



## Medical Fixture Design with CT Images

Xueshu Liu<sup>1,2</sup>, Hiromasa Suzuki<sup>1</sup> and Yutaka Ohtake<sup>1</sup>

<sup>1</sup>The University of Tokyo, [liu\\_suzuki\\_yu-ohtake@den.rcast.u-tokyo.ac.jp](mailto:liu_suzuki_yu-ohtake@den.rcast.u-tokyo.ac.jp)

<sup>2</sup>Dalian University of Technology, [liuxs@dlut.edu.cn](mailto:liuxs@dlut.edu.cn)

### ABSTRACT

In line with the recent increased use of medical devices designed for individual patients, there is a need to ensure that the surfaces of bone fixtures match the bone surface to which they are fitted. Accordingly, this paper proposes a new process to design such fixtures from CT images using a two-stage process. The first stage involves extracting the outer bone surface of the relevant region from CT images using a level set method with a new speed function to remove the inner surface. In the second stage, a preliminary model designed using commercial CAD software is adopted to trim the extracted outer bone surface obtained in the first stage. The surface of the preliminary model to be attached to the bone is then shaped based on the trimmed outer bone surface to make the desired fixture. The results of experiments performed using the proposed method indicate its suitability for the purpose at hand.

**Keywords:** bio-CAD, level set method, CT images, outer-surface extraction.

**DOI:** 10.3722/cadaps.2011.383-394

### 1 INTRODUCTION

Mechanical CAD systems have been introduced in the medical field to design fixtures to fit individual patients. [1-5]. A common practice for designing such fixtures involves the use of shape data obtained by scanning the part of the patient in question. An example is shown in Fig. 1, which illustrates a locking plate attached to fix a bone fracture [6]. The shape of the fixture involves several surfaces, some of which must fit to the bone surface. This surface can be represented in the form of a triangular mesh generated from CT images of the bone using an iso-surfacing method such as Marching Cubes [7].



Fig. 1: Locking plate [6].

As CT images capture not only the relevant part of the bone but also the surrounding body structure, the user has to cut out the desired iso-surface to produce the required part. We cut the bone image with two planes, and the surface between them is used. These two planes are generated as an *optimal cutting plane* whose reference points (points on a plane) are given by the user and their normal vectors are determined to be as parallel as possible to the bone.

This iso-surface usually has at least two surfaces (Fig. 2(b)) - the outer surface and one or more inner surfaces. Sometimes the inner surface is connected to the outer surface (Fig. 2(c)) through small holes. In our target application, we need only the outer surface to which the fixture is to be attached, so these inner surfaces must be removed. As surfaces generated with the iso-surfacing method are generally closed, it is relatively easy to isolate the outer one by selecting the largest iso-surface that encloses the others. However, when the outer surface is connected to the inner part of the bone, this inner part of the surface must also be removed. In this paper, a level set method with a novel speed function is employed to solve the problem.

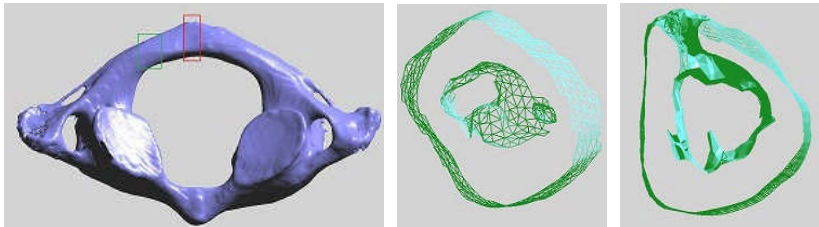


Fig. 2: CT models: (a) CT model generated by Marching Cubes [5], (b) Sectional view for the part in green rectangle, (c) Sectional view for the part in red rectangle.

Once this outer surface of the bone is isolated from CT imagery, the desired fixture model designed using commercial CAD software is used to trim the outer bone surface. It goes without saying that the fixture with this outer bone surface must fit the bone surface closely. After such a model is created, it is fed into a rapid prototyping system to produce the desired piece.

## 2 BASIC DEFINITIONS

Before describing our method in detail, we first introduce some basic representations used in the proposed approach.

- $p_i$ : the  $i$ -th voxel or grid point comprising the CT image  $V$  input to our system,  $V = \{p_i\}$ .  $\Delta g$  is the interval of the grid.
- $\rho(p_i)$ : the CT value of point  $p_i$ .
- $n(p_i)$ : the unit normal at point  $p_i$ , which is the gradient of  $\rho(p_i)$ .
- $U(p_i)$ : the 6-neighborhood of  $p_i$ .
- $B$ : the boundary grid points. We can distinguish a foreground (bone) part from the background, that is,  $\rho(p_i) - \rho_0 \geq 0$  and  $\rho(p_j) - \rho_0 < 0$  for the foreground point  $p_i$  and background point  $p_j$ , respectively (see Fig. 3).
- $\mathcal{T}$ : an iso-surface generated from  $V$  with threshold  $\rho_0$  on  $\rho$ . We use a dual contouring method to generate  $\mathcal{T}$  from  $V$ .
- $t_i$ : a triangle of  $\mathcal{T}$ .
- Using the dual contouring method [8], four triangles  $\{t_1, t_2, t_3, t_4\}$  are generated between  $p_i \in B$  and  $p_j \notin B$  (Fig. 3) as elements of  $\mathcal{T}$ . We define the grid-triangle relation  $g$  between  $p_i$  and such triangles  $\{t_1, \dots, t_{4n}\}$ , where  $n$  is the number of  $p_j \notin B$  incident to  $p_i$ . Using  $g$  makes it easy to obtain the related triangles from  $p_i$  or to obtain  $p_i$  from one of these triangles.
- $S$ :  $\mathcal{T}$  may be composed of many connected components, each of which is a two-manifold surface.  $S$  is assumed to be the largest surface that contains the others.
- $\mathcal{R}(\subset S)$ : the target outer surface of the bone to which the medical fixture will be fitted.

