VOLUME GRAPH MODEL FOR 3D FACIAL SURFACE EXTRACTION

Lei Wu¹, Houqiang Li¹, Nenghai Yu¹, Mingjing Li²

¹MOE-Microsoft Key Lab of MCC and Department of EEIS
University of Science and Technology of China, Hefei 230027, China
²Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China

ABSTRACT

3D facial extraction from volume data is very helpful in virtual plastic surgery. Although the traditional Marching Cubes algorithm (MC) can be used for this purpose, it could not separate the facial surface from other tissue surfaces. This weakness greatly limited the accuracy of the facial model and its application in plastic surgery. In this paper a volume graph model is proposed, in which the facial surface extraction is formulated as a min-cut problem and can be solved by existing graph cut algorithm. Based on this model, irrelevant tissue surfaces are effectively excluded and a more accurate 3D virtual face can be built for plastic surgery.

1. INTRODUCTION

In order to build an accurate 3D facial model for virtual plastic surgery [15], facial surfaces should be extracted from magnetic resonance images (MRI) or other volume data. However, this process is nontrivial. The main issue encountered is how to extract the facial surfaces accurately while exclude that of other surrounding tissues.

Great efforts have been made to solve the problem of 3D surface extraction from volumetric representations [3][4][6][12][13]. Among various methods, Marching Cubes algorithm (MC) [8] is the most commonly used surface extraction method in medical visualization. To improve the performance of MC, some variants of this basic algorithm have been proposed [11][10][9], such as the octree technique [7], feature sensitive surface extraction [6], etc. However, all of those methods are unable to separate facial surfaces from other tissue surfaces in the volume data successfully.

Volume data is usually organized as a 3D array, in which each element is called a voxel and contains the gray information of corresponding tissue. In MC algorithm, the 3D volume data is divided into little cubes of \( n \times n \times n \) voxels. In each cube, the voxels at the corner are called vertexes. First, object vertexes are separated from the background vertexes by a proper threshold of gray values. Then, the cube is classified into one of 15 predefined modes by the number and position of its object vertexes. For each mode of the cube, a certain triangulation method is provided. After all cubes are triangulated, the object’s surfaces are formed by the triangles in these cubes as shown in Fig.1 (a).

However, this algorithm cannot separate facial surface with surrounding tissues. Because it will ransack surfaces throughout the volume data, parts of the irrelevant tissue surfaces inside the head are also extracted. An example is illustrated in Fig.1 (b).

To overcome the weakness of MC, we propose a volume graph model for 3D face extraction, in which the facial extraction task is formulated as a min-cut problem in the graph theory. This enables us to use the existing graph cut approach to solve the problem. In this way 3D face surface can be accurately extracted without irrelevant tissues.

The rest of this paper is organized as follows. In Section 2, the graph cut theory is briefly overviewed. In Section 3, the proposed volume graph model is explained in detail. The comparison with other approaches is provided in Section 4. Finally, the conclusion is drawn in Section 5.

2. OVERVIEW OF THE GRAPH CUT APPROACH

Graph cut approach [14][12] is an optimization method to locate the global minimal cut in a graph [5]. It derives from graph theory and changes the s-t minimum cut problem, which is NP problem, into the max-flow problem that can be solved by the existing max-flow algorithm [1].

2.1. Minimum Cut Problem

In graph theory, a graph is defined as \( G(V,E) \), \( V = \{V_1,V_2,\ldots,V_n\} \), \( E = \{E_{ij}\} \), \( C = \{c_{ij}\} \), \( i,j = 1,2,\ldots,n \). \( V \) is the graph node set and \( E \) is the set of graph edges which connect
nodes. $E_{ij}$ represents the edge connecting node $V_i$ and node $V_j$ directly. $C_{ij} = C(V_i, V_j)$ is the capacity of edge $E_{ij}$, which evaluates the connection. This theory is first proposed to solve the problem of network flow, in which graph nodes represent the sites, and graph edges denote the connections between these sites. Capacity corresponds to the bandwidth on each connection. Then the cuts are defined as disabled connections in the network. The cuts capacity is evaluated by the loss of bandwidth on the cuts. The capacity of min-cuts is the least loss of bandwidth in the network to disconnect site $S$ to site $T$. The min-cuts problem can be described as follows.

$$\forall S, T \subset V, S \cap T = \emptyset,$$

$$\text{mincut}(S, T) = \min_i (\text{Cut}(A_i, \bar{A}_i))$$

s.t. $S \subseteq A_i, T \subseteq \bar{A}_i, i = 0, 1, \ldots$

$$\text{cut}(A, \bar{A}) = \sum_{V_i \in A, V_j \in \bar{A}} C_{ij} = \sum_{V_i \in A, V_j \in \bar{A}} C(V_i, V_j)$$

$A_t$ is a subset of node set $V$ and $\bar{A}_t$ consists of the rest of the nodes. The cut in a graph is defined as the sum of capacity values of cutting edges, removing which the graph is split into two disconnected sub-graphs. In s-t minimum cut problem, the goal is to find the smallest cut in graph $G$ that can cut off all paths from root set $S$ to root set $T$.

