Computer-Aided Design 67–68 (2015) 24–37

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Combining volumetric dental CT and optical scan data for teeth modeling☆

Wonhyung Jung [a,c], Seyoun Park [b], Hayong Shin [a,*]

[a] Industrial and Systems Engineering Department, KAIST, Daejeon, South Korea
[b] Radiation Oncology and Molecular Radiation Sciences Department, Johns Hopkins School of Medicine, Baltimore, MD, USA
[c] Cybermed Inc., Seoul, South Korea

HIGHLIGHTS

- A novel teeth modeling framework by combining optical scan data and dental CT images is introduced.
- Co-segmentation between optical scan data and dental CT images is performed by the graph-cut method simultaneously.
- The proposed algorithm is automatic and time efficient, and shows high fidelity.
- Teeth data with defects such as metal artifacts can be completed successfully.

ARTICLE INFO

Article history:
Received 5 March 2014
Accepted 26 April 2015

Keywords:
Teeth modeling
Teeth segmentation
Co-segmentation
Graph-cut
Advancing front method

ABSTRACT

Dental computer-aided design (CAD) systems have been intensively introduced to digital dentistry in recent years. As basic digital models, volumetric computed tomography (CT) images or optical surface scan data are used in most dental fields. In many fields, including orthodontics, complete teeth models are required for the diagnosis, planning and treatment purposes. In this research, we introduce a novel modeling approach combining dental CT images and an optically scanned surface to create complete individual teeth models. First, to classify crown and root regions for each set of data, corresponding pairs between two different data are determined based on their spatial relationship. The pairs are used to define the co-segmentation energy by introducing similarity and dissimilarity terms for each corresponding pair. Efficient global optimization can be performed by formulating a graph-cut problem to find the segmentation result that minimizes the energy. After classifying crown and root regions for each data set, complete individual teeth are obtained by merging the two different data sets. The advancing front method was successfully applied for merging purposes by considering the signed distance from the crown boundary of the surface mesh to the root surface of the CT. The teeth models which have detailed geometries obtained from the optically scanned surface and interstice regions recovered from volumetric data can be obtained using the proposed method. In addition, the suggested merging approach makes it possible to obtain complete teeth models from incomplete CT data with metal artifacts.

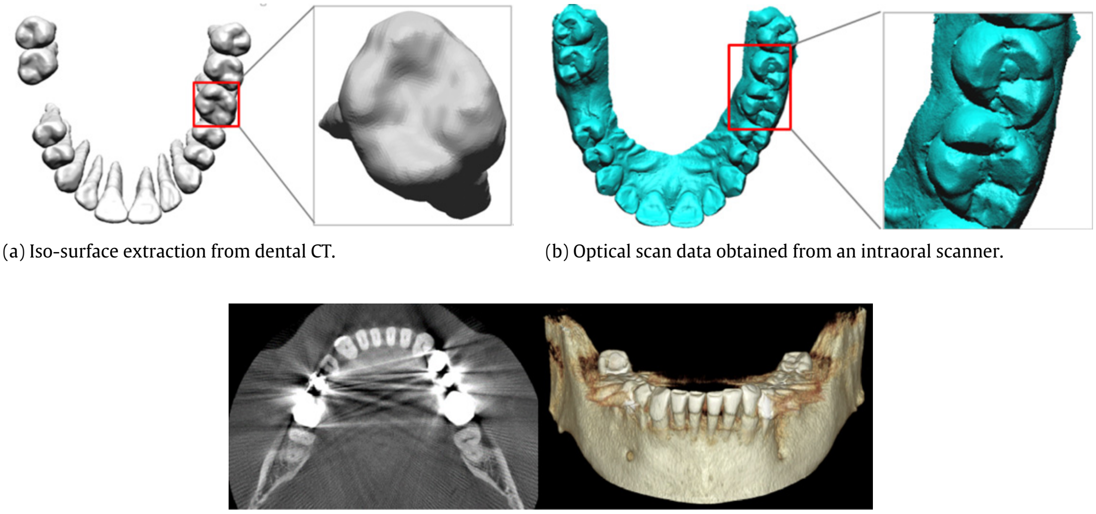
1. Introduction

Detailed digital models of individual teeth are required for many dental applications including orthodontic planning and simulation, dental implants, and jaw surgery. In the case of orthodontics, individual teeth models should be repositioned for diagnosis and planning purposes as well as simple visualization. Digital teeth models can be typically obtained by optical scanning or computed tomography (CT) scanning. One tooth is mainly composed of an exposed crown region and a root covered by gum or a jawbone. For the purpose of observing the morphology of each tooth, dental CT images are the best solution among current imaging modalities.

The importance of individual teeth segmentation has been raised and various approaches have been explored through various works. Gao et al. [1] extracted individual tooth contours explicitly from dental CT images using 2D level sets for each slice. They divided their algorithm into two stages: a single level set and coupled level sets for roots and crown segmentation, respectively. They also suggested a visualization framework of individual teeth

☆ This paper has been recommended for acceptance by Charlie C.L. Wang.
* Corresponding author.
E-mail address: hyshin@kaist.ac.kr (H. Shin).

http://dx.doi.org/10.1016/j.cad.2015.04.008
0010-4485/© 2015 Elsevier Ltd. All rights reserved.



(a) Iso-surface extraction from dental CT.

(b) Optical scan data obtained from an intraoral scanner.

(c) Metal artifacts of CT images (left) and volume rendering (right).

Fig. 1. Comparison of spatial resolutions between CT and scan data (a–b); CT metal artifacts (c).

for orthodontics [2]. Hosntalab et al. [3] applied variational level sets to the 3D volume directly after generating panoramic images in the coronal view. In addition, there are several modeling services used by practitioners in commercial packages. One of the available commercial packages is *Anatomodel* of Anatomage Dental (Anatomage Inc., San Jose, CA, USA). However, these tools in dental services require many manual segmentation operations, and they require much time and human resources. In addition, the previous studies listed above, as well as commercial packages, are based only on dental CT images.

From a modeling point of view, using only CT images has several drawbacks. Dental CT images are, in general, obtained from cone-beam CT (CBCT) scanners with a low dose exposure and a small field-of-view (FOV) compared to general medical CT scanners. The image quality of CBCT scans has many artifacts and defects for various reasons. These artifacts make CBCT images insufficient in representing the detailed geometries, as shown in Fig. 1(a), which are important for dental implants and prostheses. Another critical problem of automatic teeth modeling with CT images is metal artifacts, as shown in Fig. 1(c).

Another major branch of teeth segmentation and modeling has been performed with optical scan data. These data have much higher accuracy (about 0.01 mm) than dental CT images, in general, but only include the exposed area in the mouth without distinction between the teeth and gingival regions, as shown in Fig. 1(b). To classify teeth from the data, several approaches use curvature values as a preprocessing step [4–7]. Because it is not a trivial task to separate teeth using only geometric information, one group used a training set to extract gingival contours by applying 3D statistical models [8]. However, it is not enough to extract only gingival contours in many dental applications because there are missing regions such as the geometries of interstices by occlusion. To resolve this problem, another group attempted to recover the geometries of interstices after removing valley-shaped regions [6]. Recently, Fan et al. [9] introduced an approach to segment individual teeth in the surface mesh using two phases segmentation. The input surface mesh is divided into small patches by low-level segmentation first, and similar neighboring patches are clustered by comparing the shape compactness. Nevertheless, there are still limitations in recovering unscanned areas because the geometries are basically missing.

To resolve this problem, we introduce a novel approach to construct digital teeth models by combining an optically scanned surface and dental CT images. In this approach, we extract crown regions from scanned surfaces with higher accuracy and root parts from dental CT images simultaneously by applying a co-segmentation concept using a graph-cut. The two segmented results are then combined seamlessly by considering the boundary shapes and adjusting different resolutions.

Our main contributions can be summarized as the follows:

- As we know, this is the first work for teeth modeling by combining dental CT images and an optically scanned surface data for the purpose of teeth modeling.
- We developed a co-segmentation method using dental CT images and an optically scanned surface data together for better segmentation results.
- The seamless merging of segmented regions from two different data types was developed to obtain each tooth model.

Co-segmentation schemes have been applied to general bitmap images obtained from digital cameras [10,11] or medical images [12,13]. There has been little study on different data types. Our technical approach for co-segmentation is similar to [10], but spatial irregularities of the surface mesh have led to a quite different formulation. In addition, we applied the classified results to merge them because our final goal was to obtain complete teeth models.

2. Overall procedure

Before describing the technical details of our approach, we will explain the overall procedure first. Fig. 2 shows a conceptual diagram of the overall procedure. From the volumetric CT images $\Omega \subset \mathbb{R}^3$, we extract separate individual teeth $V = V_1 + \cdots + V_n \subset \Omega$ where V_i is a segmented tooth region, $V_i \cap V_j = \phi$ and n is the number of teeth. This procedure is performed in 2 steps: first, the initial teeth region is extracted via optimal global thresholding and region growing with local thresholding. For optimal global thresholding, the Otsu's thresholding method is used to segment hard bone regions, including jaw bones and teeth. In the initial segmentation result, some of the teeth's surfaces are adjacent to jaw bones. To separate the teeth region only, the seed points for each tooth are marked manually and extracted by region growing with local thresholding. Our local thresholding finds connected voxels that are statistically similar to voxels surrounding the user defined seed points (voxels). The neighboring voxels of the boundary of the current active region are selected if their intensity is within [mean ± SD] of the active region, where SD is the standard

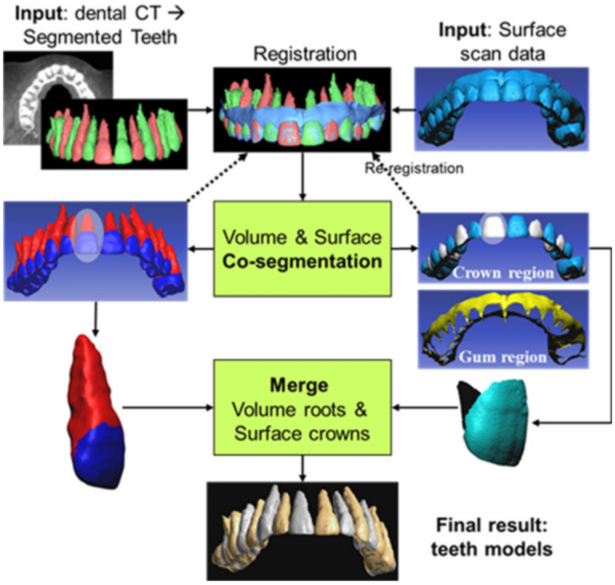

Input: dental CT → Segmented Teeth
Registration
Input: Surface scan data
Re-registration
Volume & Surface Co-segmentation
Crown region
Gum region
Merge Volume roots & Surface crowns
Final result: teeth models

Fig. 2. The overall procedure.

deviation. The statistics are updated each iteration. A single seed point may not be sufficient to select the entire tooth region in some cases. This can be simply compensated by multiple selections.

