Tailoring Biomechanical Model Meshes for Aero-Acoustic Simulations

Erik Widing[1,*], Örjan Ekeberg[2]
Department of Computational Biology,
School of Computer Science and Communication,
KTH Royal Institute of Technology,
Stockholm, Sweden
∗ Corresponding author
1 ewiding@kth.se
2 ekeberg@kth.se

Abstract

To simulate the airflow and acoustic wave propagation associated with voice production, a closed surface mesh representing the vocal tract is needed. Biomechanically, the vocal tract is composed of surfaces from several different anatomical structures. We present a method for assembling a dynamic vocal tract mesh by trimming and stitching surface meshes tracking biomechanical models of relevant structures. Two algorithms, one for trimming and one for stitching, are used to first isolate surface mesh patches that are in contact with the airway and then merge them into a closed surface. The algorithms rely on manually selected boundaries and are able to cover gaps between mesh patches. Test cases are used to illustrate how the algorithms behave in various situations. The algorithms are implemented in the toolkit ArtiSynth where many relevant biomechanical models are already available.

Keywords: mesh, voice, trimming, stitching, vocal tract.

1 Introduction

The voice is made possible by a coordinated interplay between different anatomical structures, in particular those originally evolved for breathing and eating. The vocal folds, controlled by activation of laryngeal muscles, and pressure from the lungs together create vibrations that cause acoustic waves to propagate through the vocal tract. The vocal tract forms an acoustic filter that strongly influences the sound emitted. Additionally, the airflow can cause fricative sounds at constrictions or other vibrating tissues (e.g. tongue tip or lips). This is all dynamic due to varying activation of a large set of muscles.

The number of interacting components and the different kinds of physics required to understand them makes simulation an indispensable tool for studying voice production theoretically. The acoustic waves propagate through the complex domain that the vocal tract forms with small cavities and sharp bends. The aerodynamics involves turbulent airflow due to narrow passages in the vocal tract as well as interactions with the rapidly oscillating vocal folds. The whole system is dynamically controlled via a large number of muscles and complicated by non-trivial phenomena like contact between soft and wet tissue. Analyzing such a mix of intertwined models of different types is difficult to tackle without computational tools.

EUNISON (*Extensive UNIfied-domain SimulatiON of the human voice*; www.eunison.eu) is a EU-funded project with the aim of simulating models based on the different kinds of physics involved in creating the voice: the biomechanical muscle control of the vocal tract and vocal folds, the fluid-structure interaction that takes place at the vocal folds and creates phonation, along with acoustics and aerodynamics in a dynamically changing vocal tract. The project also involves experimentation on simplified physical vocal folds and vocal tract models to validate that the simulations give accurate results. The vision is to create a unified mathematical domain in which all the different simulations take place.

ArtiSynth (Lloyd et al., 2012) is a biomechanical toolkit providing interactive 3D-simulations of the structures involved in voice production. In ArtiSynth, rigid bodies are used to model bone-structures, while finite element models (FEM) are used for soft tissue. Viscoelastic muscle models are implemented, either as point-to-point actuators or as embedded force generators. ArtiSynth allows the user to merge all these different ways of modelling into hybrid models. This makes it possible to simultaneously simulate all the biomechanics that contribute in producing the voice.

Biomechanical models of muscles and muscle activation suitable for dynamic simulations are already well developed and established (Lloyd and Besier, 2003; Sanchez et al., 2013). Models of soft tissue like the tongue and its muscle control are also available (Gérard et al., 2003). Regarding the acoustics, there exists well-established methods for computing the filtering properties based on vocal tract shapes (Arnela and Guasch, 2013). Handling the aerodynamics is a more challenging task but there exists methods to tackle this (Codina and Blasco, 2000; Hoffman and Johnson, 2006). Challenges like how to construct meshes of complicated domains suitable for simulations have also been addressed (Sazonov et al., 2006). One key part missing in this work flow is that a dynamic vocal tract needs to be created from the biomechanics so that it can be used for aerodynamic and acoustic simulations.

The focus of this work is on the vocal tract rather than on the vocal folds. The focus is more specifically on how to obtain the vocal tract mesh from the geometries of the surrounding bodies each of which is described by biomechanical models and controlled by muscles. The vocal tract is not a mechanical entity in itself but rather the absence of anatomical structures. The biomechanical bodies around it are typically modelled as rigid or soft bodies. In practice, there will often be parts that are not explicitly modelled, but just because a body is not modelled that does not mean it is not there; some form of interpolation or placeholder is then needed to complete the vocal tract geometry.