### 2.2. Maximum Flow Problem

A flow $F$ is defined as a certain kind of value that can be transported from a source $s \in V$ to a sink $t \in V$. The sum of flow infusing a node must equal the sum of flow that goes out of it. The flow on each edge must be less than or equal to the capacity of the edge. An edge on which the flow is equal to the capacity is called saturated. A maximum flow is defined as the maximized flow through the $s \rightarrow t$ path. According to Ford-Fulkerson theorem [1], the search of maximum flow from vertex $s$ to vertex $t$ equals to the extraction of the minimum cuts separating $s$ and $t$, for the flow saturated lies uniformly on the minimum cut. So the min-cut problem can be solved by the existing max-flow algorithm [5]. One of most general algorithms for max-flow problem is the Augmenting paths algorithm:

1. **Step 1**: Initialization. Set Flow to zero on each edge;
2. **Step 2**: Search for an S-T path along which more flow may be pushed;
3. **Step 3**: If no such path exists, finish; otherwise, increase the flow uniformly along this path until at least one edge becomes saturated.
4. **Step 4**: Repeat Step 2 and 3 till finish.

Ford and Fulkerson proved that a maximum flow will be obtained when this algorithm converges. Minimum cuts are the saturated edges in the max flow search process.

### 3. VOLUME GRAPH MODEL

The volume graph model is represented as $G(V, E, C, \text{Cut})$, where $V$ represents the set of voxels in the volume data, and $E$ represents the edge set. We assume that there are edge connections $e \in E$ between adjacent voxels. Nonadjacent voxels $s$ and $t$ are not directly connected to each other but through a chain of edges. These chain edges form a path $p(s, t)$. Each edge is assigned a capacity $C_{ij}$ which indicates the likelihood of the existence of object surface. Cut represents a subset of edges, cutting off which the graph can be divided into two separate sub-graphs. In this paper, we define the capacity function as Eq. (3).

$$C_{ij} = \frac{1}{1 + (G_{\sigma} * \nabla l(e_{ij}))^2}, e_{ij} \in E$$

$$\forall l(e_{ij}) = V_t - V_s$$

In Eq. (3), $G_{\sigma}$ is the Gaussian kernel function with deviation $\sigma$, while $\alpha$ is a constant parameter. $\nabla l(e_{ij})$ is the difference of gray values between the adjacent voxels. The more likely a surface exists between the two voxels, the less capacity will be assigned to the edge. The function $C_{ij}$ drops to the minimum when $e_{ij}$ lies on the object’s surfaces.

In the volume graph model, a closed surface $B$ divides the graph into inside part and outside part. This surface is represented as a set of edges which form the connection between inside voxels and outside voxels. The capacity of surface $B$ is defined as the sum of capacities of all edges in it.

$$f(B) = \sum_{e \in B} \frac{1}{(1 + (G_{\sigma} * \nabla l(e))^2)} B \subset E$$

As the definition of capacity function has determined the nearer to the surface the smaller capacity will be, the min-cuts of the volume graph will be the object surface $B^*$.

$$f(B^*) = \min \left( \sum_{e \in C_{ij}} f(e_{ij}) \right)$$

$$= \min \{ \text{cut}(A_t, \bar{A}_t), C \in \text{Cut} \} \quad (6)$$

$A_t$ is the subset of the voxel inside cut $C$, and $\bar{A}_t$ consists of the voxels outside. In order to prevent extraction of other tissue surfaces, the search of min-cut will be constrained inside a closed shell $B_t$, defined in Eq. (7). Initially, the shell should be placed at any place outside the object in the volume data where no other tissue exists. However, the position of the shell can affect the search time of surfaces. The nearer to the object surface, the faster the search can be finished.

$$\exists B_t \subset V, \forall v_k \in B_t, s.t. \| v_k - C_{ij} \| \leq \epsilon \quad (7)$$