From the initially extracted teeth region, morphological erosion is then performed until each tooth is separated. The amount of erosion is dependent on the contact surface between teeth. To determine the size of erosion probe, we construct a Euclidean distance field D of the teeth region V from the teeth boundary ∂V as shown in Fig. 3. In the distance field, saddle points $\{\mathbf{p}_i\}$ of D, represented as blue dots in Fig. 3(c), are located on the tooth boundary and the maximum value of $\{D(\mathbf{p}_i)\}$ determines the total amount of erosion. Initial teeth region is eroded as amount of $\max\{D(\mathbf{p}_i)\}$, then each tooth are separated as shown in Fig. 3(d). A cube kernel was used for erosion in our experiments for time efficiency, but a sphere can be also used. We mark the voxels in the separated region as *visited* and the other initial teeth region as *unvisited*. The separated regions are then expanded by searching its *unvisited* neighbor voxels whose distance values are less than or equal to the current voxel. A priority queue is used for the propagation by saving neighbor voxels in reverse order of D, the distance from ∂V. Propagation is performed until all teeth regions fill the initial volume V. Let us denote the boundary of teeth as $\partial V = \partial V_1 + \cdots + \partial V_n$, where ∂V_i is the set of boundary voxels of V_i.

We then align optical scan data S_0 to ∂V described in the volume coordinates by selecting several feature points from both data sets. The feature points used in the initial transformation are matched pairs selected from both data sets. They are manually selected in the same order from the scan data and CT volume in 3D. The points at the end of central incisors and cusps of molars at both ends are used in our experiments. This initial registration can be improved in an automatic manner by using the well-known iterative closest point (ICP) algorithm [14]. From this, the data set is prepared for the main steps.

Let the registered scan surface be $S \subset \mathbb{R}^3$. After obtaining ∂V and S via preprocessing, we then perform the co-segmentation of ∂V and S simultaneously into crown and non-crown regions by finding corresponding pairs. The co-segmentation result may be affected by the initial registration accuracy. To reduce the effect from initial registration, we repeat the registration and co-segmentation steps until the rigid transformation converges. The registration result of S_0 to ∂V may be improved by using only the crown regions obtained from co-segmentation as input for the ICP algorithm. From the co-segmentation results of S, the crown region of each tooth is smoothly merged with the root region from the corresponding teeth boundary among ∂V_i.

By performing this merging step for all the teeth, we can obtain a complete teeth model combining the crown region from S and the root region from ∂V. The details of each step are described in Sections 3–5.

3. Co-segmentation

The main goal of this step is to segment crown parts from two different data sets simultaneously. As described in Section 1, the teeth segmentation from each data set is not a trivial task, and there are limitations due to the acquisition method and the data's characteristics. To solve this problem, we suggest a co-segmentation method to classify crown and non-crown regions from both data sets in this section.

Rother et al. [11] introduced co-segmentation to classify similar objects lying on different backgrounds. They extended a single segmentation problem based on Markov random fields (MRF) by defining corresponding pairs and introducing an additional term to compare the similarities of histograms between them. Co-segmentation has since been applied to many applications, including medical images [12,13], large data user-interactions [15], multi-videos [16], and 3D surface shapes [17]. Global optimization of these approaches is efficiently solved by a graph-cut algorithm. Graph-cut algorithms have been extensively studied, especially in the computer vision area [18]. The concept and various applications can be found in previous studies on images [18–21] and 3D surface shapes [12,22–26], even though we do not describe the details here. Graph-cut algorithms are appropriate for binary classification problems such as foreground segmentation from backgrounds and they give an efficient framework for global energy optimization. Co-segmentation can be modeled under the same framework as single model segmentation.

Our basic framework is conceptually similar to previous studies applied to a pair of images [10,11]. In the image segmentation problem [10], corresponding pairs can be defined as all pairs of the same foreground histogram buckets because the foreground (or object) in the two images are similar. So, similarity values between corresponding pairs can be used to encourage the corresponding pixel of one image to be segmented to the foreground if a pixel of the other image is segmented to the foreground. However, because our problem is to segment crown parts of similar geometric shapes from 3D surfaces and volume data, corresponding pairs can be defined by some geometric properties between them. Furthermore, not only the similarities but also dissimilarities

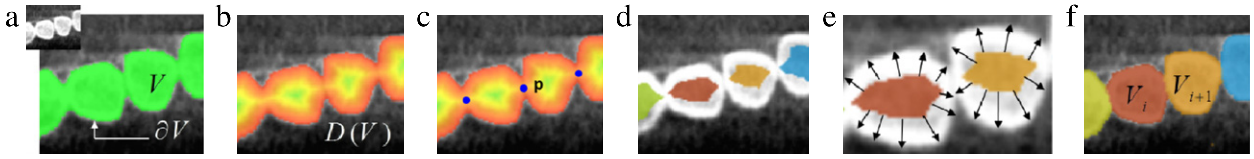
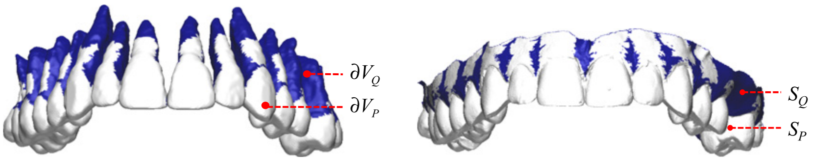


Fig. 3. Teeth separation; (a) initial teeth segmentation; (b) distance field, $D(V)$ generation; (c) saddle point detection; (d) erosion/separation; (e) propagation based on $D(V)$; (f) final separation result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Maxillary teeth from dental CT.

(b) Maxillary teeth optical scan data.

Fig. 4. Corresponding pairs; the white color represents matched regions and the blue denotes the non-matched region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

between corresponding pairs can be helpful for classifying non-crown region (root or gum) to non-crown region because it is difficult to classify non-crown regions correctly near the gingival boundary if only optical scan data are used for segmentation (see Eq. (10), Fig. 7 and Section 5.2). As a result, the details of the optimization equation differ from those of the image segmentation problems and both geometric similarities and dissimilarities are used as the meaningful properties to compare two data sets. In addition, for our purposes, the framework should recover data defects of one data type by compensating mutually from the other data type. The details of our approach are explained in the following sections.

3.1. Corresponding pairs

The first step of our co-segmentation technique is to find the set of corresponding pairs that has a neighboring relationship between two different types of data. S includes interstices and gingival regions, which are not included in ∂V, as well as crowns. This means that there are no corresponding regions in ∂V for gingival or interstice regions. Similarly, there are no corresponding regions in S for the root parts in ∂V. That is, only crown parts, which we want to extract, exist in both sides.

Because two data sets are registered in the preprocessing step, the same parts in both sides are closely located. We can determine which part has its corresponding pair by a simple distance comparison. However, this cannot be determined by a simple threshold value due to different spatial resolutions, artifacts, and registration errors. The main idea to construct corresponding pairs is *conservative rejection*; that is, we exclude the points from the pairs that are not expected to be classified as the crown region. For the region where correct classification cannot be determined only with distances, energy minimization is performed, as described in Section 3.2.

We denote the center position of the paired voxel of $\mathbf{p} \in S$ by $C_S(\mathbf{p}) = \arg\min_{\mathbf{q}_r \in \partial V} \{|\mathbf{p} - \mathbf{q}_r|\} = \mathbf{q}_t$ and the corresponding vertex of $\mathbf{q} \in \partial V$ by $C_V(\mathbf{q}) = \arg\min_{\mathbf{p}_j \in S} \left\{|\mathbf{q} - \mathbf{p}_j|\right\} = \mathbf{p}_k$. Note that $C_V(C_S(\mathbf{p}))$ does not always return $\mathbf{p}$. The ordered index set of corresponding pairs C is then defined as Eq. (1).

$$C = \left\{(j, r) | \mathbf{p}_j \in S, \mathbf{q}_r \in \partial V, \left(\mathbf{q}_r = C_S(\mathbf{p}_j) \text{ or } \mathbf{p}_j = C_V(\mathbf{q}_r)\right), \right. \\ \left. |\mathbf{p}_j - \mathbf{q}_r| < \rho, \left||\mathbf{p}_j - C_S(\mathbf{p}_j)| - |\mathbf{q}_r - C_V(\mathbf{q}_r)|\right| < \varepsilon\right\} \quad (1)$$

where ε, ρ are user-defined constants. As shown in Eq. (1), the corresponding pairs are determined by two criteria: one is the Euclidean distance and the other is the closeness between bidirectional mapping points. We set $\rho = 2.0$ mm, which is large enough considering the initial registration errors, and ε as the diagonal length of a voxel because the second criterion is affected by the resolution of the sparser data.

We denote a set of regions whose vertices are all included in the corresponding pairs defined in Eq. (1) as S_P and voxels as ∂V_P. We then denote the complement of S_P and ∂V_P as $S_Q \subset S$ and $\partial V_Q \subset \partial V$, respectively, and these are the sets of regions that are not matched with any corresponding pairs. Fig. 4 shows each region.

3.2. Co-segmentation model

We define binary random variables that compose the energy equation. There are two kinds of variables for single segmentation of S and V, respectively. The first type is described as $x_j^{(S)}$ and $x_r^{(V)}$ which represent whether the corresponding vertex or voxel belongs to the crown region or not. $x_j^{(S)}$, corresponding to $\mathbf{p}_j \in S$, is defined as Eq. (2). Similarly, $x_r^{(V)}$ for $\mathbf{q}_r \in \partial V$ is defined as Eq. (3).

$$x_j^{(S)} = \begin{cases} 1 & \text{if } \mathbf{p}_j \in S \text{ is classified as crown region} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$x_r^{(V)} = \begin{cases} 1 & \text{if } \mathbf{q}_r \in \partial V \text{ is classified as crown region} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The other type is introduced to compare the similarities between neighboring nodes of each data type. Here, we use a one-ring neighbor, denoted by $N^{(S)}$, for S, and a 6-neighborhood scheme, expressed as $N^{(V)}$, ∂V.

$$y_{j,k}^{(S)} = \begin{cases} 1 & \text{if } x_j^{(S)} \neq x_k^{(S)} \\ 0 & \text{otherwise,} \end{cases} \qquad y_{r,t}^{(V)} = \begin{cases} 1 & \text{if } x_r^{(V)} \neq x_t^{(V)} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Using two types of random variables, the Markov random field formulation for the segmentation of S and V can be generally expressed as Eqs. (5) and (6), respectively [18].

$$\min \sum_{j \in S^I} K_j x_j^{(S)} + \sum_{(j,k) \in N_S^I} B_{j,k}^{(S)} y_{j,k}^{(S)} \quad (5)$$

$$\text{subject to } x_j^{(S)} \in \{0, 1\} \quad \text{for } j \in S^I,\ j = 1, \ldots, m$$

$$y_{j,k}^{(S)} = \begin{cases} 1 & \text{if } x_j^{(S)} \neq x_k^{(S)} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{r \in \partial V^I} K_r x_r^{(V)} + \sum_{(r,t) \in N_V^I} B_{r,t}^{(V)} y_{r,t}^{(V)} \quad (6)$$

$$\text{subject to } x_r^{(V)} \in \{0, 1\} \quad \text{for } r \in \partial V^I,\ r = 1, \ldots, p$$

$$y_{r,t}^{(V)} = \begin{cases} 1 & \text{if } x_r^{(V)} \neq x_t^{(V)} \\ 0 & \text{otherwise} \end{cases}$$

where S^I and ∂V^I represent the index set of vertices in S and voxels of ∂V, respectively. Similarly, N_S^I and N_V^I represent the pairwise index set of the neighbors of S and ∂V, respectively.

