52nd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
10 – 13 September 2007

K. Sickel / V. Daum / J. Hornegger

Shortest Path Search with Constraints on Surface Models of in-the-ear Hearing Aids

Abstract

One part of the production of modern hearing aids is the integration of a ventilation tube. This is usually done by applying a shortest path search algorithm on a non-convex ear impression. For the functionality of this so-called vent the path search algorithm has to fulfill several constraints like smoothness and space consumption of the vent. The proposed solution is based upon a modified version of Dijkstra's algorithm to address the constraints and to increase accuracy. The algorithm was evaluated by experts and validated to perform better for the application than a shortest path solution.

1. Introduction

Approximately eight percent of the world population suffers from hearing loss. This handicap impairs the life quality of persons concerned and can lead to social isolation. Hearing aids improve the life quality of these approximately 560 million people in the world [1]. Modern hearing aids can be categorized into two major classes: behind-the-ear (BTE) devices and in-the-ear (ITE) devices. The ITE devices, especially the completely in the canal (CIC) devices, are almost invisible to other people [2]. The major disadvantage of the ITE devices is that the production is complex, expensive and difficult. The production process starts with an ear impression of the patient made by an audiologist. This impression is scanned and the resulting surface mesh is smoothed and trimmed to fit the requirements of an ITE device. The physical impression is then built by a stereolithography machine and outfitted with the necessary components such as receiver and microphone. After some finishing tasks, like varnishing and polishing, the device can be sent to the customer. Due to the fact that an ITE closes the ear no pressure equalization is possible. To avoid this disagreeable effect a ventilation tube is integrated into the device. For a good air flow it is required that the vent is preferably short and smooth. The vent placement is currently done during the virtual editing of the scanned impression (shell) by using a shortest path algorithm.

Figure 1(a) shows an example of the ventilation tube. The computed path is not always favorable for the manufacturing process. In an ITE only a small amount of space can be used to fit in the rigid electronic components such as receiver, microphone and battery. To ensure the tiniest possible device the vent should be placed in a way that there is enough space for the other components. The usable space is approximated by circles fitted into slices of the impression which corresponds approximately to the shapes of the electronic components, see Figure 2. Altogether the production process enforces the following constraints on a new path search algorithm.

1. The vent shall be short.
2. The vent shall consume only a small amount of usable space inside the shell.
3. The vent shall avoid sharp bends to ensure a good air flow.
4. The computation time of the vent path shall be real time (< 2 sec).

2. Preliminaries

The shortest path problem is well-known in computational geometry. Accordingly there exist a large amount of algorithms to solve this particular problem. These algorithms can be grouped into exact and approximate algorithms, see e.g. [3] for an extensive overview. Due to run-time constraints it is favorable to concentrate on approximate algorithms, which are in general faster than the exact ones. We used Dijkstra's algorithm (DA) [4] as starting point for further investigations, because it has an $O(n \log n)$ complexity and can handle non-convex data. Additionally its objective function, which is the Euclidean length between the path nodes, can be easily adapted to additional requirements. A drawback of the DA is that the path length depends on the mesh granularity. To avoid this problem the Selective Refinement (SR) of Kanai and Suzuki [5] was employed and also modified. The SR introduces so-called Steiner Points into the mesh, see Figure 1(b). These additional points allow the path search to go across triangles by using so-called onface edges. Kanai and Suzuki showed in [5] that the computed path is on average only 0.4% longer than the exact path. The algorithm by Kanai and Suzuki uses the DA iteratively. In each iteration step Steiner Points are introduced and the search graph is reduced to the vertices in the current path and the neighbors of these vertices. In the first iteration the whole mesh is used. To improve the run-time this was modified such that in the first iteration the DA is used and starting with the second iteration Steiner Points are introduced. The algorithm terminates if either the difference between current and last path is smaller than a threshold or a fixed maximum number of iterations is reached. The number s of Steiner Points per edge is chosen to depend on the edge length |e| and an user given value γ according to the formula:

$$s = \lfloor |e| / \gamma \rfloor - 1 \tag{2.1}$$

3. Extension of the basis algorithm

The basis algorithm (DA+SR) fulfills the first and the last requirements of the given problem, but does not consider the space consumption and strong bends in the path. Especially the consumption of usable space is crucial for the production of small hearing devices and avoiding strong bends assures the correct functionality of the vent. To take these two points into account the so-called Volume Measure (VM) and Curvature Measure (CM) are introduced.