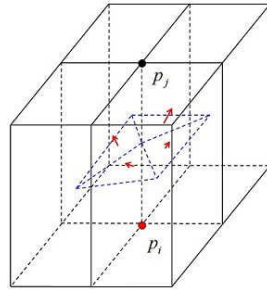


Fig. 3: Grid-triangle relation:  $p_i$  is obtained from the four triangles, or the four triangles are obtained from  $p_i$ .

### 3 BONE CUTTING

In this method, the user is requested to specify two triangles  $t_s \in S$  and  $t_e \in S$ .  $\mathcal{R}$  is defined as the region of  $S$  between two planes through these two triangles. Their planes are generated as *optimal cutting plane* [7]. Below are some definitions needed to derive the optimal cutting plane.

- $\pi(q, v)$ : a plane through a grid point  $q$  with normal  $v$ .
- $N(\pi)(\subset B)$ : the relevant neighborhood of  $\pi$ .  $N(\pi)$  is a group of grid point  $\{p_i\}$  meeting the following requirements: (1)  $p_i$  is a boundary grid point; (2) the Euclidean distance from  $p_i$  to  $\pi$  is less than the given threshold  $\delta$ ; (3)  $p_i$  belongs to the same shape part as  $q$  if  $\pi$  encompasses multiple shape parts. Accordingly, in Fig. 4, only the pink and green points are components of  $N(\pi(q, v))$ . Different colors indicate that they are on different sides of  $\pi(q, v)$ .

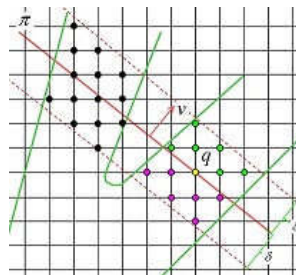


Fig. 4: Relevant neighborhood  $N(\pi)$ :  $q$  and the pink and green points are  $N(\pi)$ .

- $N^+(\pi)(\subset N(\pi))$ : the front side of  $N(\pi)$ . In Fig. 4, the green grid points and  $q$  are components of  $N^+(\pi)$ . Accordingly,  $N^-(\pi)$  indicates the back side of  $N(\pi)$ .

Let us define an optimal cutting plane, which is a plane whose normal is perpendicular to the point normal in its relevant neighborhood. We use a discrete version of the algorithm proposed in [9] to find the optimal cutting plane at point  $p$ . We define a number of candidate unit normal vectors  $\{v_i\}$ , from which we select  $v^* \in \{v_i\}$  optimizing the following equation:

$$v^* = \operatorname{argmin}_{v \in \{v_i\}} \sum_{p_j \in N(\pi(p, v))} \| \langle v, n(p_j) \rangle \| \tag{3.1}$$

The question here is how to define the candidate normal vectors  $\{v_i\}$ . The spacing of all these vectors involves a unit sphere (i.e., a Gaussian sphere) that can be parameterized by the relevant latitude and longitude angles  $(\theta, \phi)$ . Because of the symmetry involved, we consider only the hemisphere of  $\theta \leq 0, \leq 180^\circ$  and uniformly discretize these parameters to obtain sample vectors. In our experiments, we sampled these angles every  $30^\circ$ .

We compute two optimal cutting planes near  $t_s$  and  $t_e$  as given by the user. Using the grid-triangle relationship, we obtain the related boundary grid points  $p_s = g(t_s)$  and  $p_e = g(t_e)$  from these two triangles and compute two optimal cutting planes  $\pi_s = \pi(p_s, v_s^*)$  and  $\pi_e = \pi(p_e, v_e^*)$ . As shown in Fig. 5,  $\mathcal{R}$  corresponds to a region between  $\pi_s$  and  $\pi_e$ . Thus, we extract  $M(\subset B)$  between  $\pi_s$  and  $\pi_e$  with the following simple propagation algorithm:

```

Push all grid points of  $N^+(\pi_s)$  to  $h$ 
 $M := \{\}$ ;
while  $h$  is not empty do
   $x := pop(h)$ ;
   $M := M \cup \{x\}$ ;
   $y := (U(x) \cap B) \setminus (N^+(\pi_e) \cup M)$ ;
  push all the unvisited grid points of  $y$  to  $h$ .
end while

```

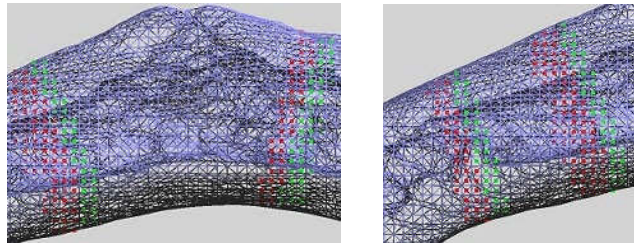


Fig. 5:  $N(\pi_s)$  and  $N(\pi_e)$  are shown with red and green grid points.  $\pi_s$  and  $\pi_e$  are located between the red and green grid points, and are generated from the left to right.