The coefficients of boundary terms, $B_{j,k}^{(S)}$ in Eq. (5) and $B_{r,t}^{(V)}$ in Eq. (6), are defined as Eq. (7) in our problem. Note that $B_{j,k}^{(S)}$ is defined by a geometric property, the angle between the normal vectors, because S is a surface mesh. In the case of CT data, it is known that the intensity of the enamel region, which only crowns have, is slightly higher than that of the root bone. We can see this in our formulation of $B_{r,t}^{(V)}$, but it is not meaningful to compute the intensity differences directly because it is hard to detect the exact boundary of the volume data. The intensities are also not consistent. For this reason, we define $B^{(V)}_{r,t}$ to compute the difference between the average intensities by using neighbor voxels within a certain thickness.

$$B^{(S)}_{j,k} = \begin{cases} \dfrac{\tau}{1 - C_{j,k}} & \text{if } |1 - C_{j,k}| > \xi \\ \tau/\xi & \text{otherwise,} \end{cases}$$

$$\text{where } C_{j,k} = \max\left(\mathbf{n}_j \cdot \mathbf{n}_k, 0\right) \tag{7}$$

$$B^{(V)}_{r,t} = \begin{cases} \dfrac{\max\left(\varDelta_{\min}, 1\right)}{|I'_r - I'_t|} & \text{if } |I'_r - I'_t| > \xi, \\ 1/\xi & \text{otherwise} \end{cases}$$

$$\text{where } I'_r = \frac{1}{|N| + 1}\left(I_r + \sum_{t:(r,t)\in N} I_t\right),$$

$$\varDelta_{\min} = \min_{(r,t)\in N^{(V)}} \left|I'_r - I'_t\right|$$

$\mathbf{n}_j$ is the normal vector of the jth vertex in S, τ is a user-defined constant, and I_r is the rth voxel intensity in Ω. ξ is the tolerance to prevent the denominator from becoming zero, and we set $\tau = 10^{-3}$ and $\xi = 10^{-5}$ in our experiments. If ξ is small enough, its value does not affect the segmentation results.

In our co-segmentation problem, it was not useful to determine the probability that one vertex or a voxel belongs to the crown by using its own properties. Therefore, we did not determine the K_r and K_j values for a vertex or a voxel that belonged to the crown by using its own properties in the co-segmentation problem. Instead, we applied hard constraints (using K) to both data terms of the vertices and voxels in S_Q and V_Q in order to classify them as non-crown regions. In this study, the value of K is assigned as:

$$K = 1 + \max\left(\max_{j\in S} \sum_{k:(j,k)\in N^{(S)}} B^{(S)}_{j,k}, \max_{r\in V} \sum_{t:(r,t)\in N^{(V)}} B^{(V)}_{r,t}\right).$$

For the co-segmentation, we can apply a similar concept to the single model segmentation. The two energy equations in Eqs. (5) and (6) can be considered as local regional terms for each data type. The co-segmentation model can be easily extended from Eq. (5) by adding a boundary term between corresponding pairs of S and ∂V [10]. To compare the similarity between corresponding pairs of different data, we define additional binary variables as shown in Eq. (8).

$$m_{j,r} = \begin{cases} 1 & \text{if } x^{(S)}_j = 1 \text{ and } x^{(V)}_r = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

$$n_{j,r} = \begin{cases} 0 & \text{if } x^{(S)}_j = 0 \text{ and } x^{(V)}_r = 0 \\ 1 & \text{otherwise.} \end{cases}$$

Our final co-segmentation model is then composed of two regional terms for each data type, a similarity term and a dissimilarity term, as defined in Eq. (9).

$$\begin{aligned} \min K \sum_{j\in S^I_Q} x^{(S)}_j + \sum_{(j,k)\in N^I_S} B^{(S)}_{j,k} y^{(S)}_{j,k} + K \sum_{r\in \partial V^I_Q} x^{(V)}_r \\ + \sum_{(r,t)\in N^I_V} B^{(V)}_{r,t} y^{(V)}_{r,t} - \sum_{(j,r)\in C} \eta_{j,r} m_{j,r} + \sum_{(j,r)\in C} \delta_{j,r} n_{j,r} \end{aligned} \tag{9}$$

$$\text{subject to } x^{(S)}_j \in \{0,\ 1\} \quad \text{for } j \in S^I,\ j = 1, \ldots, m$$

$$x^{(V)}_r \in \{0,\ 1\} \quad \text{for } r \in \partial V^I,\ r = 1, \ldots, p$$

$$y^{(S)}_{j,k} = \begin{cases} 1 & \text{if } x^{(S)}_j \neq x^{(S)}_k \\ 0 & \text{otherwise} \end{cases}$$

$$y^{(V)}_{r,t} = \begin{cases} 1 & \text{if } x^{(V)}_r \neq x^{(V)}_t \\ 0 & \text{otherwise} \end{cases}$$

$$m_{j,r} = \begin{cases} 1 & \text{if } x^{(S)}_j = 1 \text{ and } x^{(V)}_r = 1 \\ 0 & \text{otherwise,} \end{cases}$$

$$n_{j,r} = \begin{cases} 0 & \text{if } x^{(S)}_j = 0 \text{ and } x^{(V)}_r = 0 \\ 1 & \text{otherwise} \end{cases}$$

where $\eta_{j,r}$ and $\delta_{j,r}$, are the coefficients for similarity and dissimilarity, respectively. They are computed as defined in Eq. (10).

$$\eta_{j,r} = \begin{cases} \exp\left(\dfrac{-\alpha|\mathbf{p}_j - \mathbf{q}_r|}{\bar{d} - |\mathbf{p}_j - \mathbf{q}_r|}\right) \exp\left(\dfrac{\beta\left(|\mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r)| - 1\right)}{|\mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r)|}\right) \\ \quad \text{if } |\mathbf{p}_j - \mathbf{q}_r| < \bar{d} \text{ and } \mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r) \neq 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{10}$$

$$\delta_{j,r} = \begin{cases} \exp\left(\dfrac{\gamma_j|\mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r)|}{|\mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r)| - 1}\right) & \text{if } |\mathbf{n}_j \cdot \nabla_\Omega(\mathbf{q}_r)| < 1 \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{n}_j$ is the normal vector of $\mathbf{p}_j \in S$ and $\nabla_\Omega(\mathbf{q}_r)$ is the gradient vector of $\mathbf{q}_r \in \partial V$.

$\bar{d}$ is the mean value of $|\mathbf{p}_j - \mathbf{q}_r|$ for all corresponding pairs (j, r) in C. $\gamma_j = \kappa^{\max}_j/\kappa_t$ if $(\kappa^{\max}_j > |\kappa^{\min}_j|$ and $\kappa^{\max}_j > \kappa_t)$ otherwise $\gamma_j = 1$. $\kappa^{\max}_j$ is the maximum normal curvature at $\mathbf{p}_j \in S$, $\kappa^{\min}_j$ is the minimum normal curvature, and κ_t is a threshold value. $\kappa_t = 0.5$ was used by experiments in our study. In the case of α and β, they are the parameters to adjust the sensitivity of sub-terms of $\eta_{j,r}$, which are based on the changes of $|\mathbf{p}_j - \mathbf{q}_r|$ and normal vector deviation. We select $\alpha = 4.0$ to make the first sub term almost zero around $\bar{d}/2$, and $\beta = 2.5$ so that the second sub term becomes almost zero when the angular difference between two normal vectors is bigger than 65° from the experiments.

The dissimilarity term is determined by two observations: (1) A negative high curvature area tends to be a part of the boundary between the teeth and gum by the negative minima rule [27]. (2) A positive high curvature area is usually presented in the crown area.

As shown in the $\eta_{j,r}$ of Eq. (10), the coefficient of the similarity term is composed of two subparts. The first consideration is that the spatially close pairs may be considered as the same region. So the left term of Eq. (10) is assigned the larger value of the spatially close pairs as a similarity reward. However, it is not easy to distinguish gingival contours from crowns using only this term because the spatial distances of these regions are close to the root parts as shown in Fig. 5(a). Corresponding pairs in the same crown region may have similar normal vectors. So, we compare geometric similarities simultaneously by comparing the normal vectors of corresponding pairs. The second term of $\eta_{j,r}$ gives the larger value to pairs that have similar normal vectors. As shown in Fig. 5(b), in order to represent values of the second term, most of the corresponding pairs in the gingival contours have values close to zero. Fig. 5(c) of the figure shows the $\eta_{j,r}$ values to multiply two terms. As shown in (c), most of the pairs in the gingival contours have values close to zero because the pairs in this region have larger normal vector deviation, even though the distance is very close. This works fine in that pairs with a positive high curvature in the gum region are segmented easily as non-crown regions. This is because both of the $\delta_{j,r}$ and $\eta_{j,r}$ values are small and their segmentation results are then dependent on the neighbors by $B^{(S)}_{j,k}$ and $B^{(V)}_{r,t}$.

$\eta_{j,r}$ is a *compensation* value for crown pairs when they are close and have similar normal vectors. On the other hand, $\delta_{j,r}$ is the *penalty* term for the cases where one of them is classified as crowns even though the normal vectors are different. By introducing the additional dissimilarity term as well as similarity, we can classify non-crown regions more accurately. The differences are discussed in Section 5.