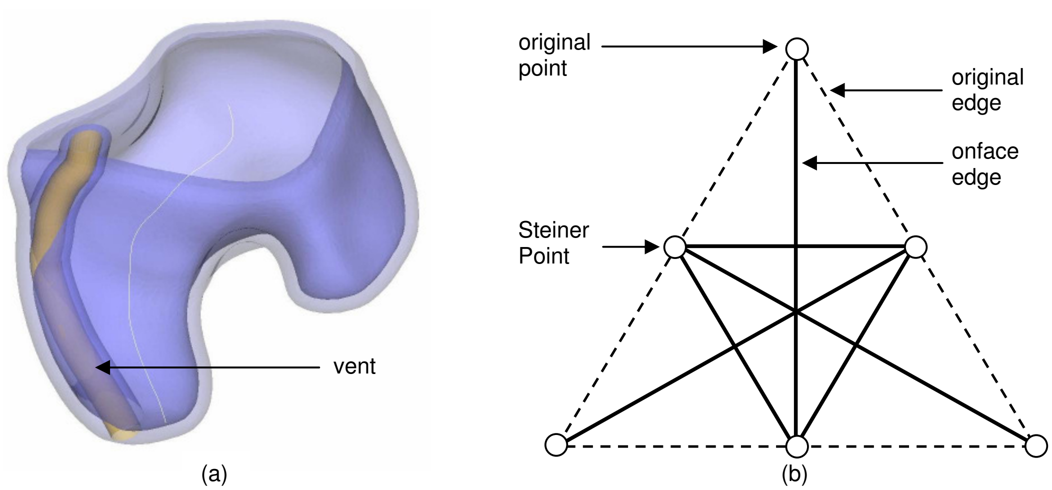

Figure 1. (a) Ear impression with ventilation tube in editing software. (b) Introducing of Steiner Points into a triangle mesh.

3.1 Volume Measure

The aim of the VM is to optimize the space consumption of the vent. To approximate the usable space in the shell we assume that a circle is the ideal representation for the space consumption of the electronic components. These circles are fitted into slices which are approximately perpendicular to the canal of the shell, see Figure 2. The number of slices depends on the number of vertices in the mesh. To incorporate the space usage of this circular region by the vent a weight ω_{VM} is introduced. To compute $\omega_{VM,i}$ at each vertex v_i a virtual vent is positioned, see Figure 2(a). If the vent cannot be

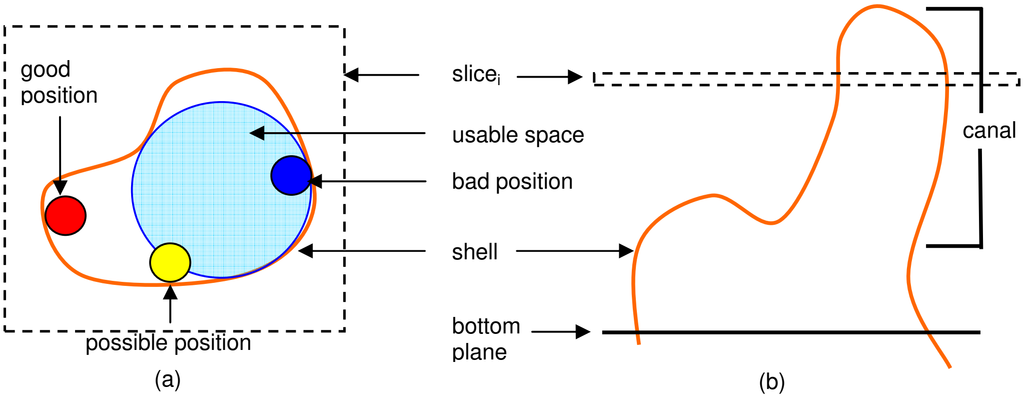

Figure 2. (a) Slice of the shell with example vents in a good, possible and bad position. (b) The impression is sliced approximately perpendicular to the centerline. The number of slices depends on the number of vertices in the impression.

