtaining different extremum points for each of them. This is a limitation of our system currently, and the cavities in the volume data must be removed in advance.

The results in Fig. 6(c) also suggest another limitation of our system. In this data, each connector has small flat sides, and two connectors can contact each other with the flat sides. Such face contact violates our assumption that two parts are contacted with either a point or an edge.

6 Conclusion

This paper introduced an efficient template matching method for bin-scanning, where identical parts stored in a bin and those scanned by X-ray CT are segmented in a fully automatic manner. Our method converts an input CT volume into a graph, which we referred to as the MSG, and leveraged the well-studied subgraph matching algorithm to accelerate the template matching. After the template is matched to the subgraphs in the volume of heaped parts, we can also calculate the rigid transformation from the template to each part. The experimental results showed that our system worked well for a heap of parts with various shapes, such as teapot, screw, and case.

In contrast, the detection ratio can be somewhat lower when a part involves random cavities included during a manufacturing process, and two parts are contacted with their faces. As with many previous methods for template matching [66] and point set registration [4, 67], symmetric

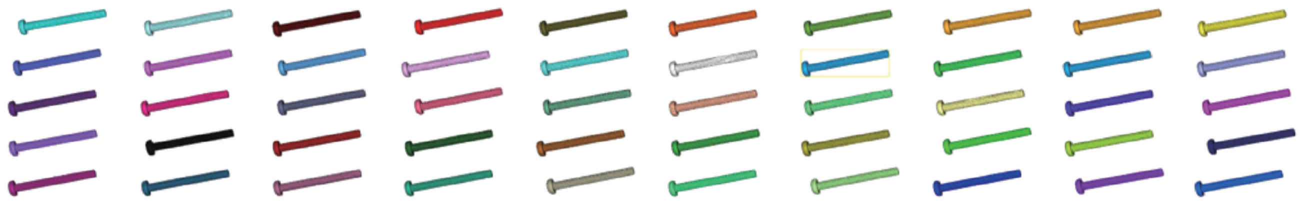
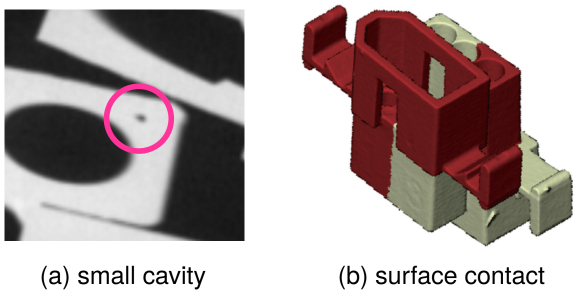


Fig. 8 Individual screws in the example in Fig. 6(a) segmented by our system. The screws are shown by placing them in evenly arranged cuboids by inversely applying the rigid transformation matrices also calculated by our system. Although the positions are slightly misaligned due to the positional displacements of local maximum points on the distance fields, the identification of the parts is succeeded with well-estimated postures of the parts.



(a) small cavity

(b) surface contact

Fig. 9 Limitations of the current graph-based template matching. When (a) small cavities are included in a part, and (b) two parts contact with a face, the distance field of the part will be inconsistent with that of the template. The inconsistency propagates to the graph structure; thus, our system cannot match the parts appropriately in these cases.

shapes of the parts may hinder obtaining a unique rigid transformation. For future work, we would like to alleviate these problems by developing a graph construction insensible to small cavities and a more sophisticated graph matching algorithm applicable to the parts with symmetric shapes and face contacts.

Acknowledgements

We are grateful to Dr. Katsuaki Kawachi for conducting rigid body simulations for the teapots. We would like to thank Dr. Takashi Michikawa, RIKEN, for helpful discussions on the distance transformation.

Declarations

Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.

Funding: No specific funding.

Authors' contributions: Yuta Yamauchi: Algorithms, Methodology, and Software. Tatsuya Yatagawa and Yutaka Ohtake: Theories and Algorithms. Hiromasa Suzuki: Conceptualization and Methodology.

References

[1] Barrow HG, Tenenbaum JM, Bolles RC, Wolf HC. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, SRI International, Menlo Park CA, Artificial Intelligence Center, 1977.

[2] Rosenfield A, Vanderbrug GJ. Coarse-Fine Template Matching. IEEE Transactions on Systems, Man, and Cybernetics, 1977, 7(2): 104–107, doi:10.1109/TSMC.1977.4309663.

[3] Yoo JC, Han TH. Fast normalized cross-correlation. Circuits, systems and signal processing, 2009, 28(6): 819–843, doi:10.1007/s00034-009-9130-7.

[4] Besl P, McKay ND. A method for registration of 3-D shapes. IEEE Transasctions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239–256, doi:10.1109/34.121791.