The surfaces of both the rigid bodies and the FEM-bodies in ArtiSynth are represented as meshes. A surface mesh is an entity consisting of vertices and faces. Each vertex has a position in 3D-space and each face connects three or more vertices. From here on, we will only consider meshes with triangular faces. Rigid bodies are described by a rigid surface mesh while a FEM-body is typically described by a volume-mesh but it is straightforward to extract its surface mesh. This makes it possible to extract the surrounding surfaces forming the vocal tract.

By identifying the parts of the surrounding surface meshes that are in contact with the vocal tract and then merging them together, it is possible to obtain a surface enclosing the vocal tract. This will create a closed tube that can serve as boundary conditions for simulations of acoustics and aerodynamics. One advantage of this approach is that the mesh vertices are still tied to their original biomechanical structures, and will therefore follow their movements. This makes the vocal tract mesh dynamic as it inherits the movements of the underlying biomechanical models. One disadvantage is that the resulting vocal tract mesh may end up with a highly varying size of its faces. The merging process has to fill in any gaps between the meshes in order to get a closed tube that is only open at the two ends, that is the mouth and the larynx. For this purpose two algorithms, one for trimming and one for stitching, have been developed. By trimming we mean an algorithm that cuts away unwanted parts of a mesh and by stitching we mean an algorithm that merges different meshes by filling in the gaps between the meshes with new faces.

Among the many different trimming-techniques (Attene et al., 2006; Bruyns et al., 2002; Shamir, 2008), there are three main categories (Zhang et al., 2011). First, there are boundary-based algorithms where sample points on the intended boundary of trimming are given. We use the term "boundary" to refer to a closed curve on the surface described by the mesh. The method includes an algorithm that connects the points into one connected loop that forms the boundary. After that, the outside of the boundary is cut away and the mesh then only consists of the part that is inside the boundary. This approach is very sensitive to the choice of boundary (Fan et al., 2011; Zhang et al., 2010) so great care must be taken when choosing the boundary. The second approach (Ji et al., 2006) is to mark an area fairly similar to the part of the mesh that is desired. The method then uses an 'active contour'-algorithm to decide what parts to keep. It then uses mesh scissoring algorithms to decide where the boundary should be and cut the mesh along it. This approach has the obvious disadvantage that the algorithms might choose a boundary that does not conform with the users intentions. The third and last main approach is to label some sample points as foreground or background in which foreground includes the desired parts and background is the rest of the mesh. The method then uses a region-growing algorithm to decide for the rest of the mesh which points belong to foreground and which points belong to background. This approach has the advantage that it is less dependent on the users input but on the other hand it may be difficult to achieve the desired results.

Stitching techniques mainly focus on merging meshes that are in close contact with each other (Yau et al., 2003), or, if the meshes are far apart, adjust their relative position and posture (Turk and Levoy, 1994). This is, however, different from what we need in this work, where the meshes to be merged are already in their correct positions and only need to be merged into a closed tube.

2 Methods

Both trimming and stitching have been implemented in ArtiSynth, which makes it possible to apply them to the different existing models, in particular the model described by Stavness et al. (2011), in order to produce a dynamic vocal tract. The algorithms operate on user defined *stitchpaths* along which the trimming and subsequent stitching will be performed. A stitchpath is represented by a *stitchlist* consisting of sequences of connected vertices.

2.1 Trimming

The trimming technique that has been used in this work is an adapted boundary-based approach in which a connected sequence of vertices on the boundary is given as input. This is motivated by the fact that we have precise requirements on which parts of the different biomechanical components should be parts of the vocal tract.

Trimming is performed along a closed boundary that defines what part of the mesh should remain. In contrast to most trimming and cutting algorithms (Bruyns et al., 2002) the boundary is given as a list of connected vertices in the mesh, so no new vertices or edges are created. The algorithm performs two recursive depth-first traversals starting from a vertex inside the boundary. This vertex is specified in the code along with the vertex list defining the boundary. The first traversal stops whenever it reaches the boundary. Therefore this traversal will visit all vertices inside the boundary. In a second traversal, the boundary is ignored that means all vertices are visited. Consequently, the first traversal visits only vertices that should be kept and the second visits all vertices in the mesh. Therefore, vertices that are visited only in the second traversal and are not part of the boundary are flagged as vertices to be removed.