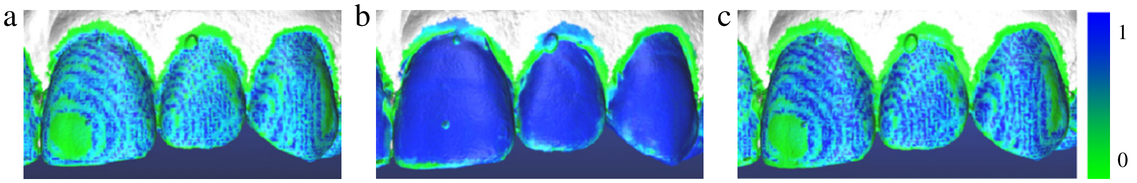


Fig. 5. The color coding of η in Eq. (10); (a) the left term of η; (b) the right term; (c) the final value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

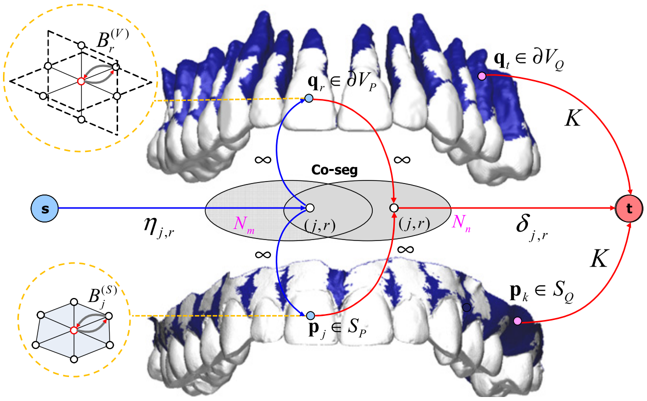


Fig. 6. Conceptual model of the graph construction.

3.3. Graph construction

Our problem can further be converted to a minimum (s, t)-cut problem, the general graph-cut method, similar to the approach introduced in [10]. Let $N_x = N_S \cup N_V$ where N_S is the set of nodes in the graph corresponding to $x_j^{(S)}$ and N_V is the nodes corresponding to $x_j^{(V)}$. We also subdivide N_S and N_V into N_S^+, N_S^-, N_V^+, N_V^-, which are the set of nodes from S_P, S_Q, ∂V_P, and ∂V_Q, respectively. In addition, N_m and N_n are the sets of nodes corresponding to C, where $|C| = |N_m| = |N_n|$. They also represent similarity and dissimilarity nodes between S_P and ∂V_P which correspond to the variables m and n, respectively. The constructed graph is $G = (N \cup \{s, t\}, A)$, where $N = N_x \cup N_m \cup N_n$. To convert our problem in Eq. (9) to a min-cut problem, we connect all the nodes corresponding to the vertices in S_Q and voxels in V_Q to sink node t as arcs with the cost K. All the similarity nodes in N_m are connected to the source node s as arcs with cost η while the dissimilarity nodes in N_n are connected to t with cost δ, which is different from [10]. They are connected to/from the nodes in N_x with infinite capacity, as shown in Fig. 6. The neighbor nodes of S and ∂V are connected in both directions.

For a finite cut $(O \cup \{s\}, T \cup \{t\})$ of G, we denote the set of nodes connected to the source node s in N_x by O_x and those in N_m and N_n by O_m and O_n, respectively. In the same way, the nodes that are connected to the sink node t among N_x, N_m, and N_n are denoted by T_x, T_m, and T_n respectively. Eq. (9) can then be converted to a minimum (s, t)-cut problem by Theorem 1.

Theorem 1. *The minimum (s, t)-cut cost function of graph $G = (N \cup \{s, t\}, A)$ gives the same solution of Eq. (9) by setting $x_i = 1$ when the ith node of N_x is included in O that is connected to the source node s. $m_{j,r} = 1$ for each similarity node in N_m is included in T connected to the sink node t, and $n_{j,r} = 1$ for each dissimilarity node in N_n is included in O.*

Proof. Let $(O \cup \{s\}, T \cup \{t\})$ be a partition of $N \cup \{s, t\}$ forming a finite (s, t)-cut in G. This cut corresponds to a feasible solution because if $m_{j,r} = 1$ then $x_j = 1$ of N_S and $x_r = 1$ of N_V. If $n_{j,r} = 0$ then $x_j = 0$ and $x_r = 0$; otherwise an infinite cost arc will contribute to the cut capacity which violates the finiteness of the cut. The cut capacity is:

$$C(O \cup \{s\}, T \cup \{t\}) = \sum_{i \in O_x^-} K + \sum_{i \in O_x, j \in T_x} B_{i,j} + \sum_{i \in T_m} \eta_i + \sum_{i \in O_n} \delta_i.$$

Because $T_m = N_m - O_m$, the cut capacity can be reformulated as $K\left|O_x^-\right| + \sum_{i \in O_x, j \in T_x} B_{i,j} + \left(\sum_{i \in N_m} \eta_i - \sum_{i \in O_m} \eta_i\right) + \sum_{i \in O_n} \delta_i$. Since $\sum_{i \in N_m} \eta_i$ is constant, minimizing $C(O \cup \{s\}, T \cup \{t\})$ becomes equivalent to minimizing $K|O_x^-| + \sum_{i \in O_x, j \in T_x} B_{i,j} - \sum_{i \in O_m} \eta_i + \sum_{i \in O_n} \delta_i$, which is exactly the same as the objective value of Eq. (9).

After solving Eq. (9) with the min-cut/max-flow algorithm, we save a set of resulting faces whose vertices are $x_j^{(S)} = 1$ as $S_C \subset S$ and a set of voxels where value $x_r^{(V)} = 1$ as $\partial V_C \subset \partial V$. The remaining regions $S_N = S - S_C$ and $\partial V_N = \partial V - \partial V_C$ are also saved, as shown in Fig. 7(b). We then merge S_C and ∂V_N, as described in Section 4.

3.4. Metal artifacts

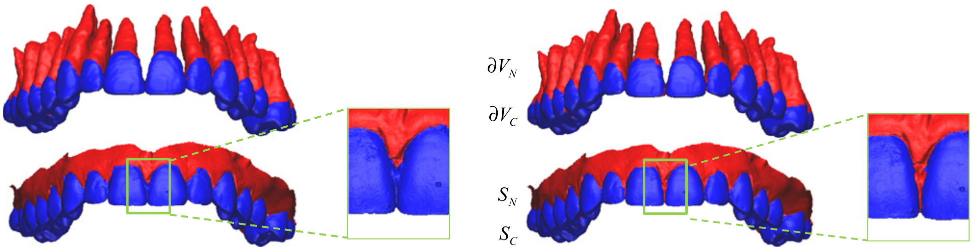
In many cases of dental CT images, critical artifacts occur due to metallic materials such as implants or prostheses, as shown in Fig. 1(c). This makes it hard to separate teeth from each other and to obtain the geometries of teeth. While our corresponding pair scheme and Eq. (9) give good results for normal dental CT without metal artifacts, it is not enough for other cases. However, the main advantage of co-segmentation is here. We can simply resolve this by complementing undesirable effects in dental CT with scanned data.

Since the shape of teeth with metal artifacts are dissimilar between CT and optically scanned data, most of the artifact region do not have corresponding pairs or are classified to the non-crown region by a dissimilarity term if Eq. (9) is applied directly, as shown in Fig. 9(a). Therefore, we detect the artifact region first, V_A in Fig. 8(d), and then make a corresponding set for V_A by finding the closest vertices from the optically scanned surface.

The artifact regions are generally included in ∂V_Q. In addition, the geometries of artifact regions are significantly different from neighboring regions. We use these two observations to detect artifact regions and perform segmentation.

Let the set of voxels be ∂V_R, which are collected from the z-direction extreme point, an apex of the root, until meeting V_P, as shown in Fig. 8(b). The seed regions of artifacts denoted by V_S to find the area affected by metal artifacts V_A are then determined by $\partial V_S = V_Q - \partial V_R$ in Fig. 8(c). V_A can then be found by searching from V_S to ∂V_R with a simple region growing method, as shown in Fig. 8(d).

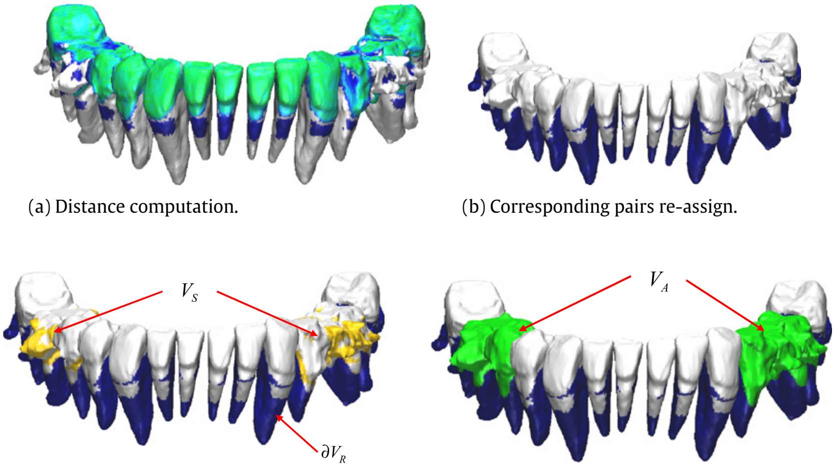
When we apply Eq. (9) to the model with artifacts, the similarity term is only applied in the case of the corresponding pairs of V_A, except for the dissimilarity term for co-segmentation. If the dissimilarity term is not applied to these corresponding pairs of V_A, the crown is found from the optically scanned data and the matched crown regions in V_A are classified by corresponding pairs. Fig. 9 shows our co-segmentation result of the model with metal artifacts.



(a) Co-segmentation result without dissimilarity term.

(b) Co-segmentation result with dissimilarity term.

Fig. 7. Examples of co-segmentation; the data sets used in (a) and (b) are the same; (b) shows better capability to classify non-crown regions such as interstices between teeth.



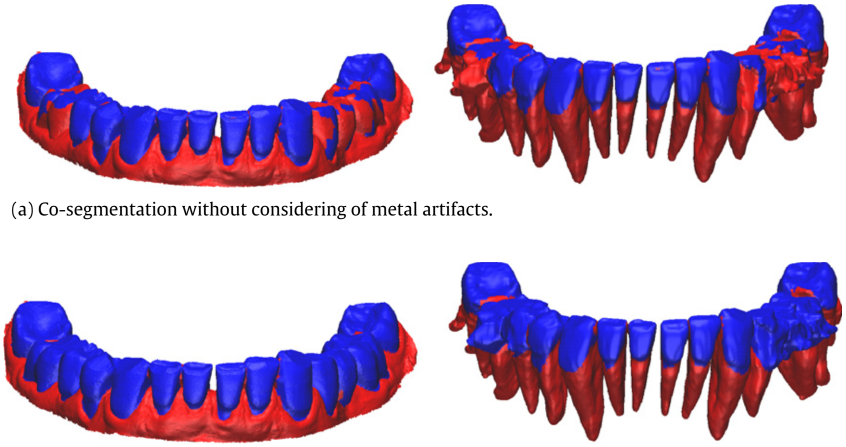
(a) Distance computation.

(b) Corresponding pairs re-assign.

(c) Seed region finding.

(d) Detection of artifact region of teeth.

Fig. 8. Detection of the teeth affected by metal artifacts.



(a) Co-segmentation without considering of metal artifacts.