Two examples are shown in Fig. 6. The blue grid points show  $M$ , and the red and green grid points are  $N^-(\pi_s)$  and  $N^+(\pi_e)$ , respectively.

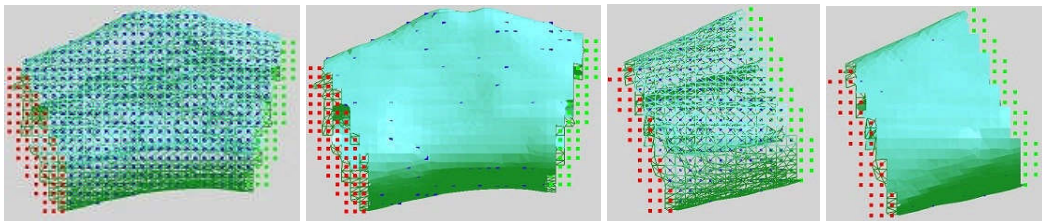


Fig. 6:  $M$ : the blue grid points are components of  $M$ , and the red and green grid points are  $N^-(\pi_s)$  and  $N^+(\pi_e)$ , respectively.

#### 4 INNER SURFACE REMOVAL

In some cases, the inner surface is connected to the outer surface through a small hole (Fig. 2(c)), while in other cases there is no connection (Fig. 2(b)). With the latter, the inner surface can be removed simply by selecting all the triangles connected to  $M$ . With the former, however, the connection hole to the inner part of the bone must be dealt with. For this purpose, an approach is proposed based on the level set method, which is a computational technique for tracking a propagating interface over time ([10, 11]). It involves embedding the shape of objects as a zero-level set of a higher dimensional surface that evolves under the influence of the surface features without violating surface regularity. As it evolves, the zero-level set contours can develop singularities and sharp corners, or the topology

itself might be changed. In order to remove inner surfaces, we generate a large zero-level set surface that contains the bone, and then evolve this surface toward the bone. The zero-level set surface must touch the bone's outer surface before it touches the inner surfaces, so its evolution is stopped at the outer surface and at any small holes in it. We also prevent the zero-level set surface from going inside the bone from its ends. In this way, the outer surface is extracted. The next section outlines a number of definitions needed to describe our method.

#### 4.1 Definitions

In the level set method, a closed surface  $\mathcal{G}(t)$  can be expressed implicitly  $\mathcal{G}(t) = \{x | \varphi(t, x) = 0\}$ , where  $\varphi(t, x)$  is a time-dependent level set function whose absolute value is given as the distance between  $x$  and  $\mathcal{G}(t)$ , and whose sign depends on whether point  $p_i$  is inside or outside  $\mathcal{G}(t)$ . Using a forward time-difference scheme,  $\mathcal{G}(t)$  can be written in an updated form.

$$\varphi(t + \Delta_t, p_i) = \varphi(t, p_i) - |\nabla\varphi(t, p_i)| F(t, p_i) \Delta_t$$

(4.1)

where  $F(t, p_i)$  is the *speed function* bound to be orthogonal to the zero-level set and can be arbitrarily defined in order to steer the evolving front propagation toward the desired shape [12, 13].  $\Delta_t$  is the time interval. For simplicity, we let  $\Delta_t = 1, |\nabla\varphi(t, p_i)| = 1$  in this paper.

The key point of applying the level set method is definition of a proper speed function so that the zero-level set will shrink and stop at the desired location. Before outlining our speed function in detail, other definitions used in this method are given.

- $\chi(p_i)$ : a binary function to indicate the location of grid point  $p_i$  in the input CT images,

$$\chi(p_i) = \begin{cases} 1, & \rho(p_i) \geq \rho_0 \\ 0, & \text{otherwise} \end{cases}$$

- $\varphi(t, p_i)$ : the signed distance from  $p_i$  to  $\mathcal{G}(t)$  at time  $t$ .
- $Z(t) \subset V$ : the boundary grid points of the zero-level set surface,

$$Z(t) = \{p_i | \varphi(t, p_i) < 0 \cap (\varphi(t, p_j) \geq 0, \exists p_j \in U(p_i))\}.$$

- $\tau(t, p_i)$ : the *waiting time* of a grid point at time  $t$ .  $\tau(t, p_i) = n\Delta_t, (t > n\Delta_t)$ , s.t.  $\varphi(t, p_i) = \varphi(t - \Delta_t, p_i) = \dots = \varphi(t - n\Delta_t, p_i) \neq \varphi(t - (n + 1)\Delta_t, p_i)$ .
- $\tau_{max}$ : the upper limit of *waiting time*.
- $D_{max}(p_i, p_j)$ : the maximum coordinate distance from  $p_i$  to  $p_j$ .

$$D_{max}(p_i, p_j) = \max \{|x_i - x_j|, |y_i - y_j|, |z_i - z_j|\}$$

where  $x_i, y_i, z_i$  denote the  $x, y, z$  coordinates of grid point  $p_i$ .