$C_{ij}$ is the minimum cuts candidate on the graph in i-th iteration. $C_{ij}^{\partial}$ is defined as a random cuts near the facial surface. $\| \| \|$ represents the distance measurement. $B_t$ is not the facial surface but an approximate initial shell with predefined thickness $\epsilon$, which covers $C_{ij}^{\partial}$ and divides the volume into approximate background region $T_0$, shell $B_t$ and approximate object region $S_0$. So far, the shell may not
contain the real facial surface. Then the local min-cuts will be searched inside the shell $B_i$. When the local min-cuts are found inside the shell, the shell will be deformed according to the min-cuts shape with the invariant thickness $\varepsilon$. As the definition of the capacity function has determined that the nearer to the facial surface the smaller cuts will be, capacity will lead the local min-cuts moving towards the facial surface and the shell will deform accordingly. When the facial surface is covered by the shell, the min-cuts will be the facial surface and the shell will not invade into other tissue across facial surface. Because it is searching for min-cuts, the more surface is covered, the smaller capacity will be. If it moves across the facial surface, few surfaces will be covered and the capacity will rise. The whole process can be described into the following steps.

Step 1: Set initial shall $B_0 \Rightarrow S_0, T_0$
Step 2: For the $i$ th iteration, find $C_{i\min}$ inside $B_i$
Step 3: form new Shell $B_{i+1}$ from $C_{i\min}$, s.t. $\forall v_k \in B_{i+1}, \|v_k - C_{i\min}\| \leq \varepsilon$
Step 4: if $B_{i+1}$ equals $B_i$, then $B_i$ is the object surface and return; otherwise go to step 2.

4. EXPERIMENT

We conduct the comparison between MC and VGM on the common MRI dataset of half human head consisting of 138 colored MRI each with size of 600*800, which is widely used as the test bed for evaluation of MRI processing algorithms.

4.1. Experimental Setting

In order to evaluate both approaches, we compared the precision and recall of both extraction results. The manually marked facial surface is used as the ground truth. These measurements are defined as follows.

$$\text{precision} = \frac{T_{p+}}{(T_{p+} + F_{+})}$$

(8)

$$\text{recall} = \frac{T_{r+}}{(T_{r+} + T_{-})}$$

(9)

As tiny distortion from the manual marking is allowed, we define a threshold on the minimum distance between an extracted pixel and a manually marked pixel to judge whether the extracted pixel is positive or negative. If the distance between an automatically extracted pixel and its nearest marked pixel is smaller than a predefined threshold, the extracted pixel is classified into truth positive $T_{p+}$; otherwise false positive $F_{+}$. On the contrary, if the distance between a manually marked pixel and its nearest extracted pixel is smaller than the threshold, the marked pixel is classified into true positive $T_{r+}$; otherwise true negative $T_{-}$.

In the evaluation, we calculate precision, recall of these two methods under 10 different thresholds, from 1 pixel to 10 pixel interval. By using the threshold, we can better compare the two 3D extraction algorithms with the allowance of tiny distortion from the manual marking. Otherwise the comparison makes little sense, as shown in Table 1 when the threshold is set to one, because distortion-free surface extraction is unavailable.

4.2. Experimental Evaluation

We compare the results for the proposed Volume Graph Model (VGM) based approach with the MC approach. As MC algorithm is sensitive to parameters, the results turn out to be variant according to the size of cube $n \times n \times n$. To find the best performance of MC, we tried four different sizes ($n=2, 4, 8, 16$) notated as MC2, MC4, MC8, MC16. The comparisons under different thresholds are listed in Table 1 and Table 2.

Fig.2 illustrates the precision and recall of the two approaches. We can see that both VGM and MC perform better if we loosen the threshold. VGM has statistically better performance in precision and recall when the threshold is set under 5, above which the comparison has little meaning. The reason for MC’s bad precision is somehow due to too many irrelevant surface voxels that it extracts from the volume data. The number of irrelevant voxels in MC will increase when using smaller cube size. So the smaller the cube size is, the lower the precision will be and the more facial points will be extracted, which results higher recall.