(b) Co-segmentation by detecting artifact regions.

Fig. 9. Comparison of the co-segmentation results with/-out considering of metal artifacts; the blue denotes crown regions and the red color represents non-crown regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Merging the surface mesh and the volume

The goal of this study is to obtain a combined teeth model by aggregating strengths of each data type. The selected regions are not generally smoothly connected to ∂V, as shown in Fig. 10. For the vertices of S_C, we assign the tooth index with an index of the closest voxel in ∂V. The merge step is performed for the separated tooth independently after dividing S_C by $\cup_{i=1}^{n} S_{C_i}$.

The important point in this step is to maintain S_C as much as possible, because the accuracy of the optical scan data is much higher than the accuracy of dental CT. In order to achieve our goal, we adopt an approach that creates triangles from the boundaries of S_C while keeping the consistency of the surface normal vectors and triangle size. The overall procedure of this step is illustrated in Fig. 11.

For our purpose, the concept of the marching triangle algorithm introduced by Hilton et al. [28] is appropriate for determining new vertex positions and triangles from the boundary of S_C. This approach gives a fine solution to generate iso-surfaces from the arbitrary volume data [29] with an adaptive mesh size that considers curvature values while maintaining the aspect ratio of the triangles [30–33] by considering a 3D Delaunay surface

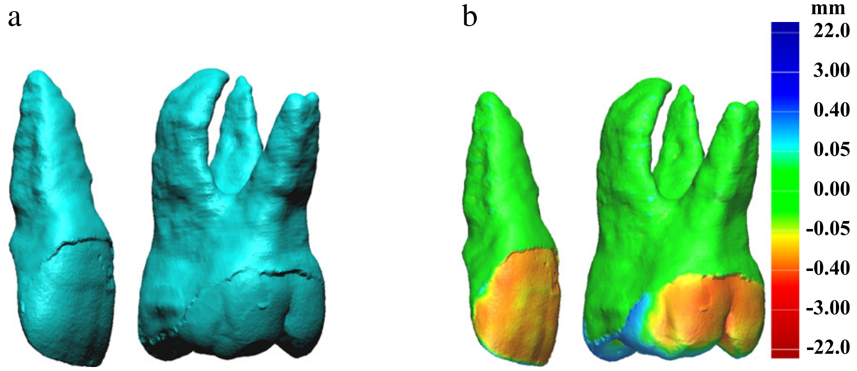


Fig. 10. Simply combined teeth models; (a) the simple stitching results of the boundary of S_C and the iso-surface from ∂V; (b) signed distance map of S_C from ∂V.

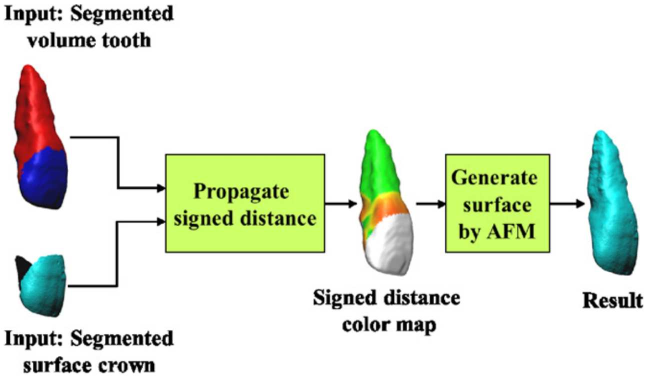

Input: Segmented volume tooth
Propagate signed distance
Generate surface by AFM
Signed distance color map
Result
Input: Segmented surface crown

Fig. 11. The overall procedure of the merging step.

constraint [28]. The difference is that our problem is not to create an iso-surface from a given implicit function or a scalar field, but to generate smoothly combined triangles to ∂V from the boundary of S_C. Let us denote the set of boundary edges and vertices of S_C by ∂S_C. The key idea in this step is to compute signed distances of ∂S_C from ∂V and assign them to the nearest voxels in ∂V. By propagating the values to ∂V_N by using a transition scheme that changes gradually, we are able to find the positions where new vertices lie. The details are described in this section.

To determine the vertex positions of the merged surface, we compute signed distance values of vertices in ∂S_C from ∂V. We then assign the values to the nearest voxels in ∂V. Let the distance value of an assigned voxel $\mathbf{q}$ as $h_0(\mathbf{q})$. We propagate the signed distances to the non-visited neighbor cells in ∂V_N. The value of the newly selected voxel is determined by averaging the visited neighbor cells to generate a smooth surface. These values represent the distances to the target surface from ∂V and we can determine the position of a new vertex using them during the advancing front method (AFM) procedure.

Another consideration is to make the target surface more closely resemble ∂V moving away from ∂S_C. We define the modified distance function $H(\mathbf{q})$ as Eq. (11) to apply this consideration by adopting the *geodesic* distance $d_{\partial V_C}$ from the nearest voxels of ∂S_C.

$$H(\mathbf{q}) = \begin{cases} h_0(\mathbf{q}) \times \dfrac{D_t - d_{\partial V_C}(\mathbf{q})}{D_t} & \text{if } d_{\partial V_C}(\mathbf{q}) < D_t \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where D_t is a user-defined constraint. We use the half value of the z-directional height of S_C as D_t for each tooth. Note that $H(\mathbf{q})$ is stored at $\mathbf{q} \in \partial V$, but the meaning is a target signed distance from ∂V indicating the position of the new vertex of the merged surface.

4.1. New vertex position computation

After computing the target signed distances as a preprocessing step, we then find a new vertex position through three steps. At each boundary face, f, whose edge $E \in f$ is included in ∂S_C, we can generate a new initial vertex $\mathbf{v}_0$ on the same plane with f and in the opposite direction at the midpoint of E, as shown in Fig. 12(a).

The next step is to find the normal vector of $\mathbf{v}_0$. We approximate a normal vector of $\mathbf{v}_0$ by comparing the k-nearest voxel positions from $\mathbf{v}_0$. We define $\mathbf{n}_{\mathbf{v}_0}$ as shown in Fig. 12(b):

$$\mathbf{n}_{\mathbf{v}_0} = \frac{\mathbf{v}_0 - \frac{1}{k} \sum\limits_{i \in N_k(x)} \mathbf{p}_i}{\left| \mathbf{v}_0 - \frac{1}{k} \sum\limits_{i \in N_k(x)} \mathbf{p}_i \right|}.$$

The new position $\mathbf{v}_{new}$ is then created along the $\mathbf{n}_{\mathbf{v}_0}$ direction from $\mathbf{v}_0$. Finally we can determine the new vertex position using Eq. (13). Since computing an accurate distance value is very time-consuming and not necessary in this application, we approximate the distance by averaging the pre-calculated values at voxels. Rather than simply averaging $\mathrm{H}(\mathbf{p}_r)$, we consider the angular distributions of $\mathbf{p}_r$. The weight of $\mathbf{p}_r$ is computed as an angular occupancy after projecting to the plane that passes $\mathbf{v}_0$ and has the normal vector as $\mathbf{n}_{\mathbf{v}_0}$. Let us denote the projected point of $\mathbf{p}_r$ as $\mathbf{q}_r$ and its circumcenter as $\mathbf{c}_r$. The weight of $\mathbf{q}_r$ is $\lambda_r(\mathbf{v}_0)$, as shown in Fig. 12(c).

The target distance to be maintained at $\mathbf{v}_0$ is then defined in Eq. (12).

$$h_t(\mathbf{v}_0) = \sum_{r \in N(\mathbf{v}_0)} \lambda_r(\mathbf{v}_0) \cdot H(\mathbf{p}_r^{\mathbf{v}_0}), \quad \text{where } \lambda_r(\mathbf{v}_0) = \frac{\theta_r}{2\pi}. \tag{12}$$

The distance at arbitrary position $\mathbf{v}$ can be computed as Eq. (13) where $\mathbf{n}_r^{\mathbf{v}_0}$ is the normal vector of $\mathbf{p}_r^{\mathbf{v}_0}$; the new vertex position $\mathbf{v}_{new}$ is then determined by the condition $h(\mathbf{v}_{new}) = h_t(\mathbf{v}_0)$.

$$h(\mathbf{v}) = \sum_{r \in N(\mathbf{v}_0)} \lambda_r(\mathbf{v}_0)(\mathbf{v} - \mathbf{p}_r^{\mathbf{v}_0}) \cdot \mathbf{n}_r^{\mathbf{v}_0}. \tag{13}$$

From $\mathbf{v}_0$ as the initial value, $\mathbf{v}_{new}$ can be iteratively found along the projection direction $\mathbf{n}_{\mathbf{v}_0}$ by a binary search. The practical experiments show that $\mathbf{v}_{new}$ is found in approximately five iterations within 0.001 mm tolerance.

4.2. Mesh completion by AFM

As mentioned before, even though our approach is technically similar to marching triangles, we call our method AFM since we do not extract an iso-surface. There are two possible cases where we can create new triangles: one of them is to create a face

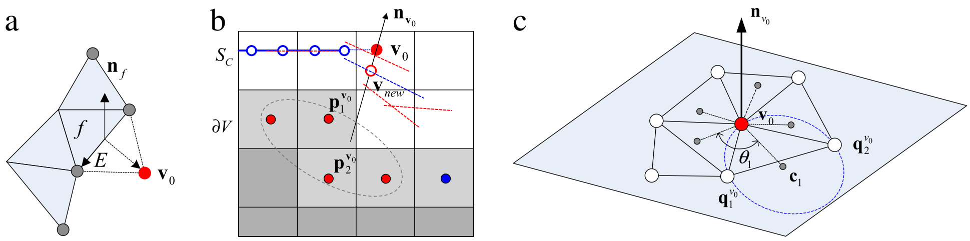
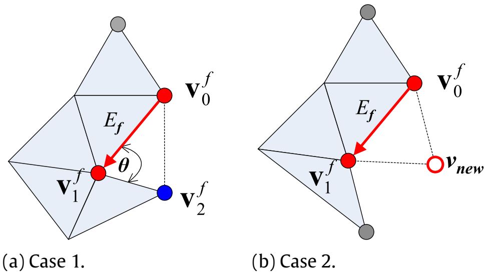

a
$\mathbf{n}_f$
f
E
$\mathbf{v}_0$
b
$\mathbf{n}_{\mathbf{v}_0}$
S_C
$\mathbf{v}_0$
$\mathbf{v}_{new}$
$\mathbf{p}_1^{\mathbf{v}_0}$
∂V
$\mathbf{p}_2^{\mathbf{v}_0}$
c
$\mathbf{n}_{\mathbf{v}_0}$
$\mathbf{v}_0$
θ_1
$\mathbf{c}_1$
$\mathbf{q}_2^{\mathbf{v}_0}$
$\mathbf{q}_1^{\mathbf{v}_0}$

Fig. 12. Computation of the new vertex position; (a) new initial vertex position; (b) normal vector of the projection plane; (c) angular distribution computation for weight computation.