- $W(t, p_i)$ : the neighborhood of boundary grid point  $p_i \in Z(t)$  at time  $t$ .  $W(t, p_i) = \{p_j \in Z(t) | \varphi(t, p_i)\varphi(t, p_j) > 0, D_{max}(p_i, p_j) \leq \left\lceil \frac{\delta}{\Delta_g} \right\rceil\}$ , where  $\delta$  is the approximate diameter of the small holes. A 2D example is shown in Fig. 7. The red points enclosed in green triangles are  $W(t, p_i)$  in the case of  $\left\lceil \frac{\delta}{\Delta_g} \right\rceil = 4$ .
- $\widehat{W}(t, p_i)$ : a support consisting of a subset of  $W(t, p_i)$  satisfying two conditions:  $\chi(p_j) = \chi(p_i)$  and  $n(p_j) \cdot n(p_i) > \alpha$ , where  $\alpha$  is a given threshold.
- $\lambda(t, p_i)$ : the support ratio for boundary grid point  $p_i \in Z(t)$  at time  $t$ .

$$\lambda(t, p_i) \triangleq \frac{|\widehat{W}(t, p_i)|}{|W(t, p_i)|}$$

where  $|\cdot|$  is the number of elements in a set.

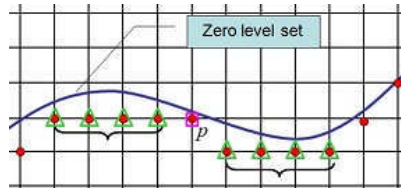


Fig. 7:  $W(t, p_i)$ : the red grid points enclosed in green triangles are components.

## 4.2 Initialization

Usually, we initialize the zero-level set surface for  $\mathcal{G}(t)$  at  $t=0$  to be like a sphere involving the whole of  $M$  (the blue grid points in Fig. 6). We compute an axis-aligned bounding box for  $M$  and use its center  $c$  as that of the sphere. The radius  $r$  is estimated as half of the longest diagonal dimension of the box. We initialize the level set as  $\varphi(0, p_i) = \text{dis}(p_i, c) - r$ , where  $\text{dis}$  denotes the Euclidean distance from  $p_i$  to  $c$ .

In addition, the waiting time for all grid points is initialized as zero, i.e.,  $\tau(0, p_i) = 0, \forall p_i$ .

## 4.3 Speed Function

We let  $\mathcal{G}(t)$  shrink toward the bone and finally stop at the outer surface. This shrinking can be performed by increasing the signed distance  $\varphi(p_i)$  over time. The speed function is given as outlined below.

Loop the following procedure for  $t = 0, \Delta_t, 2\Delta_t, \dots$  until the zero-level set stops moving.

**for** each grid point  $p_i \notin Z(t)$  **do**

**if**  $-\gamma \leq \varphi(t, p_i) < 0$  **then**

$F(t, p_i) = 0$

**else**

$F(t, p_i) = \gamma$

**end if**

**end for**

where  $\gamma$  is a positive constant.

In order to maintain the topology of the zero-level set surface, care must be taken to avoid holes. The distance function is generally increased at every time step by  $\gamma$ , so points inside the zero-level set with negative  $\varphi$  can develop positive values over time. Around these positive points, holes are generated in the zero-level set. Accordingly, it is necessary to prevent these negative points from becoming positive by not increasing the signed distance value, i.e., by setting zero speed.

To make sure the zero-level set surface never touches interior surfaces without first passing through the outer surface, it must be ensured that the zero-level set surface stops at the ends of the cut bone.

We extend  $N^-(\pi_s)$  to  $C_s$  in order to cap the ends of the bone using the following algorithm:

$H :=$  the axis-aligned bounding box of  $N^-(\pi_s)$ ;

Push all grid points of  $N^-(\pi_s)$  to  $h$

$C_s := \{\}$ ;

**while**  $h$  is not empty **do**

$x := \text{pop}(h)$ ;

$C_s := C_s \cup \{x\}$ ;

$y := U(x) \cap H \cap (-\delta < \text{dis}(U(x), \pi_e) < 0) \setminus C_s$ ;

    push all the unvisited grid points of  $y$  to  $h$ .

**end while**

An example is shown in Fig. 8. The blue curve represents the outer surface, and the green curve is the inner surface.  $C_s$  contains all grid points enclosed by the bounding box obtained with  $N^-(\pi_s)$ , including the red, black and green grid points in Fig. 8. Thus,  $C_e$  is defined.