It is also necessary to determine which faces to remove. Any face containing a vertex marked for removal is also marked for removal. This test is, however, not sufficient; faces that have all their vertices on the boundary may be on either side of the boundary. For faces with all vertices on the boundary, if the order of its vertices is the same as their order on the boundary, the face will be on the inside and should remain. For this test to work, we require that both the boundary and all the faces have a consistent orientation, that is, that their vertices are specified in a clockwise or anti-clockwise order.

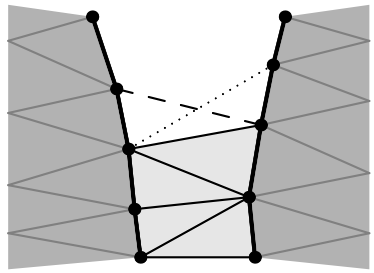


Figure 1. Schematic illustration of the stitching algorithm. Black dots indicate the vertices in the two stitchlists, which are here stitched sequentially from the bottom. In each step, the length of the two alternative edges (dotted and dashed) is compared, and the shortest one is used to form the next face.

2.2 Stitching

Since the stitching technique needs to cover gaps between meshes, the approach is to add faces in the gaps between the meshes. In this way no new vertices are created, which is good since new vertices would not naturally follow the movements of any bodies. Stitching is performed pairwise between stitchlists and the algorithm is designed to work for stitchpaths that are approximately parallel. This assumption is reasonable in the applications that we are facing and we have therefore not explored the behaviour in extreme situations when the method breaks down. We use a modified merge-sort algorithm, similar to the hole triangulation in the hole-filling algorithm described by Liepa (2003), which adds new faces in the gaps between the vertices of the lists (Figure 1), but with the difference that our algorithm can stitch meshes that are initially not connected. The algorithm allows an arbitrary number of steps forward along one of the stitchlists without advancing the other which means that the two stichlists are not restricted to have the same number or density of vertices.

If three patches are stitched together naively using this algorithm, one face at their common meeting point will be missing (Figure 3b). This could be solved by simply adding the missing face, but, to avoid introducing such special cases, we have chosen a method where patches are always stitched pairwise. The user-supplied parameters for such an operation are two stitchlists and a flag requesting that other stitchlists of the patches are concatenated if they become connected in the process (Figure 3c-3d).

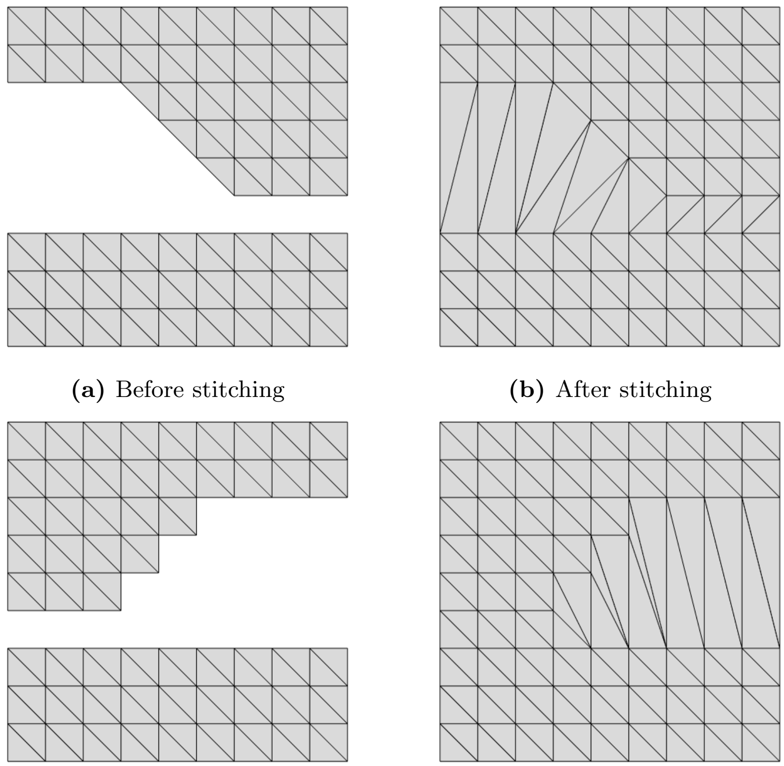
3 Results

In order to verify that the algorithms work properly, a number of test cases have been created. A few cases are presented here, along with the results when assembling a vocal tract. The figures in this section are all generated with ArtiSynth.