(a) Case 1.

(b) Case 2.

Fig. 13. Two cases to generate a new triangle.

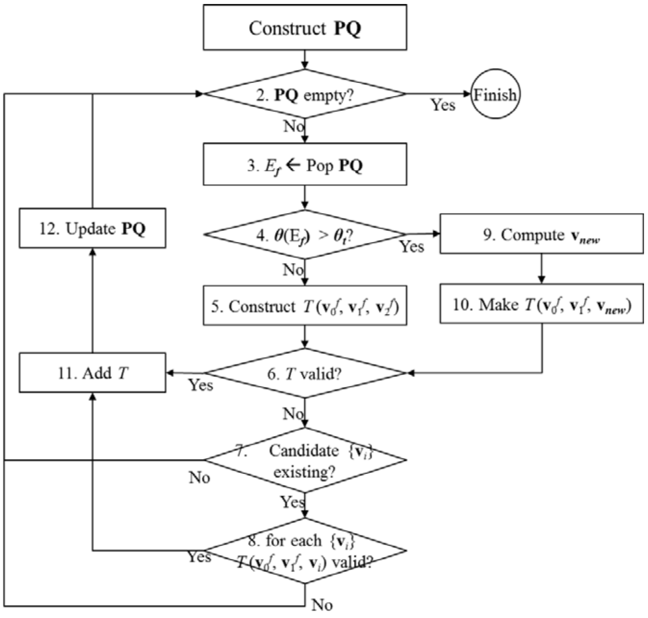
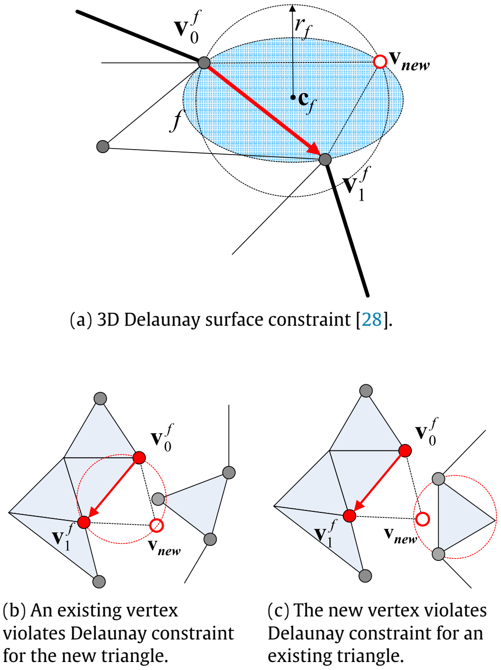

Construct PQ
2. PQ empty?
Yes
Finish
No
3. E_f ← Pop PQ
12. Update PQ
4. $\theta(E_f) > \theta_t$?
Yes
9. Compute $\mathbf{v}_{new}$
No
5. Construct $T(\mathbf{v}_0^f, \mathbf{v}_1^f, \mathbf{v}_2^f)$
10. Make $T(\mathbf{v}_0^f, \mathbf{v}_1^f, \mathbf{v}_{new})$
11. Add T
Yes
6. T valid?
No
7. Candidate $\{\mathbf{v}_i\}$ existing?
No
Yes
8. for each $\{\mathbf{v}_i\}$ $T(\mathbf{v}_0^f, \mathbf{v}_1^f, \mathbf{v}_i)$ valid?
Yes
No

Fig. 14. Overall procedure of AFM.

with existing vertices and the other one is to connect an existing edge and a new vertex generated by the method introduced in Section 4.1. Fig. 13 shows the two cases. Fig. 13(a) shows the case involving connection with adjacent edges whose angle is less than θ_t. If the angle with an adjacent edge in ∂S_C is larger than θ_t, we check the 3D Delaunay surface constraint to prevent local folding or self-intersections. If there is a vertex $\mathbf{v}_i$ within the Delaunay sphere, then we connect it to the current edge E_f. Otherwise, a new vertex is created.

The overall procedure of our AFM is described in Fig. 14. First, we initialize the priority queue (PQ), which contains boundary active edges in the order of the angles to the next connected edges shown in Fig. 13(a) in ascending order. To compute angles efficiently and maintain PQ, the boundary edges are stored as a half edge structure. If $\theta > \theta_t$, we compute the new vertex position $\mathbf{v}_{new}$ for the current active edge E_f, and check the 3D Delaunay surface constraints. If the triangle T satisfies the constraint, we add T as a new face to the merge result.



(a) 3D Delaunay surface constraint [28].

(b) An existing vertex violates Delaunay constraint for the new triangle.

(c) The new vertex violates Delaunay constraint for an existing triangle.

Fig. 15. 3D Delaunay surface constraint and non-satisfied cases.

4.2.1. *3D Delaunay surface constraint*

The 3D Delaunay surface constraint is similar to the Delaunay constraint of 2D triangulation. With a triangular face, the meaning of the condition is that there should be no point inside the sphere, called a *Delaunay sphere*, whose center and radius are consistent with those of the circumcircle of the triangle [28]. Fig. 15(a) shows a conceptual diagram and Fig. 15(b),(c) describes cases where $\mathbf{v}_{new}$ does not satisfy this constraint. To maintain a good aspect ratio, we need to make a new triangle by using the existing vertex which satisfies this constraint.

4.2.2. *Checking validity of a new triangle*

If we only create a new triangle when it satisfies the 3D Delaunay surface constraint by checking the vertices as shown in Fig. 15, it may cause numerous cracks [33]. In the case where the 3D Delaunay surface constraint is not satisfied, we can find candidate vertices $\{\mathbf{v}_i\}$ inside the Delaunay sphere. Among them, we select the vertex that satisfies the Delaunay surface constraint when a new triangle is created with it. Fig. 16 shows several cases of checking the validity of new triangles and the yellow color represents the solution of each case. Fig. 17 shows the result of the merging step suggested in this work. Two different regions are connected successfully while maintaining the details of crown features.

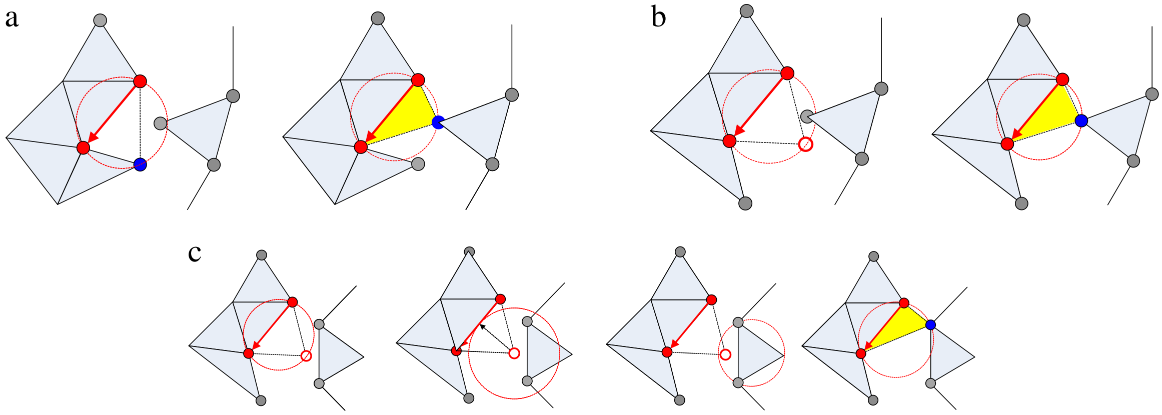
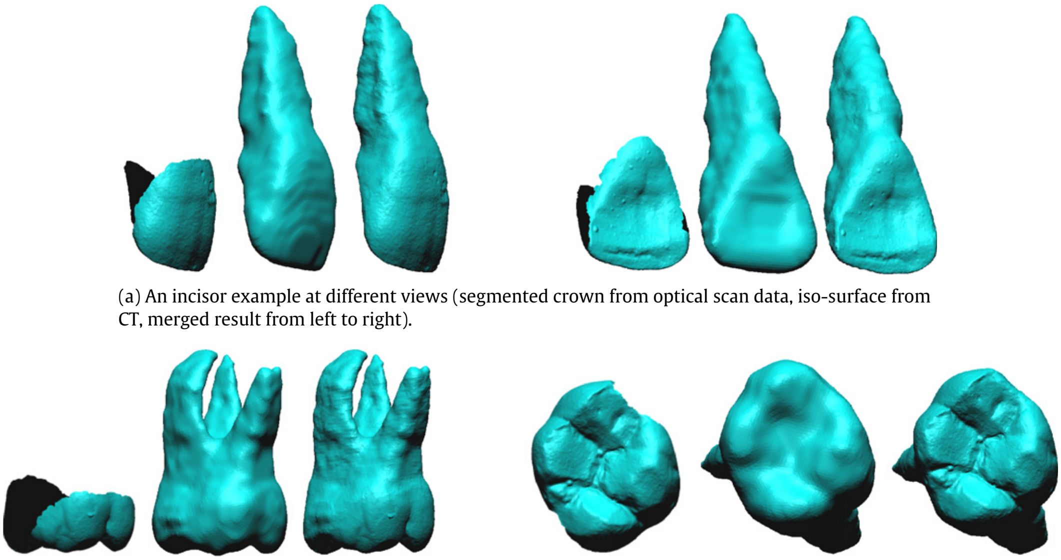


Fig. 16. Validity check of a new triangle: three different cases of creating a triangle with existing vertices; (a) the case violates the 3D Delaunay constraint of case 1 in Fig. 13; (b) the case violates the constraint of case 2; (c) the case when a new vertex violates the constraint of the existing face. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) An incisor example at different views (segmented crown from optical scan data, iso-surface from CT, merged result from left to right).

(b) A molar example at a different views (segmented crown from optical scan data, iso-surface from CT, merged result from left to right).

Fig. 17. Comparison between optical scan data, iso-surfaces from dental CT, and the merged models of individual teeth from different view directions.

4.2.3. Adaptive triangle size

Because the resolution between surface meshes and dental CT images are different, we do not need to maintain the scanned surface mesh resolution in a root region higher than CT resolution. A large number of triangles may degrade the efficiency and usability in many applications. For efficiency, triangle sizes adopt the size of the propagated triangle and local curvature with a given error ε. Eq. (14) shows the rule to determine the new edge length of the new triangle T [31].

$$L(T) = 2 \times \sqrt{\frac{1}{\left(\kappa_{|\max|}\right)^2} - \left(\frac{1}{\kappa_{|\max|}} - \varepsilon\right)^2} \tag{14}$$

where $\kappa_{|\max|} = \max(|\kappa_{\max}|, |\kappa_{\min}|)$ and $\kappa_{\max}$, $\kappa_{\min}$ are the normal maximum curvature and minimum curvature of a voxel, respectively. The normal curvature of volume data can be computed by the method introduced in [34], and we set $\varepsilon = 0.001$ in our experiments. In Eq. (14), $1/\kappa_{|\max|}$ is the radius of the smallest osculating circle at a tooth's surface point as shown in Fig. 18. The length of the target edge $L(T)$ is then calculated by Eq. (14).