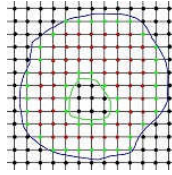


Fig. 8:  $C_s$ : All grid points, including the red, black and green ones, are components of  $C_s$ .

In each time step, the following procedure is used to update the speed function and waiting time:

**for**  $p_i \in Z(t) \cap \{C_s \cup C_e\}$  **do**

$F(t, p_i) = 0;$

$\tau(t, p_i) = 0;$

**end for**

**for**  $p_i \in Z(t) \setminus \{C_s \cup C_e\}$  **do**

**if**  $\lambda(t, p_i) \geq \frac{1}{4}$  **then**

$$F(t, p_i) = \begin{cases} \gamma; & \text{for } \chi(p_i) = 0 \\ 0; & \text{for } \chi(p_i) = 1 \end{cases}$$

$\tau(t + \Delta_t, p_i) = 0;$

**else**

**if**  $\tau(t, p_i) < \tau_{max}$  **then**

$F(t, p_i) = 0;$

$\tau(t + \Delta_t, p_i) = \tau(t, p_i) + \Delta_t;$

**else**

$$F(t, p_i) = \begin{cases} \gamma; & \text{for } \chi(p_i) = 1 \\ 0; & \text{for } \chi(p_i) = 0 \end{cases}$$

$\tau(t + \Delta_t, p_i) = 0;$

**end if**

**end if**

**end for**

In Fig. 9, suppose  $W=6$ . The candidates are shown as red circles enclosed in black squares, and their normals are shown with light blue arrows. The only boundary grid points meeting the conditions outlined here are those from *support* to  $p_i$ . In this example, suppose that  $\alpha=0.6$  and we know only that the boundary grid point on the right side of  $p_i$  is the support to  $p_i$ , so  $|\widehat{W}(t, p_i)| = 1$ . Thus, we obtain  $\lambda(t, p_i) = \frac{1}{6}$  for this situation. According to the speed function, the level set will cut off part of the feature. In other words, the zero-level set will not stop at the current location.

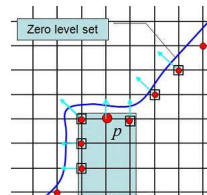


Fig. 9:  $(t, p_i)$  : The normal is one of the most important parameters considered in the calculation.



Before calculation, the user should define  $W$  and  $\alpha$  according to the model. In Fig. 9, suppose the light blue part is a narrow bridge. We want the level set to stop at the current location. To this end, we let  $|W(t,p)|=4$  and  $\alpha=0.4$ , and we then have  $\lambda(t,p_i) = \frac{3}{4}$ . Considering the speed function, we know the level set will stop there as it receives *enough support*.

*Waiting time* is needed here because different parts of the level-set surface touch the boundary of the input model at different times. According to the speed function, we know that when the level set touches the corner of the model, the support ratio is very low (the green points in Fig. 10(a)). As a result, the level set goes into the input model (the blue curve and the dashed curve for the next iteration), which is not desirable. After introducing waiting time to the speed function, the result is different; the level set stops at the boundary of the input model (the red curve in Fig. 10(b)).

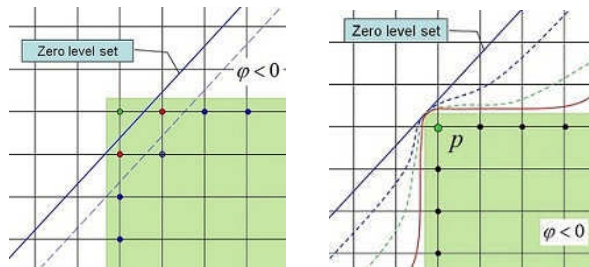


Fig. 10: Effect of waiting time: (a) without waiting time. (b) with waiting time.

There are several constants to be determined here.

$\gamma$ : To maintain the stability of the level set method, a CFL condition that requires the zero-level set surface to cross no more than one grid cell at each time step must be followed. Thus, we can ascertain the speed using the following equation:

$$(4.2) \quad \gamma \leq \frac{\Delta_g}{\Delta_t}$$

$\tau_{max}$  can be calculated using the following equation:

$$(4.3) \quad \tau_{max} = \left\lceil \frac{\delta}{\gamma} \right\rceil$$

The procedure for the proposed method is shown in Fig. 11. We extract the bone part using the method outlined in the previous section (Fig. 11(a)), and the inside view may be as shown in Fig. 11(b). We generate the zero-level set (the yellow triangles in Fig. 11(c)). It is not necessary for the zero-level set surface to be perfectly spherical; the only requirement is that it should contain the whole of the bone part. The surface is then made to evolve using the proposed speed function. After the zero-level set stops moving, we may have a zero-level set surface such as that shown in Fig. 11(d). It should be noted that the zero-level set is different from the iso-surface of the bone, which is why parts of the zero-level set surface are inside the bone while others are outside it. However, none of the zero-level set surface touches the inner surface. Next, we need to obtain the outer surface of the bone part.



