

# An Interactive Geometric Technique for Upper and Lower Teeth Segmentation

Binh Huy Le<sup>1</sup>, Zhigang Deng<sup>\*1</sup>, James Xia<sup>2</sup>, Yu-Bing Chang<sup>2</sup>, Xiaobo Zhou<sup>\*2</sup>

<sup>1</sup> Department of Computer Science, University of Houston, Houston, Texas

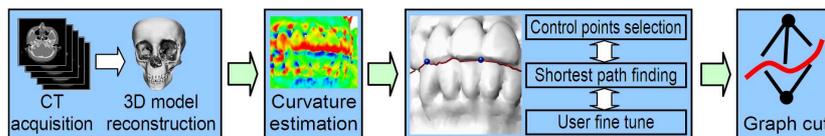
<sup>2</sup> The Methodist Hospital Research Institute, Houston, Texas

\* Emails of corresponding authors: zdeng@cs.uh.edu and xzhou@tmhs.org

**Abstract.** Due to the complexity of the dental models in semantics of both shape and form, a fully automated method for the separation of the lower and upper teeth is unsuitable while manual segmentation requires painstakingly user interventions. In this paper, we present a novel interactive method to segment the upper and lower teeth. The process is performed on 3D triangular mesh of the skull and consists of four main steps: reconstruction of 3D model from teeth CT images, curvature estimation, interactive segmentation path planning using the shortest path finding algorithm, and performing actual geometric cut on 3D models using a graph cut algorithm. The accuracy and efficiency of our method were experimentally validated via comparisons with ground truth (manual segmentation) as well as the state of art interactive mesh segmentation algorithms. We show the presented scheme can dramatically save manual effort for users while retaining an acceptable quality (with an averaged 0.29 mm discrepancy from the ideal segmentation).

## 1 Introduction

Computed tomography (CT) images are commonly used in cranio-maxillofacial (CMF) surgery, orthodontics and dentistry. It is especially true when cone-beam CT (CBCT) scanners are introduced. CBCT scanners have much lower radiation than medical spiral CT scanners while the thickness of each slice is much thinner (0.125mm-0.4mm per slice thickness). CBCT scanners are now extensively used in dental offices to replace the plain cephalometric and panoramic radiographic machines. One of the main interests in CMF surgery and orthodontics is the teeth. In order to quantify the deformity accurately, a CT scan is usually completed when the maxillary (upper) and mandibular (lower) teeth are in centric occlusion (peak and valley on the teeth bite down tightly). This brings us a major problem: *the separation of the maxillary and mandibular teeth*. Due to the irregular 3D geometry of the teeth, they are usually segmented manually by drawing the maxillary and mandibular teeth on each cross-sectional slice. It is time consuming and difficult to segment the peaks of one jaw and the valleys of the opposite jaw on a single slice. It becomes even more difficult if the scatterings (artifacts) are present due to dental restoration and orthodontic braces.



**Fig. 1.** Schematic view of this interactive approach for the lower and upper teeth segmentation

The popularized scheme of current 3D mesh segmentation approaches is to extract and optimize certain application-specific mesh features. One common issue of these automated mesh segmentation approaches is that users cannot control or refine the process if the approaches fail to produce plausible segmentation results on complicated geometric models. Many semi-automatic mesh segmentation algorithms [1–5] were attempted for the purpose of mesh segmentation. Researchers have also developed teeth-specific segmentation techniques to separate individual teeth from a teeth dataset [6, 7]. These approaches assume that the upper and lower 3D teeth can be perfectly separated by a plane. Nevertheless, it is nontrivial to extend these methods for the separation of complicated geometrical models such as the maxillary and mandibular teeth in this work.

In this paper, we present a novel interactive technique for segmenting the upper and lower teeth with limited user interventions. First, we reconstruct a 3D triangular mesh model from the acquired teeth CT images. Then, we compute the curvatures of the triangular mesh such that they are more sensitive in the vertical direction than in the horizontal direction. Then, through minimized user interventions such as selecting several control points, we construct a cost function and search for the optimal segmentation path on the mesh. Finally, the graph cut algorithm [8] is employed to handle the remaining isolated sticky parts. Figure 1 shows the schematic view of our approach.