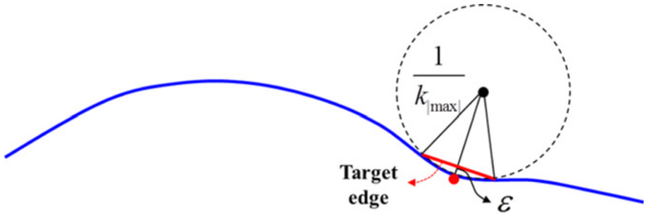


Fig. 18. Edge length calculation; the blue curve represents the tooth surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 19 shows a comparison of the results considering the adaptive size (Fig. 19(c)) and otherwise (Fig. 19(b)) for the same tooth model.

4.3. Metal artifacts

To handle metal artifact data in this merging step, we simply remove the crowns that have artifacts from the merging area. Most of the crown surfaces are recovered from the scan data, and the missing part can be restored by smooth hole filling considering

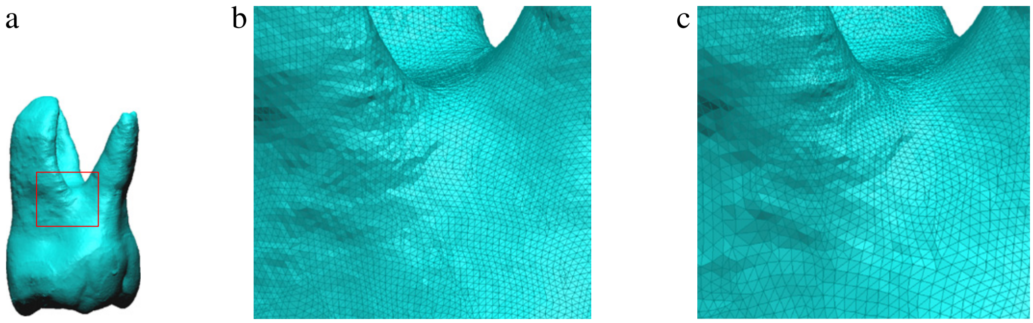


Fig. 19. Adaptive mesh generation; (a) a single molar and highlighted region; (b) mesh generation without considering the size; (c) mesh generation with adaptive size.

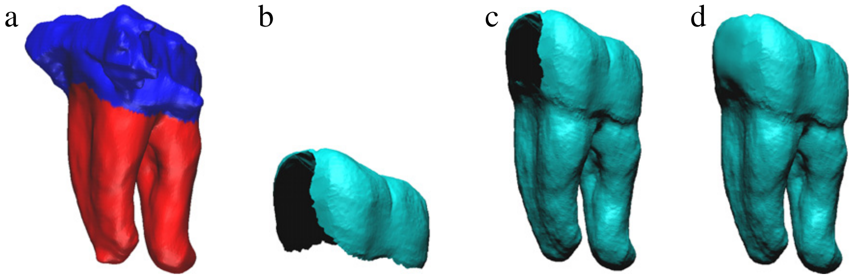


Fig. 20. Merging result of the tooth with the metal artifact; (a) co-segmentation result of a tooth extracted from dental CT; (b) a crown obtained by co-segmentation with scanned surface; (c) merging result; (d) the final complete tooth after filling the hole.

Table 1
Patient raw data.

Set	Region	CT size resolution (mm)	Metal artifact	Optical scan data				No. corresponding pairs
				Scan	No. points	No. faces	EL[c] (mm)	
1	Maxilla	640, 640, 314 0.250, 0.250, 0.250	No	IO[a]	167,845	333,538	0.174	131,452
2	Mandible	512, 512, 510 0.160, 0.160, 0.160	No	PS[b]	59,555	117,994	0.320	94,872
3	Mandible	512, 512, 103 0.216, 0.216, 0.400	Yes	IO	137,125	272,531	0.190	115,443

[a] Intraoral scanning.
[b] Plaster cast.
[c] Average edge length.

the boundary shapes [6,35–37]. Fig. 20 shows the final tooth result with metal artifacts.

5. Experimental results

5.1. Patient data selection

We selected three data sets for the experiments, as shown in Table 1. Even though the number of data sets was not large, it was enough to test the usability and efficiency of our algorithm for various cases because each data set included more than 10 teeth. To test various kinds of data, different data sets from different regions were used.

The optical scan data for sets 1 and 3 were acquired from the iTero intraoral scanner of Align Technology Inc. (San Jose, CA, USA), and its manufacturer's accuracy specification was 20 μm. Set 2 scan data was acquired from Dental Wings Inc.' table scanner (Dental Wings Inc., Montreal, QC, Canada), whose accuracy specification was 15 μm.

5.2. Results

Figs. 21, 22, and 23 show the co-segmentation and merging results for data sets 1, 2, and 3, respectively. As shown in these three examples, the proposed algorithm gives high quality results for various data sets regardless of the different shape of teeth without any training set.

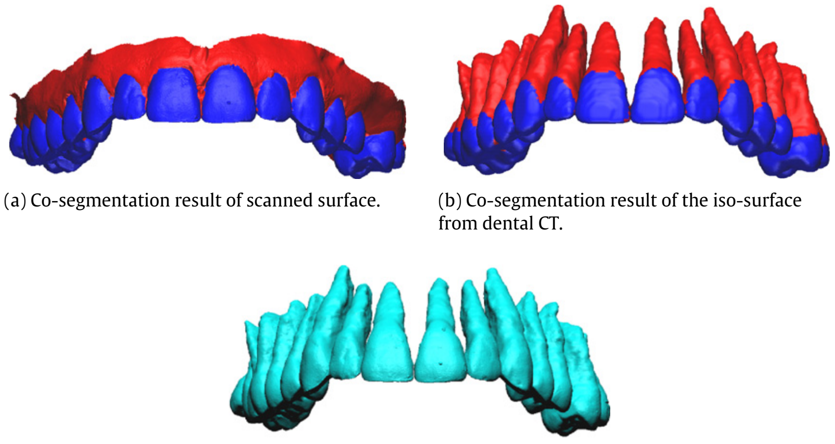
Table 2
Computational time.

Set	Co-segmentation (s)	Merging (s)	Total (s)
1	10.26	15.24	25.50
2	8.36	14.85	23.21
3	8.98	15.02	24.01

We implemented our algorithm using Microsoft Visual C++ version 10.0. The testing platform is a personal computer composed of an Intel Core i7 3 GHz CPU and 8 GB main memory. The computation time with this platform is shown in Table 2.

Table 2 shows the total time required for generating all the teeth models of each data set—15 teeth, 12 teeth, and 14 teeth for data sets 1, 2, and 3, respectively. The computation time of co-segmentation is affected by the number of corresponding pairs and the number of voxels and points classified as the crown region for each data type. In the case of the merging step, the computation time is mainly affected by the voxel resolution and the number of teeth in our implementation.

From the experiments, our approach shows reasonable time efficiency. This approach is much faster than manual work, and most applications of teeth model completion are in the diagnosis or planning stage and do not need to be performed in real time.

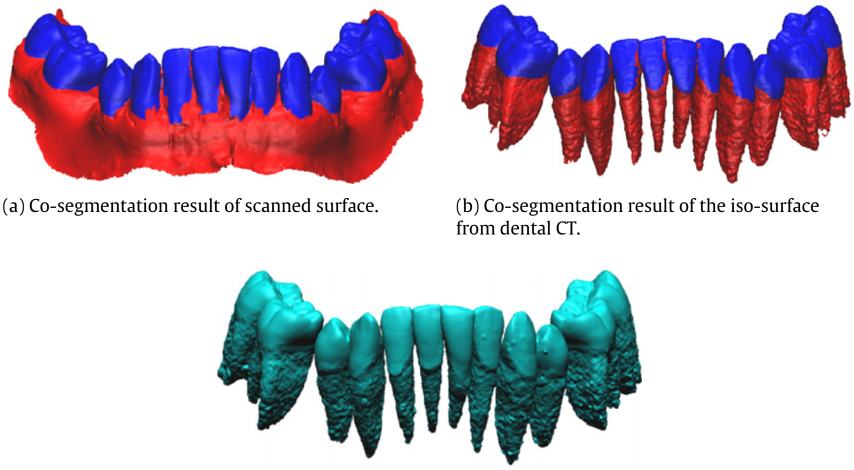


(a) Co-segmentation result of scanned surface.

(b) Co-segmentation result of the iso-surface from dental CT.

(c) Combined result surface.

Fig. 21. Final result of a maxilla case with 15 teeth (data set 1).



(a) Co-segmentation result of scanned surface.

(b) Co-segmentation result of the iso-surface from dental CT.

(c) Combined result surface.

Fig. 22. Final result of a mandible case with 12 teeth (data set 2).

5.3. Evaluation

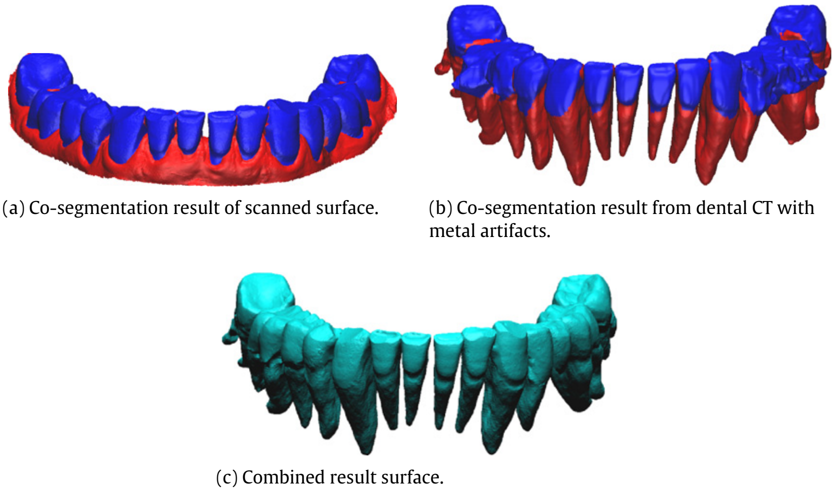
In our problem, it is not simple to measure numerical accuracies of the segmentation results. Especially for volumetric data, accuracy is dependent on the spatial resolution of the volume and scanning position. In addition, for our problem, the co-segmentation result is more important than single segmentations. One of the valid methods is to compare the manual segmentation results obtained from specialists in this field by considering that manual segmentation is practically used in many dental applications. Therefore, we measured traditional sensitivity and specificity based on the manual segmentation results, widely used in medical image segmentation, to evaluate the proposed method for co-segmentation.