[5] Rusu RB, Blodow N, Beetz M. Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2009, 3212–3217, doi:10.1109/ROBOT.2009.5152473.

[6] Berg A, Berg T, Malik J. Shape matching and object recognition using low distortion correspondences. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, 2005, 26–33 vol. 1, doi:10.1109/CVPR.2005.320.

[7] Ullmann JR. An algorithm for subgraph isomorphism. Journal of the ACM (JACM), 1976, 23(1): 31–42, doi:10.1145/321921.321925.

[8] Cordella L, Foggia P, Sansone C, Vento M. A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(10): 1367–1372, doi:10.1109/tpami.2004.75.

[9] Zhou F, la Torre FD. Factorized Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(9): 1774–1789, doi:10.1109/tpami.2015.2501802.

[10] Yu T, Wang R. Graph matching with low-rank regularization. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, 1–9, doi:10.1109/WACV.2016.7477730.

[11] Rolínek M, Swoboda P, Zietlow D, Paulus A, Musil V, Martius G. Deep Graph Matching via Blackbox Differen-

tiation of Combinatorial Solvers. In A Vedaldi, H Bischof, T Brox, JM Frahm, editors, Proceedings of the European Conference on Computer Vision (ECCV), 2020, 407–424, doi:10.1007/978-3-030-58604-1_25.

[12] Wang R, Yan J, Yang X. Neural Graph Matching Network: Learning Lawler's Quadratic Assignment Problem with Extension to Hypergraph and Multiple-graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, doi:10.1109/tpami.2021.3078053.

[13] Sarlin PE, DeTone D, Malisiewicz T, Rabinovich A. SuperGlue: Learning Feature Matching With Graph Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[14] Milnor J. Morse Theory. 1963, doi:10.2307/2312441.

[15] Westin CF, Warfield S, Bhalerao A, Mui L, Richolt J, Kikinis R. Tensor Controlled Local Structure Enhancement of CT Images for Bone Segmentation. j-LECT-NOTES-COMP-SCI, 1998, 1496: 1205–1212, doi:10.1007/bfb0056310.

[16] Kang Y, Engelke K, Kalender WA. A New Accurate and Precise 3D Segmentation Method for Skeletal Structures in Volumetric CT Data. IEEE Trans. Med. Imaging, 2003, 22(5): 586–598, doi:10.1109/tmi.2003.812265.

[17] Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62–66, doi:10.1109/tsmc.1979.4310076.

[18] Soille P, Vogt P. Morphological segmentation of binary patterns. Pattern Recognition Letters, 2009, 30(4): 456–459, doi:10.1016/j.patrec.2008.10.015.

[19] Fukunaga K, Hostetler L. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Trans. Inf. Theor., 2006, 21(1): 32–40.

[20] Carreira-Perpinan MA. Fast nonparametric clustering with Gaussian blurring mean-shift. In Proceedings of the 23rd International Conference on Machine Learning, 2006, 153–160, doi:10.1145/1143844.1143864.

[21] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24: 603–619, doi:10.1109/34.1000236.

[22] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. International Journal of Computer Vision, 1988, 1(4): 321–331, doi:10.1007/bf00133570.

[23] Caselles V, Kimmel R, Sapiro G. Geodesic active contours. International journal of computer vision, 1997, 22(1): 61–79.

[24] Pardo X, Carreira M, Mosquera A, Cabello D. A snake for CT image segmentation integrating region and edge information. Image and Vision Computing, 2001, 19(7): 461–475, doi:10.1016/S0262-8856(00)00092-5.

[25] Taud H, Martinez-Angeles R, Parrot J, Hernandez-Escobedo L. Porosity estimation method by X-ray computed tomography. Journal of Petroleum Science and Engineering, 2005, 47(3): 209–217, doi:10.1016/j.petrol.2005.03.009.

[26] Boykov Y, Jolly MP. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In Proceedings of Eighth IEEE International Conference on Computer Vision (ICCV 2001), 2001, 105–112.

[27] Felzenszwalb PF, Huttenlocher DP. Efficient Graph-Based Image Segmentation. Int. J. Comput. Vision, 2004, 59(2): 167–181, doi:10.1023/b:visi.0000022288.19776.77.

[28] Liu L, Raber D, Nopachai D, Commean P, Sinacore D, Prior F, Pless R, Ju T. Interactive Separation of Segmented Bones in CT Volumes Using Graph Cut. In Medical Image Computing and Computer-Assisted Intervention (MICCAI 2008), volume 5241, 2008, 296–304, doi:10.1007/978-3-540-85988-8_36.

[29] Vincent L, Soille P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. IEEE Trans. Pattern Anal. Mach. Intell., 1991, 13(6): 583–598, doi:10.1109/34.87344.