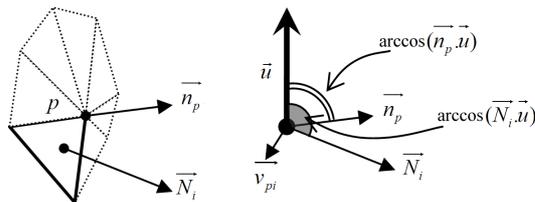
## 2 Our Approach

### 2.1 3D Teeth Model Reconstruction

The CT image data of patients’ craniofacial skeleton were acquired while the patients were on a centric occlusion. The CT scans were completed using a standard scanning algorithm: a resolution of  $512 \times 512$  at 0.625-1.25 mm slice thickness, 25cm or lesser field-of-view (FOV),  $0^\circ$  gantry tilt, and 1:1 pitch. Then, we use the open-source OsiriX imaging software (<http://www.osirix-viewer.com>) to reconstruct their corresponding 3D triangular mesh from the CT images. We remove triangles distant from the teeth to reduce unnecessary computations.

### 2.2 Curvature Estimation

Based on the observation that the upper and lower teeth are expected to be segmented along an approximate horizontal direction, we compute the curvature



**Fig. 2.** Illustration of how the curvature of  $P$  is computed

of each vertex of a 3D teeth model such that it is more sensitive to the vertical direction while less sensitive to other directions. In this process, we use a pre-defined up vector and then calculate the curvature  $K_P$  for each vertex  $P$  as the averaged normal vector difference between the normal at the vertex  $P$  and the normals of  $P$ 's neighboring triangles as follows:

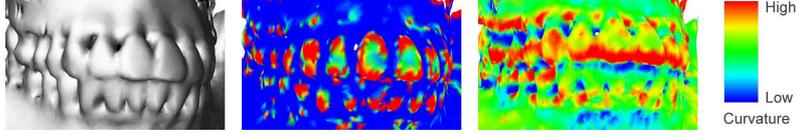
$$K_P = \frac{\sum_{i \in nbhd(P)} (\arccos(\vec{n}_p \cdot \vec{u}) - \arccos(\vec{N}_i \cdot \vec{u})) \frac{\text{sign}(\vec{u} \cdot \vec{v}_{pi})}{|\vec{v}_{pi}|}}{|nbhd(P)|} \quad (1)$$

Here  $nbhd(P)$  is the set of  $P$ 's neighboring triangles,  $\vec{n}_p$  is the normal vector of  $P$ ,  $\vec{N}_i$  is the normal vector of a neighboring triangle  $F_i$ ,  $\vec{u}$  is the pre-defined up vector,  $\vec{v}_{pi}$  is the vector from  $P$  to the centroid of  $F_i$ , the sign function  $\text{sign}(\vec{u} \cdot \vec{v}_{pi})$  determines the vertical direction of  $\vec{v}_{pi}$ . Eq. 1 incorporates the orientation into the curvature estimation in the following way: angles between normal vectors and the up vector are extracted, then, the sign function is used to determine the concaveness of a neighboring triangle. Finally, we weighted average the estimated curvatures of all neighboring triangles, and the weights are inversely proportional to the distance to the center of the triangle ( $v_{pi}$ ).

The above estimated curvatures can be used to determine the convex or concave property of the mesh vertices (Figure 2). Vertices with positive curvatures are called convex vertices; otherwise they are concave ones. In this work, we want the segmentation path to travel through the concave area of the mesh by simply thresholding the convex vertices. Note that since the acquired CT slices are in the XY plane, we generally specify the up vector as the  $Z^+$  direction. We also compared our curvature estimation scheme with the widely-used Gaussian curvature estimation algorithm [9]. As shown in Figure 3, the curvatures computed by our approach better separate the concave crevices in the horizontal direction (i.e., potential segmentation path area, shown as blue color in Figure 3) on the 3D teeth model than the Gaussian curvatures [9] where low curvature crevices are distributed everywhere.