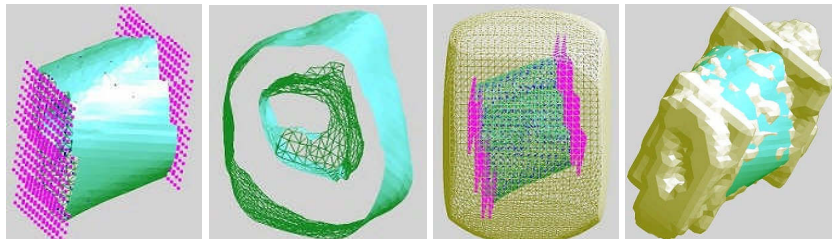


Fig. 11: Procedure of inner-surface removal: (a) Bone part; (b) Inside view; (c) Zero-level set surface; (d) Final zero-level set surface.

### 5 HOLE FILLING

Using the algorithm presented in the previous section, we can successfully remove the inner surface as per the experimental results shown in Fig. 12. However, if the inner surface is connected to the outer surface, a hole is generated as shown in Fig. 12(b).

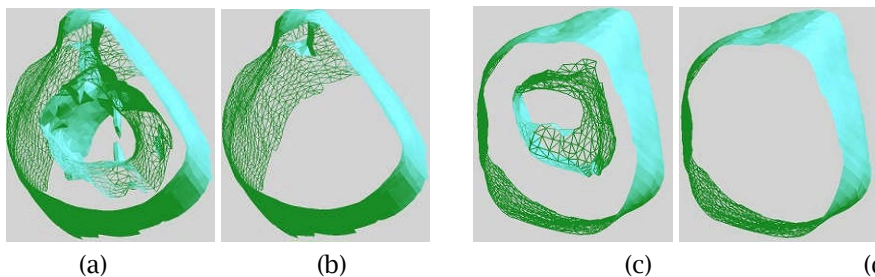


Fig. 12: Examples of inner-surface removal: (a) Before inner surface removal; (b) After removal; (c) Before inner surface removal; (d) After removal.

Hereafter, we use  $Z$  to denote the zero-level set obtained in the previous section. As  $Z$  contains all the boundary grid points related to the zero-level set surface, we define a subset  $\hat{Z}$  of  $Z$  as a part located between  $C_s$  and  $C_e$  by applying the method used to extract  $M$  from  $B$  described in Section 3. That is, we apply the method to  $Z$  instead of  $B$ . A 2D example is shown in Fig. 13. In this example,  $C_s$  and  $C_e$  are shown with green curves. The green grid points surrounded by pink squares or triangles are  $Z$ . Those enclosed by pink squares are  $\hat{Z}$ . The blue dashed line indicates the surface of the bone. The green grid points are  $M$ , the black ones are  $\rho(p_i) < \rho_0$ , and the red ones are  $\rho(p_i) \geq \rho_0$ .

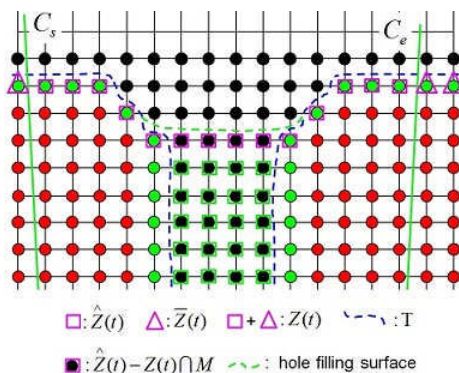


Fig. 13: Bone with hole: relationship between zero-level set surface and the bone surface.

By comparing  $Z$  and  $M$ , it can be seen that a difference exists where the hole is located because  $Z$  stopped at the entrance of the hole. In Fig. 13, the black grid points enclosed in pink squares correspond to the hole. We define this hole region as  $H$  and its surrounding region as  $K$ , that is,  $H = \hat{Z} - K$  and  $K = \hat{Z} \cap M$ . In Fig. 13,  $H$  is indicated by the black grid points enclosed in pink squares. If the bone has no hole,  $H$  is empty. In Fig. 14(a) and Fig. 14(b),  $H$  is shown by blue grid points. In order to fill the hole of  $H$ , we change the CT values of the points so that the iso-surface generated later has no hole around  $H$  (see the green dashed curve in Fig. 13). This change is made through a two-step process. In the first step, we change the CT values of the points in  $H$  (the black grid points enclosed in pink squares). For each  $p_i \in H$ ,  $\rho(p_i) = \rho(p_j)$ , where  $p_j$  is the nearest point in  $K$  to  $p_i$ .

Next, we change the CT values of points that lie outside the object but inside  $Z$ . These are shown by the black grid points enclosed in green squares in Fig. 13. These points  $p_i$  satisfy the two conditions of  $\rho(p_i) < \rho_0$  and  $\varphi(p_i) < 0$ . We change their CT values to a constant  $\rho_1 \geq \rho_0$ .

The desired outer surface, consisting of triangles related to grid points  $\hat{Z}$ , can be obtained using the grid-triangle relationship  $g$ . That is, the surface is defined by  $\mathcal{R} = \{t_i \in \mathcal{T} | g^{-1}(t_i) \in \hat{Z}\}$ . The triangular mesh for  $\mathcal{R}$  is obtained using a dual contouring method with the adapted CT values. The experimental results are shown in Fig. 14(c) and Fig. 14(d). In this example, the hole is filled and no extra iso-surface is generated.