3.1 Test Cases

The stitching is required to handle stitchpaths that are not necessarily parallel. This is highlighted in the examples presented in Figure 2, where the upper patches are the results of trimming. In the first example, the triangulation allows a stitchpath that is relatively smooth (Figure 2a). Figure 2b shows these two meshes stitched together. The stitched faces are similar to the rest of the mesh in the part of the stitch where the stitchpaths are close, but the stitched faces become more and more elongated as the gap between the two parts widens.

This highlights two potentially problematic properties inherent in this type of stitching. The first is that the area of different faces on a stitched mesh might differ quite a lot. The second is that some of the



(a) Before stitching **(b)** After stitching

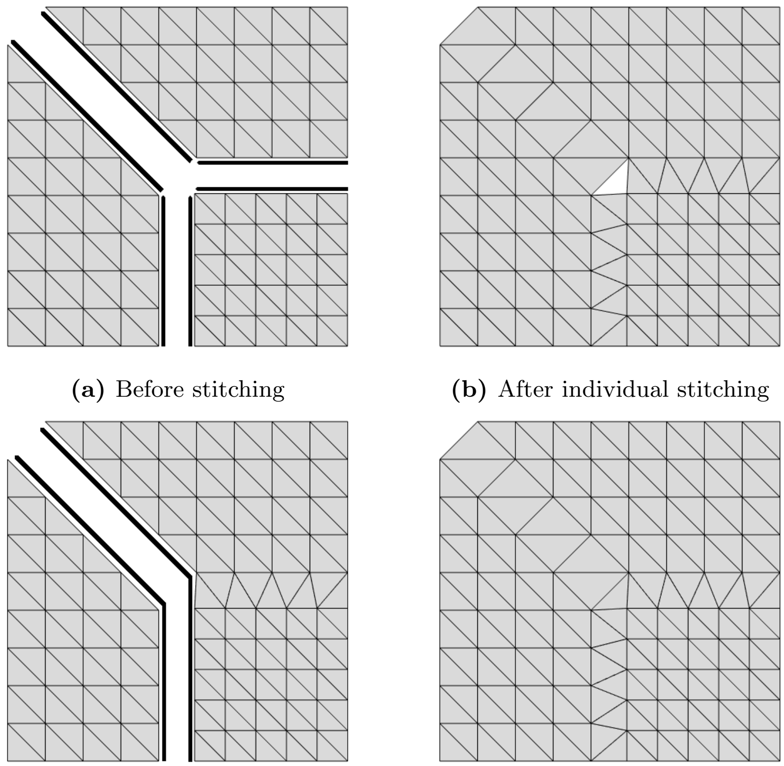
(c) Before stitching **(d)** After stitching

Figure 2. Two examples of stitching along non-parallel stitchpaths. a) Mesh before stitching where the stitchpaths make smooth turns but are not always parallel. b) Mesh after stitching. Note that some faces are elongated due to the varying distance between the stitchlists. c) Mesh before stitching. Due to the inconvenient triangulation of the mesh it is not possible for the stitchpath to make smooth turns. Instead several sharp turns are made. d) Mesh after stitching. Because of the sharp angles the stitchpath is almost parallel to the stitched faces, which creates both large and small angles.

angles of such elongated faces can be very small, which might cause problems further down in the work flow, in particular in connection with the acoustic simulations.

In the second example, the triangulation is made so that the stitchpath is forced to make several sharp turns (Figure 2c). Figure 2d shows the resulting mesh after stitching. The stitched faces match the rest of the mesh well in the area where the gap between the different parts is small, but in the area where the gap is larger, the faces are elongated. In the area where the transition between small and large gaps takes place, there are some triangles that are even more obtuse than in Figure 2b due to the fact that the gap increases but also because the number of vertices in the stitchlists are different in this area.