### 2.3 Interactive Segmentation Path Planning

Because the upper and lower teeth are irregularly intertwined each other, an automatic method that fully depends on the curvature guidance would not always



**Fig. 3.** Original teeth model (left), visualization of Gaussian curvatures (middle) and visualization of the curvatures by our approach (right)

produce plausible segmentation. Thus, our approach allows users to select several control points on the model to guide the segmentation planning. Our approach computes an optimal segmentation path that travels through low curvature areas while satisfying the user-specified control points. During this process, users can interactively add or change the control points, and the corresponding segmentation paths will be updated in real-time.

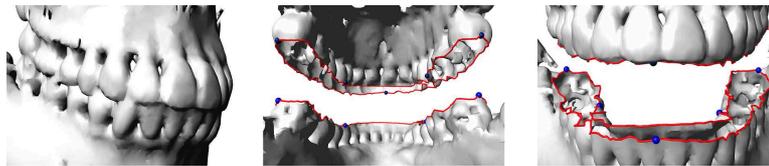
We employ the Dijkstra shortest path algorithm [10] to find the optimal segmentation path between two control points. This algorithm takes two control points as the source and the destination and then minimizes the overall cost. A cost function  $C$  between two neighboring vertices  $i$  and  $j$  is defined as follows:

$$C(i, j) = \frac{d(i, j)}{|K_i| + |K_j|} \quad (2)$$

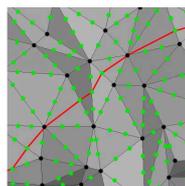
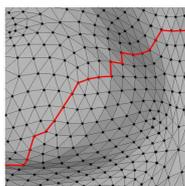
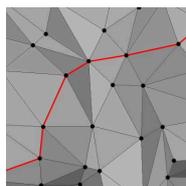
Where  $d(i, j)$  is the Euclidean distance between vertex  $i$  and vertex  $j$ ,  $K_i$  and  $K_j$  are the computed curvatures for vertices  $i$  and  $j$ , respectively (Eq. 1).

If the searched shortest path only consists of existing vertices on the 3D model, then it highly depends on the given mesh topology and may not be smooth. To alleviate this problem, many mesh segmentation methods perform a model subdivision (refinement) after an initial cut [5, 4, 3]. However, due to the lack of user controls on the mesh refinement process, the segmentation path on the refined mesh might measurably deviate from the original computed path, even though it may appear smoother. Mitchell et al. [11] tackle this problem by partitioning an edge into intervals so that the exact segmentation computation can be performed. The computational complexity of this algorithm is  $O(n^2 \log n)$  in the worst case where  $n$  is the number of vertices on the mesh, which is not efficient for practical interactive applications.

In this work, to achieve a trade-off between algorithm efficiency and accuracy, we use a simple while efficient partition strategy that divides an edge into a minimum number of equal intervals and each of them has a user-defined minimum length  $d_{min}$ . The curvatures of interval ends are linearly interpolated based on the two end points of the corresponding mesh edge, and we further assume only if two interval-ends belong to the same triangle, then these two points have a cost value; otherwise infinity. The computational complexity of our algorithm is  $O(n \log n)$  where  $n$  is the number of vertices on the mesh. Figure 4 shows the searched optimal segmentation path from the outside/inside views. Users selected 3 control points in the front side and 2 points inside the mouth. All these



**Fig. 4.** Results of interactive segmentation path planning (model #1). Selected control points are shown as blue points and the searched segmentation paths are red curves.



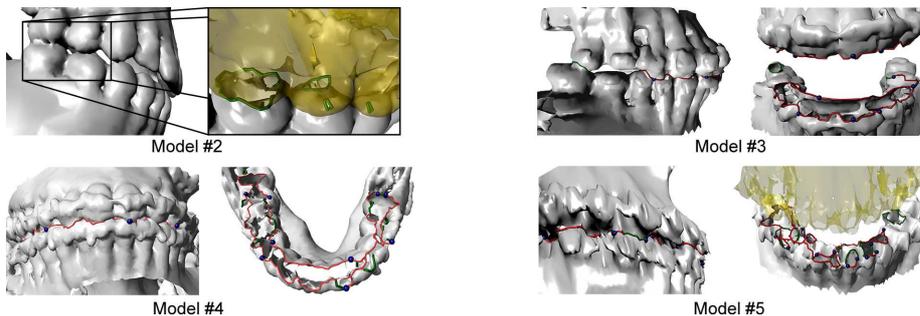
**Fig. 5.** The searched shortest segmentation path on a mesh without edge-partition (left), on the subdivision mesh (middle), and the mesh with our even edge-partition scheme (right)