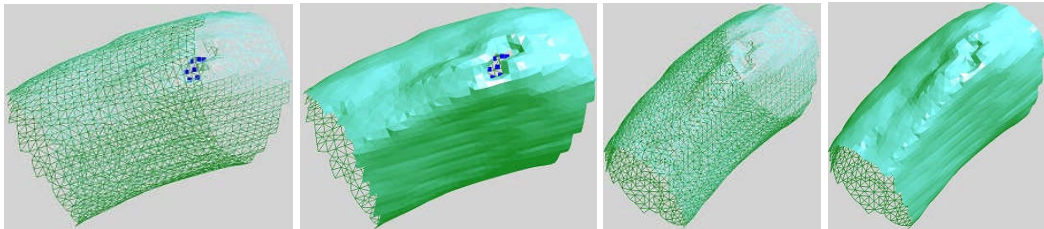


Fig. 14: Examples of hole filling. Boundary grid points are shown in blue.

## 6 FIXTURE DESIGN

After the desired bone surface is obtained, we design 3D solid model of a fixture as shown in Fig. 15. The bottom surface of this fixture must fit the bone surface, so we trim the bone surface to use as a surface of the fixture model. This procedure consists of two steps. In the first step, we design the fixture using commercial CAD software. An example is shown in Fig. 15(a). In the preliminary design, the two main surfaces of the piece are flat, and the model has a uniform thickness. One of these two surfaces will be attached to the bone. A common problem with fixtures designed using commercial software is that flat surfaces do not fit bone surfaces well, as the latter type of surface is hardly ever planar (see Fig. 2(a)). Even if the fixture's main surface is modified to include curvature, the designer cannot guarantee a close fit because nothing is known about the bone surface. Accordingly, in the second stage we trim the outer bone to make it a main surface so that the fixture will fit closely.

To trim the outer bone surface, as part of design improvement we import the preliminary model and adjust its posture to meet our requirements. This is done through a series of simple operations such as movement and rotation. An example is shown in Fig. 15(b). After the boundary vertices of the preliminary model have been obtained and projected onto the outer bone surface, we complete this trimmed bone surface through re-triangulation (Fig. 15(c)). After making the side surface between the trimmed bone surface and the preserved flat surface of the preliminary model (Fig. 15(d)), we obtain the desired fixture that will closely fit the bone surface. Some experimental results are outlined in the next section.

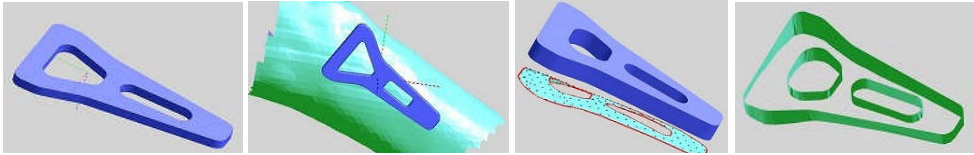


Fig. 15: Design improvement. (a) Preliminary design; (b) Position adjustment; (c) Trimming of outer surface; (d) Side surface generation.

## 7 EXPERIMENTAL RESULTS

The method proposed in this paper was applied to obtain experimental results. CT models of two different bone parts are shown in Fig. 16(a) and Fig. 16(c). The first model is a pipe-like bone, in which the middle part is usually higher than the other parts. However, for the middle part of second model, the right part is lower than the left part. With the proposed method, two locking plates are obtained as shown in Fig. 16(b) and Fig. 16(d). The model in Fig. 16(b) has a different thickness in line with the shape of the bone. This is because one of the main surfaces is flat and variant thickness makes sure that this kind of fixtures can be made to fit the bone closely. Surely, the thickness can be controlled as required (Fig. 16(d)) and can also be made uniform by re-meshing the mesh model. Compared with the model shown in Fig. 1, fixtures obtained with the proposed method can fit the bone better as the main surface that fit to the bone is curved face. Figure 16(e) is a mesh model representation of Fig. 16(d). After we export the model, we can make the fixture for the patient using the rapid prototyping technique. Figure 16(f) is the actual model of the image shown in Fig. 16(d).

