Towards real-time two-dimensional wave propagation for articulatory speech synthesis

Victor Zappi, Arvind Vasuvedan, Andrew Allen, Nikunj Raghuvanshi, and Sidney Fels

Citation: Proc. Mtgs. Acoust. **26**, 045005 (2016); doi: 10.1121/2.0000395
View online: https://doi.org/10.1121/2.0000395
View Table of Contents: http://asa.scitation.org/toc/pma/26/1
Published by the Acoustical Society of America

Proceedings of Meetings on Acoustics

Volume 26 http://acousticalsociety.org/

171st Meeting of the Acoustical Society of America

Salt Lake City, Utah

23-27 May 2016

Physical Acoustics: Paper 1pPA8

Towards real-time two-dimensional wave propagation for articulatory speech synthesis

Victor Zappi
Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy; victor.zappi@gmail.com

Arvind Vasuvedan
Department of Electric and Computer Engineering, University of British Columbia, Vancouver, BC, Canada; arvind@ece.ubc.ca

Andrew Allen
Google, Inc. Mountain View, CA; bitllama@google.com

Nikunj Raghuvanshi
Microsoft Research, Redmont, WA; nikunjr@microsoft.com

Sidney Fels
Department of Electric and Computer Engineering, University of British Columbia, Vancouver, BC, Canada; ssfels@ece.ubc.ca

The precise simulation of voice production is a challenging task, often characterized by a tradeoff between quality and speed. The usage of 3D acoustic models of realistic vocal tracts produces extremely precise results, at the cost of running simulations that may take several minutes to synthesize a few milliseconds of audio. In contrast, 1D articulatory vocal synthesizers rely on highly simplified acoustic and anatomical models to achieve real-time performances, but can only partially match the spectra of realistic vocal tracts. In this work, we present a novel articulatory vocal synthesizer, based on a fast 2D propagation model running on a graphics card (GPU). The system can run in real-time under specific conditions and, differently from 1D synthesizers, allows for simulating airflow propagation through asymmetric and curved geometries. This paper covers details on the GPU implementation of the different components of the system, including the 2D Finite-Difference Time-Domain wave solver and the excitation mechanism. A preliminary evaluation is presented, using area functions to simulate static vowels. Three different resolutions are tested, combined with two alternative ways of discretizing the 2D geometries. The computed formants are overall characterized by small positional errors while computational times are comparable with those from 1D systems.

Published by the Acoustical Society of America

 [DOI: 10.1121/2.0000395]

1. INTRODUCTION

Among the several speech synthesis techniques available nowadays, Articulatory Vocal (AV) synthesis is one of the most challenging. Differently from other speech technologies, an AV synthesizer aims to simulate the physical phenomena underlying vocal production, including the propagation of acoustic waves throughout the human upper vocal tract. The numerical solution of pressure propagation in complex 3D geometries, like the ones inherent to speech utterances, is a heavy computational problem and current proposed solutions provide precise results but require very long simulation times (e.g., 60 minutes [1] and 44 minutes [2], for 5ms of audio).

Faster simulations can be achieved by bounding propagation to one dimension only. This approach can achieve real-time performance and produce a good match of vowels' first formants (i.e., resonances) [4, 5, 6], but it implies some caveats. By definition, 1D propagation describes the motion of planar waves and the emergence of fundamental modes. As a consequence, the only geometries that can be directly simulated by 1D systems are straight cylindrical tubes, whose radial symmetry cancels out any transverse mode. Under this constraint, concatenations of cylindrical segments can be used to obtain shapes that well approximate the lower part of real vocal tracts' spectra, where the effects of transverse modes are negligible [3]. Some techniques exist to augment the acoustics of a cylindrical tube and approximate the effects of additional geometrical features, like curvature [8] and branching (e.g., piriform fossae, subglottal tract)[9, 10]. However, the complexity of realistic vocal tract geometries is still out of reach. In such irregular domains, resonances and antiresonances attributable to non-planar propagation heavily characterize speech acoustics, especially beyond 5 kHz [7]. The approximation of these effects in 1D simulations is still an open challenge and, as a result, 1D models generally produce a poor match of the higher end of the speech spectrum [2, 7]. This approach has a moderate effect on the intelligibility of the synthesis output, but strongly influences its perceived naturalness [11].