[30] Comaniciu D. Image segmentation using clustering with saddle point detection. In International Conference on Image Processing, 2002, III–297–III–300.

[31] Flitton GT, Breckon TP, Bouallagu NM. Object Recognition using 3D SIFT in Complex CT Volumes. In Proceedings of British Machine Vision Conference, 2010, 1–12, doi:10.5244/c.24.11.

[32] Lowe DG. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 2004, 60(2): 91–110, doi:10.1023/b:visi.0000029664.99615.94.

[33] Papazov C, Burschka D. An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In Asian Conference on Computer Vision, 2010, 135–148, doi:10.1007/978-3-642-19315-6_11.

[34] Drost B, Ulrich M, Navab N, Ilic S. Model globally, match locally: Efficient and robust 3D object recognition. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, 998–1005, doi:10.1109/CVPR.2010.5540108.

[35] Vock R, Dieckmann A, Ochmann S, Klein R. Fast template matching and pose estimation in 3D point clouds. Computers & Graphics, 2019, 79: 36–45, doi:10.1016/j.cag.2018.12.007.

[36] Mian A, Bennamoun M, Owens R. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(10): 1584–1601, doi:10.1109/tpami.2006.213.

[37] Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM Transactions on Graphics, 1987, 21(4): 163–169, doi:10.1145/37402.37422.

[38] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, 3431–3440, doi:10.1109/cvpr.2015.7298965.

[39] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image

computing and computer-assisted intervention, 2015, 234–241, doi:10.1007/978-3-319-24574-4_28.

[40] Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In International conference on medical image computing and computer-assisted intervention, 2016, 424–432, doi:10.1007/978-3-319-46723-8_49.

[41] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, 2117–2125, doi:10.1109/cvpr.2017.106.

[42] Zhou X, Takayama R, Wang S, Hara T, Fujita H. Deep learning of the sectional appearances of 3D CT images for anatomical structure segmentation based on an FCN voting method. Medical Physics, 2017, 44(10): 5221–5233, doi:10.1002/mp.12480.

[43] Zhou Y, Xie L, Shen W, Wang Y, Fishman EK, Yuille AL. A Fixed-Point Model for Pancreas Segmentation in Abdominal CT Scans. In Medical Image Computing and Computer Assisted Intervention (MICCAI 2017), 2017, 693–701, doi:10.1007/978-3-319-66182-7_79.

[44] Li P, Zhou XY, Wang ZY, Yang GZ. Z-net: An anisotropic 3D DCNN for medical CT volume segmentation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, 2906–2913, doi:10.1109/iros45743.2020.9341426.

[45] Dou Q, Yu L, Chen H, Jin Y, Yang X, Qin J, Heng PA. 3D deeply supervised network for automated segmentation of volumetric medical images. Medical image analysis, 2017, 41: 40–54, doi:10.1016/j.media.2017.05.001.

[46] Cui Z, Li C, Wang W. ToothNet: automatic tooth instance segmentation and identification from cone beam CT images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 6368–6377, doi:10.1109/cvpr.2019.00653.

[47] Gruber R, Gerth S, Claußen J, Wörlein N, Uhlmann N, Wittenberg T. Exploring Flood Filling Networks for Instance Segmentation of XXL-Volumetric and Bulk Material CT Data. Journal of Nondestructive Evaluation, 2021, 40(1).

[48] Januszewski M, Maitin-Shepard J, Li P, Kornfeld J, Denk W, Jain V. Flood-filling networks. arXiv preprint arXiv:1611.00421, 2016.

[49] Forman R. Morse Theory for Cell Complexes. Advances in Mathematics, 1998, 134(1): 90–145, doi:10.1006/aima.1997.1650.

[50] Edelsbrunner H, Harer J, Zomorodian A. Hierarchical Morse Complexes for Piecewise Linear 2-manifolds. In Proceedings of the Seventeenth Annual Symposium on Computational Geometry, SCG '01, 2001, 70–79, doi:10.1145/378583.378626.

[51] Bremer PT, Hamann B, Edelsbrunner H, Pascucci V. A topological hierarchy for functions on triangulated surfaces. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(4): 385–396, doi:10.1109/.2004.3.

[52] Paris S, Durand F. A Topological Approach to Hierarchical Segmentation using Mean Shift. In IEEE CVPR, 2007, 1–8, doi:10.1109/cvpr.2007.383228.

[53] Robins V, Wood PJ, Sheppard AP. Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. IEEE Trans. Pattern Anal. Mach. Intell., 2011, 33(8): 1646–1658, doi:10.1109/tpami.2011.95.