A more challenging task for stitching is the junction of several pairs of stitchlists. Figure 3 shows an example where stitching has been applied to three different meshes with two stitchlists each. Each mesh connects to the other two along one stitchlist each. The three pairs of stitchlists will meet in a junction in the middle of the mesh. Figure 3b shows what would happen if the three pairs were to be stitched separately. This approach does not handle any face that connects to all three parts and hence there is a hole instead of a face at the junction. Figure 3c shows the first step of the sequential approach, where the remaining stitchlists are updated whenever a pair of meshes have been stitched together. In this case



(a) Before stitching **(b)** After individual stitching

(c) After one stitching **(d)** After updated stitching

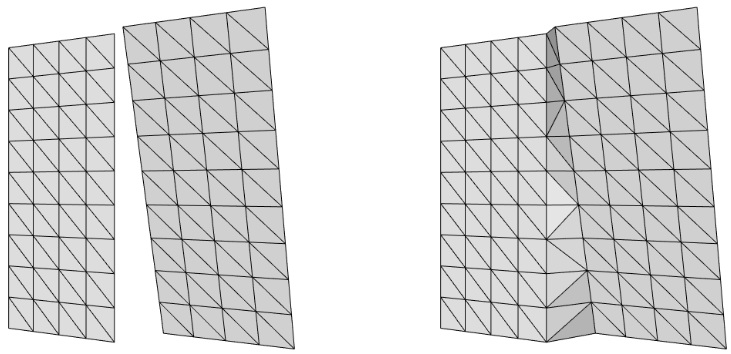
Figure 3. Stiching of three patches. Stichpaths are indicated by black arrows. a) Mesh before stitching. b) Mesh after stitching of different paths individually. Note the missing face at the junction. c) Mesh after first stitching. The remaining stitchlists have been updated so that the next stitching can be made in one piece. d) Mesh after the updated stitching. Note that the face in the junction has been filled in.

one pair of stitchlists has been stitched together and the rest of them are being updated. Since two of the remaining stitchlists have now come into contact due to the stitching they are concatenated. Figure 3d shows the mesh after the second stitching, resulting in a correctly merged mesh that covers the entire gap with faces. Since it is known in advance which lists are going to be merged and which are not, it is straightforward to set this sequential procedure up in advance.

Up to this point, we have only examined how the stitching handles stitchpaths that are in the same plane. Figure 4 shows an example of stitching two meshes that are not in the same plane. That makes the stitchpaths not only non-parallel but also forces the stitched faces to be in different planes (Figure 4a). In fact, the extended planes of the two meshes intersect each other right in the middle of the intended stitching. As can be seen in Figure 4b the algorithm handles this situation well.

3.2 Vocal Tract

The trimming and stitching have been applied to different biomechanical components of the vocal tract. We here present the results when the algorithms are applied to the tongue, the maxilla and the mandible. Figure 5 shows the meshes of the different components before and after the trimming. A simplified mesh of the static vocal tract (Figure 5a) based on 3D images was used as a rigid model for the lower parts



(a) Before stitching **(b)** After stitching

Figure 4. Mesh before and after stitching. a) The meshes that are going to be stitched together are not in the same plane. b) The mesh after stitching. Even though the different parts are not in the same plane the stitching is successful.