As a step forward from pure plane wave simulations, speech synthesis research has recently explored the usage of two dimensional models for pressure propagation, which provides a valuable trade-off between speed and quality. Regardless of the chosen numerical method, the computational time needed by 2D models is much lower than the case of 3D simulations [2] and the generated speech has been reported to better resemble the natural voice [12]. This effect derives from the ability to natively simulate transverse modes along the added dimension. As a consequence, 2D models are not bound to straight cylindrical tubes, but allow for the computation of pressure propagation in more realistic geometries, with relatively little effort.

It is possible to find several examples of 2D propagation models relying on different numerical approaches. Speed et al. [13] and Wang et al. [14] explored the usage of lightweight 2D Finite-Difference Time-Domain (FDTD) methods, reporting fairly fast simulation times for static vowel synthesis, even if far from real-time. One of the challenges of this approach consists of maintaining stability when the domain boundaries (i.e., the vocal tract walls) are dynamically modified, as in the case of articulation. This is not an issue in Finite Element Models (FEMs), like the one presented by Arnela et al. [2, 15]; their system achieved very precise results (also in dynamic scenarios like diphthongs), but required long simulation runs. Mullen et al. extensively worked with 2D digital waveguide models and managed to craft a fast system capable of simulating dynamic boundaries [16]. When running at a coarse spatial resolution (i.e., 11 mm), their model reaches real-time performances.

We present a preliminary evaluation of an AV synthesizer based on a novel 2D propagation model. The solver runs a fast FDTD schema that makes use of massive core parallelism available

on commodity graphics cards (GPUs). In the current configuration, real-time performances can be achieved up to a spatial resolution of 2.24 mm (sample rate of 220.5 KHz). When finer step sizes are required, like in the case of vocal tract area functions (tubes with circular cross-sections), the measured computational times are still substantially lower than the values reported for other simulations in literature [1, 2]. The model features an algorithm to support dynamic geometry without incurring stability issues. The model can also be coupled with two glottal models that we implemented that are also run on the GPU.

The paper is structured as follows. In Section 2, we present the details of the implementation of both the GPU wave solver and the two glottal models, including details on the strategy used for simulating boundary losses. In Section 3, a set of realistic area functions is used to build 2D domains and simulate static vowel sounds, using two different approaches: one to preserve the symmetry of the tubes, the other to minimize the discretization error. Section 4 describes the procedure used to calculate the transfer functions of the 2D vocal tracts using different resolutions; a brief analysis of the position of the first formants is also included, using as a reference a high-precision 3D FEM. The paper ends with Section 5, where conclusions and future perspectives are discussed.

2. ACOUSTIC MODEL

The core of the AV synthesizer presented in this work is an acoustic model that runs almost entirely on the GPU. The main components of the model are written as OpenGL vertex and fragment shaders, handled by a C++ engine that also manages timing, control input and audio output. The first component of the model is a wave solver that computes pressure wave propagation through 2D vocal tract contours. The synthesizer also includes two alternative glottal models that can be used to excite the simulation domain and are coupled with it so back pressure influences the glottal models. Vocal tract wall losses are simulated where pressure interacts with the 2D contours; furthermore, the model implements Perfectly Match Layers (PML) to account for free field radiation, but does not yet include a head model.

A. WAVE SOLVER

The acoustic model is based on a real-time wave solver, originally designed for wind-instrument sound synthesis [17] and adapted in this work for the simulation of voice. The solver leverages the massive parallelism available on commodity GPUs to afford fast full-bandwidth computation on dynamically-changing vocal tract profiles. Present day computational power limits such simulations to two dimensions, although the solver is general and extends to 3D. Here we briefly review details from [17] relevant to the current work.

FDTD is a standard technique employed for acoustical simulations. However, the usual formulation assumes fixed boundary conditions in time, thus not allowing the dynamic geometry required for vocal tract models. The current solver proposes to solve an augmented wave equation that allows each point in space to transition between fully solid (wall) and fully open (air) states, via a scalar parameter field $0 \leq \beta(\mathbf{x},t) \leq 1$. The parameter is changed smoothly at time-scales larger than the system's main vibrational time-periods. Specifically, the technique solves the following equations for the pressure, $p(\mathbf{x},t)$ and particle velocity, $\mathbf{v}(\mathbf{x},t)$ in the interior of the domain:

$$\frac{\partial p}{\partial t} + (1-\beta)\, p = -\rho c^2 \nabla \cdot \boldsymbol{v} \tag{1}$$

$$\beta \frac{\partial \boldsymbol{v}}{\partial t} + (1-\beta)\boldsymbol{v} = -\beta^2 \frac{\nabla p}{\rho} + (1-\beta)\boldsymbol{v_b} \quad (2)$$