Table 1: Precision comparison under ten thresholds

<table>
<thead>
<tr>
<th>Threshold</th>
<th>MC16</th>
<th>MC8</th>
<th>MC4</th>
<th>MC2</th>
<th>VGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.82</td>
<td>3.95</td>
<td>1.46</td>
<td>1.01</td>
<td>18.25</td>
</tr>
<tr>
<td>2</td>
<td>22.20</td>
<td>14.82</td>
<td>7.48</td>
<td>5.61</td>
<td>78.97</td>
</tr>
<tr>
<td>3</td>
<td>33.20</td>
<td>26.99</td>
<td>18.88</td>
<td>15.63</td>
<td>95.34</td>
</tr>
<tr>
<td>4</td>
<td>42.38</td>
<td>37.69</td>
<td>29.12</td>
<td>24.26</td>
<td>97.44</td>
</tr>
<tr>
<td>5</td>
<td>48.76</td>
<td>43.21</td>
<td>33.68</td>
<td>28.35</td>
<td>98.19</td>
</tr>
<tr>
<td>6</td>
<td>52.01</td>
<td>45.49</td>
<td>35.54</td>
<td>30.02</td>
<td>98.50</td>
</tr>
<tr>
<td>7</td>
<td>53.46</td>
<td>46.76</td>
<td>36.37</td>
<td>30.76</td>
<td>98.66</td>
</tr>
<tr>
<td>8</td>
<td>54.41</td>
<td>47.46</td>
<td>37.02</td>
<td>31.45</td>
<td>98.77</td>
</tr>
<tr>
<td>9</td>
<td>55.61</td>
<td>48.12</td>
<td>37.78</td>
<td>32.31</td>
<td>98.85</td>
</tr>
<tr>
<td>10</td>
<td>56.45</td>
<td>49.80</td>
<td>38.80</td>
<td>33.39</td>
<td>98.91</td>
</tr>
</tbody>
</table>

Table 2: Recall comparison under ten thresholds

<table>
<thead>
<tr>
<th>Threshold</th>
<th>MC16</th>
<th>MC8</th>
<th>MC4</th>
<th>MC2</th>
<th>VGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.48</td>
<td>1.26</td>
<td>2.96</td>
<td>9.22</td>
<td>29.72</td>
</tr>
<tr>
<td>2</td>
<td>3.31</td>
<td>10.42</td>
<td>24.71</td>
<td>29.94</td>
<td>89.79</td>
</tr>
<tr>
<td>3</td>
<td>10.03</td>
<td>32.51</td>
<td>48.01</td>
<td>53.49</td>
<td>96.01</td>
</tr>
<tr>
<td>4</td>
<td>21.28</td>
<td>65.63</td>
<td>74.26</td>
<td>80.68</td>
<td>97.16</td>
</tr>
<tr>
<td>5</td>
<td>36.92</td>
<td>82.10</td>
<td>93.30</td>
<td>97.48</td>
<td>97.61</td>
</tr>
<tr>
<td>6</td>
<td>55.20</td>
<td>91.75</td>
<td>98.26</td>
<td>99.63</td>
<td>97.78</td>
</tr>
<tr>
<td>7</td>
<td>73.41</td>
<td>95.64</td>
<td>99.12</td>
<td>99.94</td>
<td>97.89</td>
</tr>
<tr>
<td>8</td>
<td>91.33</td>
<td>97.65</td>
<td>99.68</td>
<td>99.97</td>
<td>97.96</td>
</tr>
<tr>
<td>9</td>
<td>94.06</td>
<td>98.51</td>
<td>99.89</td>
<td>100</td>
<td>98.03</td>
</tr>
<tr>
<td>10</td>
<td>94.55</td>
<td>99.10</td>
<td>99.96</td>
<td>100</td>
<td>98.11</td>
</tr>
</tbody>
</table>
In this paper, we have proposed a volume graph model for 3D facial surface extraction from volume data. This model formulates the surface extraction task as a min-cut problem in graph theory, so that the existing graph cut approach can be adopted for the searching of facial surface. We compared the proposed approach with MC by precision and recall. Experimental results demonstrate that VGM has significant better performance over the traditional MC method. Visual exhibition of the facial extraction results show that VGM is superior to MC in excluding the irrelevant tissues’ surfaces. The high accurate facial surface extracted by VGM can help synthesize 3D face for virtual plastic surgery.

6. CONCLUSIONS

In this paper, we have proposed a volume graph model for 3D facial surface extraction from volume data. This model formulates the surface extraction task as a min-cut problem in graph theory, so that the existing graph cut approach can be adopted for the searching of facial surface. We compared the proposed approach with MC by precision and recall. Experimental results demonstrate that VGM has significant better performance over the traditional MC method. Visual exhibition of the facial extraction results show that VGM is superior to MC in excluding the irrelevant tissues’ surfaces. The high accurate facial surface extracted by VGM can help synthesize 3D face for virtual plastic surgery.

7. REFERENCES