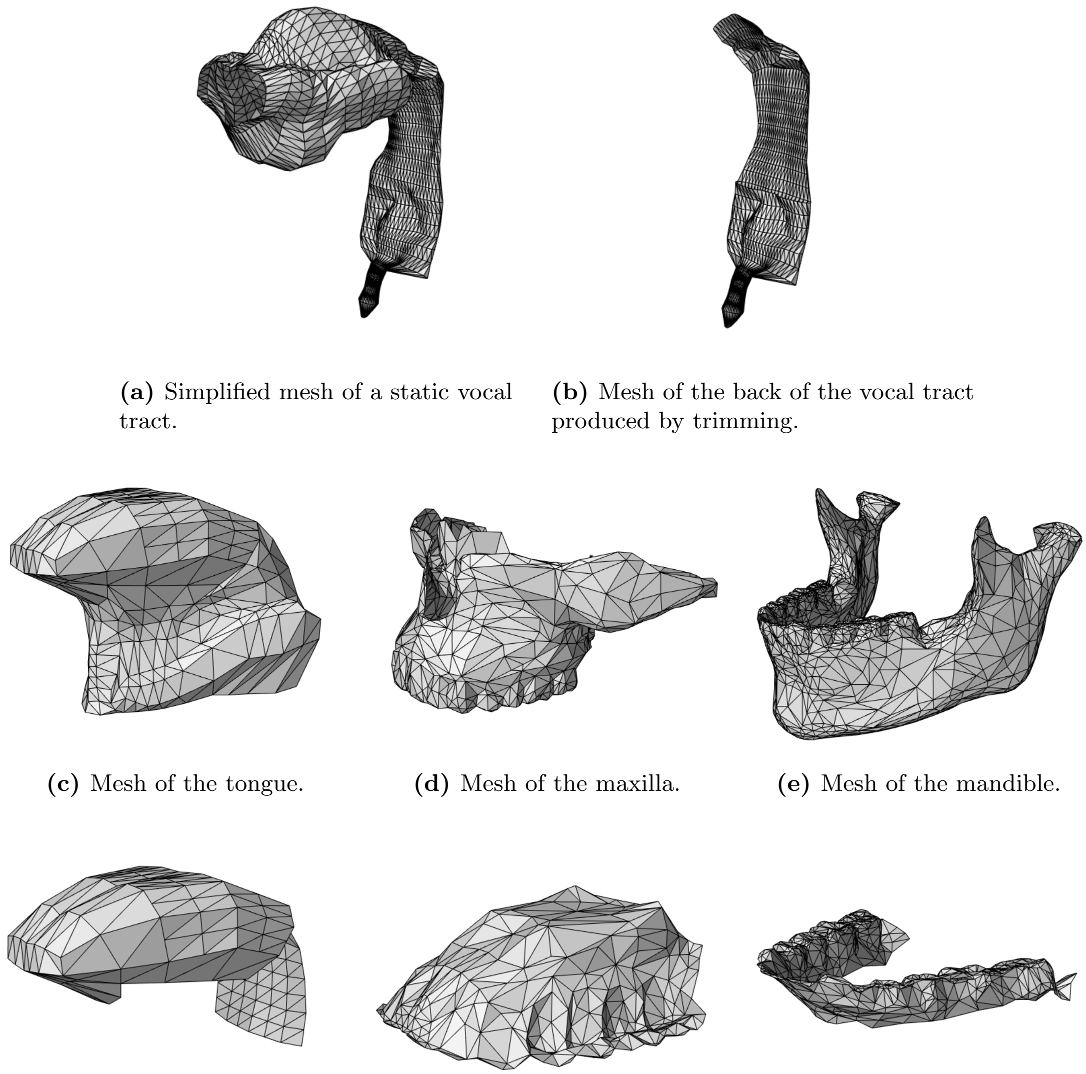
and back of the vocal tract (Figure 5b). The upper part of what remains of the mesh after the trimming represents the soft palate. The surface mesh of the tongue (Figure 5c) was taken from an existing FEM model. Some parts below the tongue tip have been kept during trimming (Figure 5f) because they actually become part of the vocal tract whenever the tongue is raised. Most of the maxilla mesh (Figure 5d) will not be part of the vocal tract and will therefore be removed in the trimming process. Most of it has been cut away but the hard palate and the teeth have been kept (Figure 5g). Like for the maxilla, most of the mandible mesh (Figure 5e) will be cut away. Only the inside of the teeth and some structure behind is kept (Figure 5h).

After the trimming of the pieces, the next step is to stitch them together into a closed tube. Figure 6 shows the meshes of all the biomechanical components together before (6a) and after (6b) stitching. Note that there are quite large variations in the size of the gap between the different components. The mesh in Figure 6b has been produced by first stitching the maxilla to the simplified mesh of the vocal tract and then stitching the mandible to the tongue. After both these stitchings, the stitchlists have been updated that has made it possible to make the rest of the stitched faces in one final stitching.

Note that the faces introduced during stitching tend to represent tissue, which is not explicitly modelled. The stitched faces between the mandible and the maxilla correspond to the cheeks. The stitched faces between the mandible and the tongue correspond to the floor of the mouth. The stitched faces between the simplified vocal tract mesh and the maxilla represent part of the soft palate. The stitched faces between the tongue and the simplified mesh of the vocal tract represent some of the tissue closest to the tongue in the lower vocal tract. There are some stitched faces that connect the tongue and the maxilla due to the stitching of updated lists even though there is little or no real tissue that actually connects them. These stitched faces rather represent the tissue connecting the mandible with the soft palate.

4 Discussion

The algorithms presented have proven useful in creating a closed vocal tract mesh from the surfaces of a set of biomechanical models of the surrounding tissue. Further, no new vertices have been introduced in the process, which means that the position of the entire vocal tract mesh is defined in terms of the moving biomechanical models.