fitted, a penalty term is added to $\omega_{VM,i}$. Second part of $\omega_{VM,i}$ is the overlapping distance between virtual vent and inner circle. A visual depiction of ω_{VM} can be seen in Figure 3, red is $\omega_{VM,i}$ = 0.0 and blue is $\omega_{VM,i} \geq 1.0$.

$$f_{i,j} = \omega_{L,i,j} + \beta \cdot \omega_{L,i,j} \cdot \omega_{VM,j} \tag{3.1}$$

The modified object function $f_{i,j}$ combines the Euclidean length $\omega_{L,i,j}$ between vertex v_i and vertex v_j with the VM. The VM is scaled with the edge length $\omega_{L,i,i}$. The variable β allows to control the influence of each part. We calculated the VM in a preprocessing step. Thus it can be examined and optimized independently of the other algorithms. The additional weights for the added Steiner Points are calculated via linear interpolation.

3.2 Curvature Measure

Due to the shape of the ear impressions it can happen that the vent has strong bends. It is also possible that the VM introduces bends in order to avoid penalty regions. To address these problems the curvature of the path is taken into account. Since DA is based on graph nodes (vertices) the incorporation of the CM requires the extension of the nodes to tuples of vertices. Each vertex can be combined with each neighbor thus one vertex with N neighbors results in N(N-1)/2 tuples. The curvature is approximated by an angle δ enclosed by three neighboring vertices. To avoid scaling artifacts it is, similar to the VM, scaled with the Euclidean length of the tuple. During the graph search only tuples with a sharing edge can be connected to a path. The additional weight is calculated as $\omega_{CM,i} = |180-\delta_i|$ and integrated into the objective function.

$$f_{i,j} = \omega_{L,i,j} + \beta \cdot \omega_{L,i,j} \cdot \omega_{VM,j} + \mu \cdot \omega_{L,i,j} \cdot \omega_{CM,j} \quad (3.2)$$

The Euclidean length $\omega_{L,i,j}$ is in this case the length of the tuple.

4. Results

The algorithm was tested on an Intel Core 2 Duo 2GHz CPU with 2GB RAM. The test data had a range of 2500 – 20000 vertices and 5000 – 40000 faces. Table 1 shows the results for the length of the path, the VM costs c_{VM} and the CM costs c_{CM}.

$$c_{VM} = \sum_{i=0}^{L} \omega_{VM,i} \quad (4.1)$$

$$c_{CM} = \sum_{i=0}^{L} \omega_{L,i} \cdot \omega_{CM,i} \quad (4.2)$$

L denotes the number of vertices respectively tuples of the path.

Algorithm combination	c_{VM}	c_{CM}	Path length
DA and SR	36.5	5.7	0.833
DA, SR and VM	22.6	8.2	0.837
DA, SR and CM	46.5	4.0	0.930
DA, SR, VM and CM	37.5	3.9	0.92

Table 1. Results of the different algorithm combinations. The used impression had about 6000 vertices.