[54] Shivshankar N, Pranav P, Natarajan V, van de Weygaert R, Bos EGP, Rieder S. Felix: A Topology based Framework for Visual Exploration of Cosmic Filaments. IEEE Transacsions on Visualization and Computer Graphics, 2015, 22(6): 1745 – 1759, doi:10.1109/tvcg.2015.2452919.

[55] Günther D, Boto RA, Contreras-Garcia J, Piquemal JP, Tierny J. Characterizing Molecular Interactions in Chemical Systems. IEEE Transacsions on Visualization and Computer Graphics, 2014, 20(12): 2476–2485, doi:10.1109/tvcg.2014.2346403.

[56] Dong S, Bremer PT, Garland M, Pascucci V, Hart JC. Spectral surface quadrangulation. In ACM SIGGRAPH 2006 Papers, 2006, 1057–1066, doi:10.1145/1179352.1141993.

[57] Fang X, Bao H, Tong Y, Desbrun M, Huang J. Quadrangulation through Morse-Parameterization Hybridization. ACM Transactions on Graphics, 2018, 37(4), doi:10.1145/3197517.3201354.

[58] De Floriani L, Fugacci U, Iuricich F, Magillo P. Morse Complexes for Shape Segmentation and Homological Analysis: Discrete Models and Algorithms. Comput. Graph. Forum, 2015, 34(2): 761–785, doi:10.1111/cgf.12596.

[59] Heine C, Leitte H, Hlawitschka M, Iuricich F, Floriani LD, Scheuermann G, Hagen H, Garth C. A Survey of Topology-based Methods in Visualization. Computer Graphics Forum, 2016, 35(3): 643–667, doi:10.1111/cgf.12933.

[60] Nagai Y, Ohtake Y, Suzuki H. SegMo: CT volume segmentation using a multi-level Morse complex. Computer-Aided Design, 2019, 107: 23–36, doi:10.1016/j.cad.2018.09.002.

[61] Zhao H. A fast sweeping method for eikonal equations. Mathematics of computation, 2005, 74(250): 603–627.

[62] Ni X, Garland M, Hart JC. Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh. ACM Trans. Graph., 2004, 23(3): 613–622, doi:10.1145/1015706.1015769.

[63] Čomić L, De Floriani L. Cancellation of Critical Points in 2D and 3D Morse and Morse-Smale Complexes. In Discrete Geometry for Computer Imagery, 2008, 117–128, doi:10.1007/978-3-540-79126-3_12.

[64] Umeyama S. Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(4): 376–380, doi:10.1109/34.88573.

[65] Barrett JF, Keat N. Artifacts in CT: Recognition and Avoidance. RadioGraphics, 2004, 24(6): 1679–1691, doi:10.1148/rg.246045065.

[66] Reddy P, Guerrero P, Fisher M, Li W, Mitra NJ. Discovering pattern structure using differentiable compositing. ACM Transactions on Graphics, 2020, 39(6): 1–15, doi:10.1145/3414685.3417830.

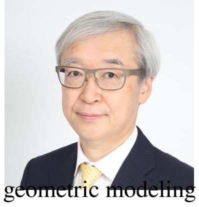
[67] Gelfand N, Ikemoto L, Rusinkiewicz S, Levoy M. Geometrically stable sampling for the ICP algorithm. doi:10.1109/IM.2003.1240258.

Author biography

Yuta Yamauchi is a student in a doctoral program at the Department of Precision Engineering, School of Engineering, the University of Tokyo. He received his master's degree from the University of Tokyo in 2013. He works at Hitachi, Ltd. as a researcher since 2013 and as an engineer since 2021. His research interest includes image/geometry/point-cloud processing and their industrial applications.

Tatsuya Yatagawa is an assistant professor at the Department of Precision Engineering, School of Engineering, the University of Tokyo. He received his Ph.D. degree from the University of Tokyo in 2015. From 2016 to 2019, he worked at the Graduate School of Science and Engineering, Waseda University, as a JSPS (Japan Society for Promotion of Science) postdoctoral research fellow. His research interest includes computer graphics/vision techniques and their applications to computer-aided engineering.

Yutaka Ohtake is an associate professor of Research into Artifacts, Center for Engineering (RACE), School of Engineering, at The University of Tokyo. He received his doctor degree in computer science and engineering from University of Aizu in 2002. His research interests include computer graphics, geometry processing, X-ray CT scanning, and their industrial applications.



Hiromasa Suzuki is a professor of Department of Precision Engineering, School of Engineering, at The University of Tokyo. He received his doctor degree in precision machinery engineering from The University of Tokyo in 1986. His research interests include geometric modeling and its applications to digital engineering. He is a fellow of JSME (Japan Society for Mechanical Engineers) and a fellow of JSPE (Japan Society for Precision Engineers).