(a) Simplified mesh of a static vocal tract.

(b) Mesh of the back of the vocal tract produced by trimming.

(c) Mesh of the tongue.

(d) Mesh of the maxilla.

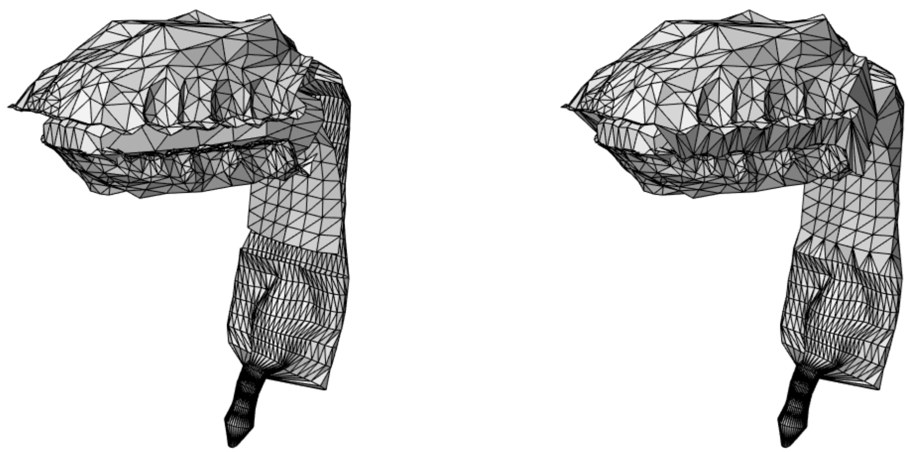
(e) Mesh of the mandible.

(f) Mesh of the tongue after trimming.

(g) Mesh of the maxilla after trimming.

(h) Mesh of the mandible after trimming.

Figure 5. The meshes of the biomechanical components that have been stitched together into a mesh of the vocal tract. The figure shows the meshes before (a, c, d, e) and after (b, f, g, h) trimming.



(a) Before stitching **(b)** After stitching

Figure 6. The different trimmed meshes of the biomechanical components of the vocal tract. a.) Different components added together. b.) Different components stitched together.

From the acoustic and aerodynamic simulations point of view it is preferable to have a boundary mesh where the faces are as close to being equilateral triangles as possible. The designs of the algorithms have been chosen with this in mind, but without introducing new vertices it is unavoidable that this will sometimes be violated. We are currently addressing this by post-processing the vocal tract mesh to get a more regular mesh, which can both serve as the boundary condition for the acoustic simulations and still track the movements of the biomechanical components. The new vertices of this post-processed mesh are no longer uniquely coupled to the biomechanical models. However, since the mesh is perfectly aligned with the surfaces of the biomechanical components, it is possible to move these new vertices by averaging the movements of nearby biomechanically controlled vertices.

One problem that has emerged is that the vocal tract mesh can sometime intersect itself. There is nothing explicit that prevents the different meshes from intersecting before they are stitched together. We may expect that the contact detection functionality in ArtiSynth would guarantee that two biomechanically modelled bodies would never have intersecting surfaces, but in fact, the contact modelling used in ArtiSynth allows for small intersections between bodies as a base for calculating the counteracting forces that prevents objects from passing through each other.

The way our trimming algorithm is implemented imposes a requirement that all the faces in a mesh have the same orientation. The stitching on the other hand does not take this into account at all. The algorithm implemented ensures that all new faces in a stitching have the same orientation but at this point there is no guarantee that this orientation conforms with the orientation of the rest of the surface.

One of the major limitations with the current implementation is, however, the lack of support for interactive creation of stitchlists within ArtiSynth. The stitchlists are currently generated by manually selecting vertices using external software and their indices are then copied directly into the code of the model. This makes the process more error prone than in software specifically created for the task (Sharf et al., 2006).

5 Conclusions

Two algorithms, one for trimming and one for stitching, have been presented. They are explicitly designed for merging surface meshes from biomechanical model components to form a continuous surface. Both algorithms rely on carefully selected boundaries and are able to bridge gaps between different meshes. Test cases have been used to investigate how the algorithms behave in various situations. This work has been done in order to be able to create a closed surface mesh of the vocal tract from the surrounding biomechanical bodies. Such a mesh is needed to simulate the airflow and acoustic wave propagation associated with voice production. The algorithms have been implemented in the biomechanical toolkit ArtiSynth in order to use them to create a dynamic vocal tract.