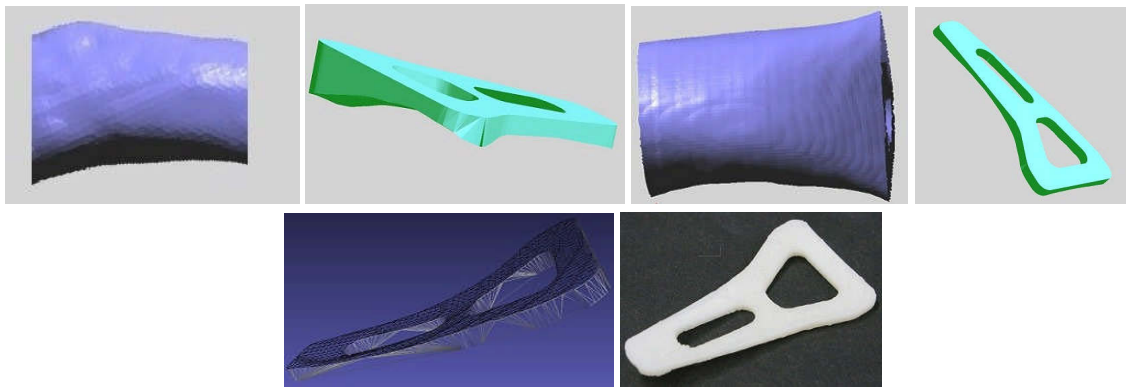


Fig. 16: Experimental results. (a) Bone model; (b) Fixture of (a); (c) Bone model; (d) Fixture of (c); (e) Mesh model of (d); (f) Real model of (d) by RP.

## 8 CONCLUSION

In this paper, we propose a new process to design medical fixtures from CT images. In the first stage, an approach to extract the outer surface from CT images is introduced. First, two optimal cutting planes are generated based on the user's selection, and then the desired bone part between the two optimal cutting planes is obtained. To remove the inner surface, a novel speed function is proposed. Finally, after filling any holes, we obtain the desired outer surface of the bone. In the second stage, we use the extracted bone surface to design the surface of the fixture to be attached to it. Through a series of simple operations, we obtain the trimmed bone surface and use it to replace the surface of the preliminary model that will be attached to the bone to create the desired fixture. The results of

experimentation showed that the proposed technique can be used to produce individual fixtures that fit the bone closely.

The method offers many opportunities for improvement and modification. For inner-surface removal, although in most cases the removal level set surface shares the same related boundary grid points with the outer surface; this is not always the case for highly concave surfaces. Accordingly, a new way to utilize the removal level set surface should be introduced to make the approach more robust. In addition, although the outer surface is extracted from CT images, its edges are rough. To combat this, a smoothing method should be introduced to optimize design. Moreover, so far there are no experimental results indicating that the proposed method can be used to detect narrow connections. In the future, we will apply the method to other models to clarify the matter.

## REFERENCES

- [1] Xuesheng Yuan; Qingxi Hu; Hangqiang Liu; Chunxiang Dai; Minglun Fang: Modeling technology and application of repairing bone defects based on rapid prototyping, International Federation for information Processing(IFIP), 207, 2006, 643-649, DOI: 10.1007/0-387-34403-9\_89.
- [2] W. Sun; B. Starly; J. Nam; A. Darling: Bio-CAD modeling and its applications in computer-aided tissue engineering, Computer-Aided Design, 37, 2005, 1097-1114, DOI:10.1016/j.cad.2005.02.002.
- [3] W. Sun; B. Starly; A. Darling; C. Gomez: Computer-aided tissue engineering: application to biomimetic modeling and design of tissue scaffolds, J. Biotechnol. Appl. Biochem. 39(1), 2004, 49-58, DOI:10.1042/BA20030109.
- [4] Kimberly J. Nielsen; Jordan J. Cox: Implementation of biomechanical mating conditions in CAD, Computer-Aided Design and Applications, 5(1-4), 2008, 338-353, DOI:10.3722/cadaps.2008.338-353.
- [5] Charlie C.L. Wang; Kai Tang: Woven model based geometric design of elastic medical braces, Computer-Aided Design, 39, 2007, 69-79, DOI:10.1016/j.cad.2006.10.001.
- [6] ARATA Co. Ltd, [http://www.arata-med.co.jp/lineup/distal\\_w.html](http://www.arata-med.co.jp/lineup/distal_w.html).
- [7] Leif P. Kobbelt; Mario Botsch; Ulrich Schwanecke; Hans-Peter Seidel: Feature sensitive surface extraction from volume data, Proceedings of ACM. SIGGRAPH 2001, LosAngels, USA, 2001, 57-66, DOI: 10.1145/383259.383265.
- [8] Tao Ju; Frank Losasso; Scott Schaefer; Joe Warren: Dual contouring of hermite data, Proceedings of ACM. SIGGRAPH 2002, San Antonio, USA, 2002, 339-346, DOI: 10.1145/566654.566586.
- [9] Andrea Tagliasacchi; Hao Zhang; Daniel Cohe-or: Curve skeleton extraction from incomplete point cloud, Proceedings of ACM. SIGGRAPH 2009, New Orleans, USA, 2009, DOI: 10.1145/1531326.1531377.
- [10] J. A. Sethian: Level set methods and fast marching methods, Cambridge University Press, Cambridge, second edition, 1999.
- [11] Stanley Osher; Ronald Fedkiw: Level set methods and dynamic implicit surface, Springer-Verlag, New York, 2003.
- [12] Hongyu Lu; Chaijie Duan; Shanglian Bao: Accurate segmentation of image with large gaps by geometric active contours, ICSP 2008, 2008, 1235-1238, DOI: 10.1109/ICOSP.2008.4697354.
- [13] Chieh-Ling Huang: Shape-based level set method for image segmentation, 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009, 243-246, DOI: <http://doi.ieeecomputersociety.org/10.1109/HIS.2009.55>.