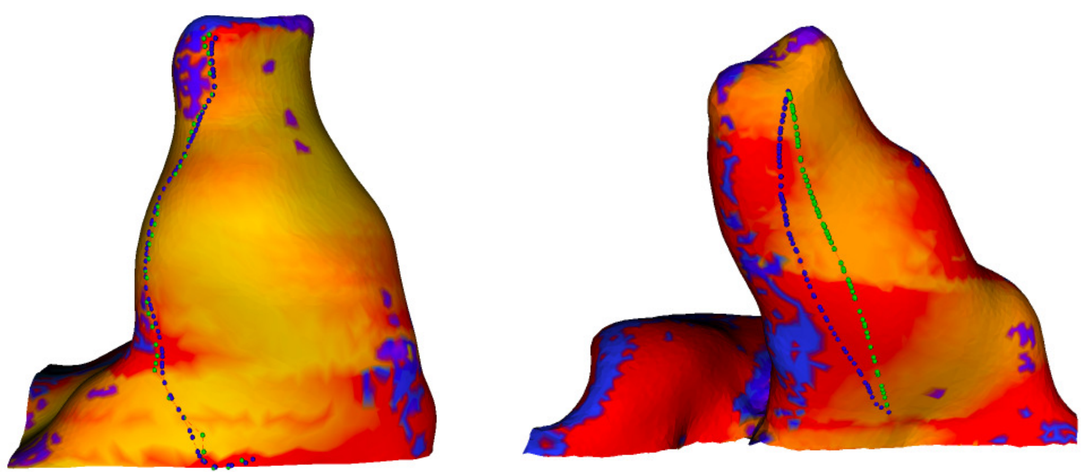

Figure 3. Example paths for (left) DA, VM, CM (green) in comparison with DA, SR, VM (blue) and (right) DA, SR (green) in comparison with DA, SR, VM (blue).

The costs in table 1 indicate that the VM significantly reduces the intersection between vent and inner circle and so optimizes the available space for the electronic components. The needed preprocessing time for the VM computation is less than a second for meshes with less than 10000 vertices. Also visible is the bad influence of the VM to the

CM costs which can be strongly reduced by using both extensions. In Figure 3(a) it can be seen that the paths favors the red regions of the surface and thus maximize the remaining space. Figure 3(b) shows the positive influence of the CM. The path is longer than a shortest path but optimizes the VM costs without introducing unfavorable bends. Table 2 compares the run times of the algorithm combinations, which reveals that the VM computation is fast. The CM computation with its tuple creation, however, needs further optimization.

Algorithm combination	Run time (6000 vertices)	Run time (12000 vertices)
Dijkstra, SR and VM	1.4 sec	2,5 sec
Dijkstra, SR and CM	86.1 sec	269,1 sec
Dijkstra, SR, VM and CM	86.5 sec	283,3 sec

Table 2. Results of the different algorithm combinations.

5. Conclusions

The developed extensions helped to address the open requirements to optimize the usable space consumption and to avoid sharp bends. Especially the VM works well and does not require that much overhead, because it can be done fast in a pre-processing step. The CM achieves an improvement for the path but violates the real-time requirement. It needs further research on how to prune the number of tuples and optimize the tuple-generation algorithm. The resulting paths, which were validated by experts, are superior to a simple shortest path solution.

References:

[1] hear-it.org. Es gibt immer mehr Hörgeschädigte. http://www.german.hear-it.org/page.dsp?forside=yes&area=134, 1st April 2007.

[2] National Institute on Deafness and Other Communication Disorders. Hearing Aids [NIDCD Health Information]. http://www.nidcd.nih.gov/health/hearing/hearingaid.asp, 1st April 2007.

[3] J.S.B Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, Handbook of computational geometry, pages 633-702. Elsevier Science, Amsterdam, 2000.

[4] E. W. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1:269-271, 1959.

[5] T. Kanai and H. Suzuki. Approximate shortest path on polyhedral surface based on selective refinement of the discrete graph and ist applications. In GMP `00: Proceedings of the Geometric Modeling and Processing 2000, page 241, Washington, DC, USA, 2000, IEEE Computer Society.

Authors:

Konrad Sickel
Volker Daum
Joachim Hornegger
Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Informatik 5 (Mustererkennung).
Martensstraße 3
91058 Erlangen
Phone: +49 9131 85 27775
Fax: +49 9131 303811
E-mail: konrad.sickel@informatik.uni-erlangen.de