Acknowledgments

The authors would like to thank the other members of EUNISON for their useful discussions and feedback. The authors would also like to express their gratitude to the ArtiSynth community, in particular Ian Stavness for extensive advice and feedback.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is supported by EU-FET grant of EUNISON [grant number 308874].

References

Marc Arnela and Oriol Guasch. Finite element computation of elliptical vocal tract impedances using the two-microphone transfer function method. *The Journal of the Acoustical Society of America*, 133 (6):4197–4209, 2013.

M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation - a comparative study. In *IEEE International Conference on Shape Modeling and Applications, 2006. SMI 2006*, pages 7–7, Los Alamitos, June 2006. IEEE Computer Science Press.

Cynthia D. Bruyns, Steven Senger, Anil Menon, Kevin Montgomery, Simon Wildermuth, and Richard Boyle. A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools. *The Journal of Visualization and Computer Animation*, 13(1):21–42, 2002.

Ramon Codina and Jordi Blasco. Stabilized finite element method for the transient navier–stokes equations based on a pressure gradient projection. *Computer Methods in Applied Mechanics and Engineering*, 182(3–4):277 – 300, 2000. ISSN 0045-7825.

Lubin Fan, Ligang Lic, and Kun Liu. Paint mesh cutting. *Computer Graphics Forum*, 30(2):603–612, 2011.

Jean-Michel Gérard, Reiner Wilhelms-Tricarico, Pascal Perrier, and Yohan Payan. A 3d dynamical biomechanical tongue model to study speech motor control. *Research Developments in Biomechanics*, 1:49–64, 2003.

Johan Hoffman and Claes Johnson. Stability of the dual navier–stokes equations and efficient computation of mean output in turbulent flow using adaptive dns/les. *Computer Methods in Applied Mechanics and Engineering*, 195(13–16):1709 – 1721, 2006. ISSN 0045-7825. A Tribute to Thomas J.R. Hughes on the Occasion of his 60th Birthday.

Zhongping Ji, Ligang Liu, Zhonggui Chen, and Guojin Wang. Easy mesh cutting. *Computer Graphics Forum*, 25(3):283–291, 2006.

Peter Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 200–205, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

David G. Lloyd and Thor F. Besier. An emg-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo. *Journal of Biomechanics*, 36(6):765–776, 2003.

John E Lloyd, Ian Stavness, and Sidney Fels. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 355–394. Springer, New York, 2012.

C. Antonio Sanchez, John E. Lloyd, Sidney Fels, and Purang Abolmaesumi. Embedding digitized fibre fields in finite element models of muscles. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2(4):1–14, 2013.

Igor Sazonov, Desheng Wang, Oubay Hassan, Kenneth Morgan, and Nigel Weatherill. A stitching method for the generation of unstructured meshes for use with co-volume solution techniques. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1826–1845, 2006.

Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.

Andrei Sharf, Marina Blumenkrants, Ariel Shamir, and Daniel Cohen-Or. Snappaste: An interactive technique for easy mesh composition. *The Visual Computer*, 22(9-11):835–844, 2006.

Ian Stavness, John E Lloyd, Yohan Payan, and Sidney Fels. Coupled hard-soft tissue simulation with contact and constraints applied to jaw-tongue-hyoid dynamics. *International Journal for Numerical Methods in Biomedical Engineering*, 27(3):367–390, 2011.

Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *SIGGRAPH 94 Conference Proceedings*, pages 311–318, New York, 1994. ACM.

Hong-Tzong Yau, Chuan-Chu Kuo, and Chih-Hsiung Yeh. Extension of surface reconstruction algorithm to the global stitching and repairing of stl models. *Computer-Aided Design*, 35(5):477–486, 2003.

Joyung Zhang, Jianmin Zheng, and Jianfei Cai. Interactive mesh cutting using constrained random walks. *Visualization and Computer Garphics*, 17(3):357–367, 2011.

Juyong Zhang, Chunlin Wu, Jianfei Cai, Jianmin Zheng, and Xue-Cheng Tai. Mesh snapping: Robust interactive mesh cutting using fast geodesic curvature flow. *Computer Graphics Forum*, 29(2):517–526, may 2010.