control points are illustrated as blue points. In this dataset, there is no gap between the upper and lower teeth, thus our approach can compute a continuous segmentation path surrounding the mesh. Although the distance between two selected control points is relatively large, our algorithm is able to compute a plausible segmentation path with the aid of the estimated curvatures.

We also compared our approach (mesh with even edge-partition) with the original mesh (without edge-partition) and the subdivision mesh. As shown in Figure 5, the shortest path computed from the mesh with even edge-partition is smoother than the other two cases. It is noteworthy that the segmentation path on the subdivision mesh has more vertices than that on the mesh with even edge-partition, which means more computation time. Furthermore, the shortest segmentation path computed from the subdivision mesh is still more rougher than the path on the mesh with our even edge-partition scheme.

## 2.4 Graph Cut on Teeth Models

After major segmentation paths are computed through the above interactive interface, certain isolated sticky parts may remain to be separated. We automatically segment the remaining sticky parts by applying the graph cut algorithm [8]. This algorithm works on a flow network with multi sources and sinks where nodes are mesh triangles and edges are mesh edges. The sources are triangles on top of the mouth and sinks are triangles on the bottom. In this process, if a triangle is intersected with the above searched major segmentation paths (Section 2.3), then the capacities of all the edges of this triangle are set to zero; the capacities of other network edges are set to the length of corresponding mesh edges.



**Fig. 6.** Segmentation results by our approach: Green curves are the segmentation paths solved by the graph cut, and the semi-transparent yellow region is the upper teeth.

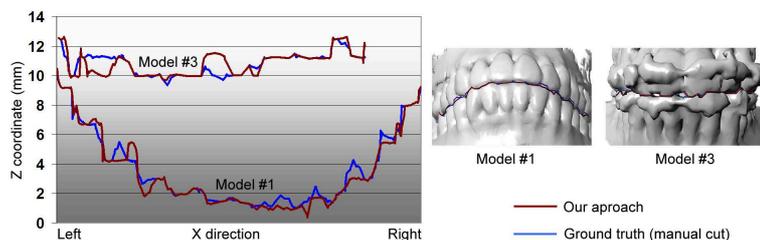
Figure 6 shows the segmentation results after the interactive segmentation path planning and the automated graph cut process are applied to a number of 3D teeth models. Due to the dental braces, many holes as well as rings and loops exist on the models. Despite of their model complexity, the combination of interactive segmentation planning (light red curves) and automated graph cut (bold green curves) is able to plausibly separate the upper and lower teeth. Note that we employ the graph cut not to improve the accuracy but to complete the segmentation, and the interactive segmentation path planning step generates the majority of the cut paths and then the graph cut fills the missing parts in the paths.

### 3 Results and Evaluation

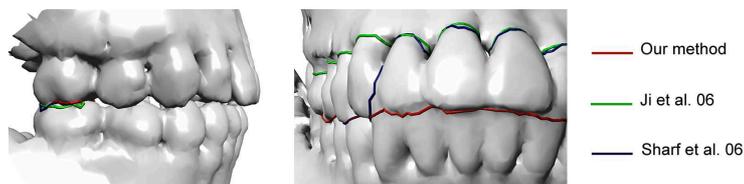
We performed two experiments including ground truth validation and comparisons between our approach and two of the state of art mesh segmentation algorithms to validate the effectiveness of our approach.