First, an expert who has been working for a dental solution company for more than 3 years manually classified the crown and non-crown regions with the scanned surface. We then compared the segmentation results to the manual classifications by considering the latter as ground-truths. To evaluate the proposed co-segmentation approach, we also tested single segmentation results using surface meshes only via graph-cut and co-segmentation results without a dissimilarity term. We only evaluated the co-segmentation results of the scanned surfaces except for the volume data and merging part, because it is difficult to obtain the ground-truth of the root region. Furthermore, it is not meaningful to compare the root regions due to their low resolution and artifacts such as partial volume effects. In addition, the exact boundary geometries of root regions are not important but the overall positions and directions are considered to be meaningful information in many dental applications.

To quantify the evaluation, we performed sensitivity-specificity analysis for the test sets. The positive region represents the crown region, and the negative region is mapped to the non-crown region in our problem. The sensitivity and specificity are defined as Eq. (15). *TP* means *true positive*, which means correctly identified as the crown region, the vertices classified as crowns on both sides of the co-segmentation result and the ground-truth. *FN* means *false negative*, which means incorrectly rejected vertices in the co-segmentation result even though they are classified as the crown region in the ground-truth. Similarly, *FP* means *false positive*, which means incorrectly identified, and TN means *true negative*, which means correctly rejected. The meaning of the symbol # is '*the number of*'.

$$\begin{aligned} \text{Sensitivity} &= \frac{\# TP}{(\# TP) + (\# FN)}, \\ \text{Specificity} &= \frac{\# TN}{(\# TN) + (\# FP)}. \end{aligned} \tag{15}$$

From Table 3, we can observe that the co-segmentation result with the additional dissimilarity term and similarity shows the highest specificity for the classification ability of non-crown



(a) Co-segmentation result of scanned surface.

(b) Co-segmentation result from dental CT with metal artifacts.

(c) Combined result surface.

Fig. 23. Final result of a mandible case with 14 teeth with metal artifacts (data set 3).

Table 3
Evaluation.

Set	Algorithm	Sensitivity	Specificity
	Surface mesh[a]	0.999	0.857
1	Co-seg_S[b]	1.0	0.919
	Co-seg_SD[c]	0.987	**0.998**
	Surface mesh	0.945	0.934
2	Co-seg_S	1.0	0.959
	Co-seg_SD	0.988	**0.997**
	Surface mesh	0.997	0.879
3	Co-seg_S	0.996	0.924
	Co-seg_SD	0.962	**0.975**

[a] Segmentation using surface mesh only.
[b] Co-segmentation using similarity term only.
[c] Co-segmentation using similarity and dissimilarity terms.

regions. On the other hand, the sensitivity with the dissimilarity term is slightly lower than the segmentation using similarity only. This is because the co-segmentation method with only the similarity term determines the region close to gingival lines as the crown.

In our approach, the higher specificity gives more meaningful results because non-crown regions are extracted from the dental CT data and small missing crown regions also can be recovered from CT images smoothly. However, non-crown regions from the scanned surface may generate erroneous geometry because the segmented crown regions from S are all used in the merging step.

The classification ability of our approach between teeth, interstices, and gingival regions is shown experimentally. Our method can be applied to dental CT containing metal artifacts with high accuracy on par with that of normal data.

6. Conclusion

We have developed a novel teeth modeling framework by combining optical scan data and dental CT images. The suggested approach can overcome difficulties in completing teeth models from each data type using existing approaches. This approach is more meaningful when one data set has defects such as metal artifacts in dental CT.

Dental computer-aided design (CAD) systems have been intensively introduced to digital dentistry in recent years. As basic digital models, individual teeth models with complete and accurate shapes are largely available in many dental fields including orthodontics, especially when planning treatments or surgeries. In addition, digitized planning data can be directly used for producing surgical guides or orthodontic appliances if we have a complete teeth model. Nevertheless, few cases have been practically used to date because it is difficult, expensive, and time-consuming to obtain an individual teeth model.

Our approach is robust and time efficient, and shows high fidelity. Because our method remedies the disadvantages and defects of one data set from another data type, it gives robust results regardless of data properties. It may lead to many related modules of dental CAD by significantly reducing the interventions of manual operations. In addition, we anticipate that our study may lead to the increased usability of digital teeth models in virtual surgery and planning and can also be used for diagnosis purposes in orthodontics. This method can also be helpful in reducing the time and cost involved fabricating orthodontic appliances.

Acknowledgments

Our program was implemented on OnDemand3D system of Cybermed Inc. This research was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Science, ICT and Future Planning (NRF-2012R1A1A3012995).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cad.2015.04.008.

References

[1] Gao H, Chae O. Individual tooth segmentation from CT images using level set method with shape and intensity prior. Pattern Recognit 2010;43(7):2406–17.
[2] Gao H, Chen JX, Chae O. A new visualization framework of dental CT images for orthodontics. In: Proceedings of the 9th ACM SIGGRAPH conference on virtual-reality continuum and its applications in industry. 2010, p. 353–60.
[3] Hosntalab M, Zoroofi RA, Tehrani-Fard AA, Shirani G. Segmentation of teeth in CT volumetric dataset by panoramic projection and variational level set. Int J Comput Assist Radiol Surg 2008;3(3–4):257–65.
[4] Kronfeld T, Brunner D, Brunnett G. Snake-based segmentation of teeth from virtual dental casts. Comput-Aided Des Appl 2010;7(2):221–33.
[5] Kumar Y, Janardan R, Larson B, Moon J. Improved segmentation of teeth in dental models. Comput-Aided Des Appl 2011;8(2):211–24.
[6] Yuan T, Liao W, Dai N, Cheng X, Yu Q. Single-tooth modeling for 3D dental model. J Biomed Imaging 2010;2010: Article no. 9.
[7] Kondo T, Ong SH, Foong KWC. Tooth segmentation of dental study models using range images. IEEE Trans Med Imaging 2004;23(3):350–62.
[8] Wu T, Liao W, Dai N. Three-dimensional statistical model for gingival contour reconstruction. IEEE Trans Biomed Eng 2012;59(4):1086–93.
[9] Fan R, Jin X, Wang CCL. Multiregion segmentation based on compact shape prior. IEEE Trans Autom Sci Eng 2014;PP(99):1–12.
[10] Hochbaum DS, Singh V. An efficient algorithm for co-segmentation. In: Proceedings of IEEE international conference on computer vision. 2009, p. 269–76.

[11] Rother C, Kolmogorov V, Minka T, Blake A. Cosegmentation of image pairs by histogram matching—incorporating a global constraint into MRFs. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR). 2006, p. 993–1000.

[12] Song Q, Bai J, Garvin MK, Sonka M. Optimal multiple surface segmentation with shape and context priors. IEEE Trans Med Imaging 2013;32(2):376–86.

[13] Song Q, Bai J, Han D, Bhatia S, Sun W, Rockey W, Bayouth JE, Buatti JM, Wu X. Optimal co-segmentation of tumor in PET-CT images with context information. IEEE Trans Med Imaging 2013;32(9):1685–97.

[14] Besl PJ, McKay ND. Method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 1992;14(2):239–56.

[15] Batra D, Kowdle A, Parikh D. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR). 2010, p. 3169–76.

[16] Chiu WC, Fritz M. Multi-class video co-segmentation with a generative multi-video model. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR). 2013, p.321–28.

[17] Hu R, Fan L, Liu L. Co-segmentation of 3D shapes via subspace clustering. Comput Graph Forum 2012;31(5):1703–13.

[18] Boykov Y, Funka-Lea G. Graph cuts and efficient N-D image segmentation. Int J Comput Vis 2006;70(2):109–31.

[19] Delong A, Boykov Y. A scalable graph-cut algorithm for N-D grids. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR). 2008, p. 1–8.

[20] Rother C, Kolmogorov V, Blake A. GrabCut—interactive foreground extraction using iterated graph cuts. ACM Trans Gr 2004;23(3):309–14.

[21] Ulén J, Strandmark P, Kahl F. An efficient optimization framework for multi-region segmentation based on Lagrangian duality. IEEE Trans Med Imaging 2013;32(2):178–88.

[22] Golovinskiy A, Funkhouser T. Consistent segmentation of 3D models. Comput Graph 2009;33(3):262–9.

[23] Li K, Wu X, Chen DZ, Sonka M. Optimal surface segmentation in volumetric images—A graph-theoretic approach. IEEE Trans Pattern Anal Mach Intell 2006;28(1):119–34.

[24] Shalom S, Shapira L, Shamir A, Cohen-Or D. Part analogies in sets of objects. In: Proceedings of eurographics workshop on 3D object retrieval. 2008, p. 33–40.

[25] Shapira L, Shamir A, Cohen-Or D. Consistent mesh partitioning and skeletonisation using the shape diameter function. Vis Comput 2008;24(4):249–59.

[26] Shapira L, Shalom S, Shamir A, Cohen-Or D, Zhang H. Contextual part analogies in 3D objects. Int J Comput Vis 2010;89(2–3):309–26.

[27] Hoffman DD, Singh M. Salience of visual parts. Cognition 1997;63(1):29–78.

[28] Hilton A, Stoddart AJ, Illingworth J, Windeatt T. Marching triangles: range image fusion for complex object modeling. In: Proceedings of international conference on image processing. Vol. 2. 1996. p. 381–84.

[29] Rosenthal P, Linsen L. Direct isosurface extraction from scattered volume data. In: Proceedings of the eighth joint eurographics / IEEE VGTC conference on visualization. 2006. p. 99–106.

[30] Akkouche S, Galin E. Adaptive implicit surface polygonization using marching triangles. Comput Graph Forum 2001;20(2):67–80.

[31] de Araujo BR, Jorge JAP. Curvature dependent polygonization of implicit surfaces. In: Proceedings of XVII Brazilian symposium on computer graphics and image processing. 2004, p. 266–73.

[32] Karkanis T, Stewart AJ. Curvature-dependent triangulation of implicit surfaces. IEEE Trans Comput Graph Appl 2001;21(2):60–9.

[33] Xi Y, Duan Y. A novel region-growing based iso-surface extraction algorithm. Comput Graph 2008;32(6):647–54.

[34] Kindlmann G, Whitaker R, Tasdizen T, Möller T. Curvature-based transfer functions for direct volume rendering methods and applications. In: Proceedings of IEEE visualization. 2003, p. 513–20.

[35] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: Proceedings of international symposium on 3D data processing visualization and transmission. 2002, p. 428–41.

[36] Yu Y, Zhou K, Xu D, Shi X, Bao H, Guo B, Shum H. Mesh editing with Poisson-based gradient field manipulation. ACM Trans Gr 2004;23(3):644–51.

[37] Zhao W, Gao S, Lin H. A robust hole-filling algorithm for triangular mesh. Vis Comput 2007;23(12):987–97.