#### 3.1 Ground Truth Validation

In order to quantify the segmentation quality by our approach, we performed the following ground truth validation experiment on two datasets: a single cut (not a loop) from the leftmost to the rightmost of a teeth model, the ground truth segmentation was generated by manually selecting hundreds of segments on the mesh (219 segments for model #1 and 128 segments for model #3), and our semi-automatic approach generated the segmentation path based on several user-specified control points (3 control points for model #1, 5 control points for model #3). We projected these two segmentation paths onto Z-axis to measure their discrepancies, because Z-axis projection maximally shows the trajectory difference. Figure 7 shows the Z-coordinates of the two validations. Their maximum errors are 1.57 mm (model #1) and 1.67mm (model #3), and the average errors are 0.28 mm (model #1) and 0.30mm (model #3), respectively.



**Fig. 7.** Ground truth comparison of two segmentation paths (blue for the ground-truth, red for our approach) in the Z-axis projection.



**Fig. 8.** Segmentation comparison of model #1 between our approach (red) and the other two approaches (blue and green).

### 3.2 Comparisons With State Of The Art

We also compared our approach with two state of the art interactive mesh segmentation algorithms [4, 5]. Figure 8 shows the comparison results. As shown in this figure, our approach significantly outperformed the other two approaches. Note that the segmentation method proposed by Ji et al. [4] is able to generate smooth segmentation paths due to its refinement algorithm if the teeth only touch in small parts, but it failed to handle this complicated teeth model. In this case, its region growing scheme is uncontrollable with a small number of seeds. Therefore, it requires users to manually select a large number of seeds.

## 4 Discussion and Conclusions

In this paper we present an effective interactive technique for the upper and lower teeth segmentation. Through numerous experiments on acquired teeth CT datasets, we found that our approach is fast (e.g., tens of times faster than manual approaches) and it required minimized user interventions such as selecting several control points to guide the algorithm process. We also compared our approach with two current interactive mesh segmentation algorithms and found that our approach significantly outperformed them on segmenting complicated teeth models.

One major limitation of current approach is that when serious artifacts exist in 2D teeth CT images, our approach might fail to produce plausible segmentation results. For example, due to the strong metal artifacts of dental implant in



**Fig. 9.** Metal artifacts on the acquired 2D teeth CT images significantly affect the accuracy of segmentation by our approach.

the acquired teeth CT images and thus 3D teeth models, the segmentation accuracy of our approach would be significantly affected (Fig. 9). In the future, we plan to explore effective and automated algorithms to postprocess the acquired 2D teeth CT images and incorporate CT image segmentation with 3D geometric segmentation.

**Acknowledgments.** This research is supported by a IBIS seed grant (Xiaobo Zhou and Zhigang Deng), and Binh Huy Le is supported by a Fellowship from the Vietnam Education Foundation.

## References

1. Gregory, A.D., State, A., Lin, M.C., Manocha, D., Livingston, M.A.: Interactive surface decomposition for polyhedral morphing. *The Visual Computer* **15**(9) (1999) 453–470
2. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. *ACM Trans. on Graph.* **23**(3) (2004) 652–663
3. Lee, Y., Lee, S., Shamir, A., Cohen-Or, D., Seidel, H.P.: Intelligent mesh scissoring using 3d snakes. In: *Pacific Conference*. (2004) 279–287
4. Ji, Z., Liu, L., Chen, Z., Wang, G.: Easy mesh cutting. *Comput. Graph. Forum* **25**(3) (2006) 283–291
5. Sharf, A., Blumenkrants, M., Shamir, A., Cohen-Or, D.: Snappaste: an interactive technique for easy mesh composition. *The Visual Computer* **22**(9-11) (2006) 835–844
6. Zhao, M., Ma, L., Ta, W., Nie, D.: Interactive tooth segmentation of dental models. In: *IEEE EMBC*. (2005) 654–657
7. Gao, H., Chae, O.: Automatic tooth region separation for dental ct images. In: *IEEE International Conference on Convergence and Hybrid Information Technology*. Volume 1. (2005) 654–657
8. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: *ICCV’01*. (2001) 105–112
9. Desbrun, M., Meyer, M., Schroder, P., Barr, A.H.: Discrete differential-geometry operators in nD. Technical report, California Institute of Technology (2000)
10. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1** (1959) 269–271
11. Mitchell, J.S.B., Mount, D.M., Papadimitriou, C.H.: The discrete geodesic problem. *SIAM J. Comput.* **16**(4) (1987) 647–668