The equation amounts to linear interpolation between the standard wave equation when β=1 (air), and enforcing some prescribed velocity boundary condition $\mathbf{v}=\mathbf{v}_b$ when β=0 (boundary). For intermediate values of β, the affected region acts as partly reflective and partly transmissive. The above equations are discretized and solved numerically similarly to standard FDTD solvers, using second-order accurate spatial and temporal derivatives with velocities sampled on a staggered grid. The β field is sampled at cell centers and PML (6 layers) is employed at the edges of the domain to absorb outgoing radiation. Since we use an explicit scheme for time integration, the time-step Δt and spatial cell size Δx must obey the related Courant–Friedrichs–Lewy (CFL) stability condition in two dimensions: $\Delta t < \Delta x/\sqrt{(c^2)}$, where c is the speed of sound in the medium.

Due to the CFL condition, the required operations for producing a fixed duration of audio scale as Δx^3 in two dimensions and Δx^4 in three dimensions. This is a key challenge for fast speech simulations, which require millimeter-resolution grids or less in order to model the smooth surfaces and narrow channels involved in vocal tracts. They also enforce update rates much higher than 20 kHz, requiring appropriate low-pass filtering to remove unwanted ultrasonic oscillations.

B. GPU IMPLEMENTATION

The 2D pressure and velocity fields are represented as two dimensional value arrays, natively supported by GPUs as image textures. Each index in these arrays only requires access to the nearest neighbors in order to compute the value for the next time-step. This access-pattern naturally fits programmable shaders supported by most modern GPUs. The system operates within the OpenGL graphics pipeline, ensuring high utilization of all available Streaming Multiprocessors (lightweight GPU cores).

The fields are packed into a 32-bit floating point RGBA (Red-Green-Blue-Alpha) texture, with simulation cells represented as pixels in the texture. For each pixel, we store the pressure at the cell center p along with the velocity component v_x on the center of the right edge of the cell and velocity component v_y on the center of the top edge of the cell. This corresponds to a staggered Yee grid, and the values are stored in the R, G and B fields of the pixel respectively. The fourth (A) channel is used for dynamic information such as the cell's current β value, the strength of PML (for absorption at domain edges), and whether or not the cell has incoming flow velocity due to glottal excitation.

In each solver step, fields for the prior two time-steps and current step are read to compute and write into the predicted field for the next step. Fields for all four steps are laid out as a 2 × 2 grid on a single 2D "billboard" texture. Storing them in one large 2D array improves memory throughput: we bind the billboard texture to the frame-buffer once at the start of simulation, obviating costly rebinds at every time-step. At each step, the solver renders to the "next" field in the billboard while reading from the other three, followed by a texture barrier synchronization to ensure all write operations are actually committed to memory, and then update pointers circularly so that "next" becomes "current" and so on.

In addition to extremely fast computation, which can reach real-time performances under specific conditions (see Section 3), one of the advantages of this approach is direct sound propagation visualization. Since pressure and velocity are computed and saved as pixels on the texture, their values can be directly interpreted as colors and effortlessly rendered on screen. This

turns the synthesizer into a valuable tool to observe the acoustics of simplified vocal tracts, including the interaction between fundamental and transverse modes.

C. BOUNDARY HANDLING

The use of the β field in equations (1) and (2) represents the reflective behavior typical of walls. Varying the field models the transition from air to wall cells without affecting the stability of the system. Through this mechanism, it is possible to model moving vocal tracts and synthesize dynamic utterances, like diphthongs.

However, when simulating small fluid-filled ducts, like the vocal tract, fully reflective boundaries do not produce realistic results. In the proximity of walls, the laminar flow described by standard wave equations becomes physically inaccurate [18], since thermal and vortical diffusion interfere with ideal propagation. In the specific case of vowel synthesis, techniques based on complex acoustic impedance have been employed to precisely calculate frequency dependent visco-thermal losses arising at simulation boundaries [7, 19].

The acoustic model we designed is based on a time domain solver and, as a consequence, it cannot employ analogous complex impedance methods. Similarly to what Takemoto and Mokhtari [20] proposed, we adapted the local reactive boundary approach (designed for 2D room acoustics by Yokota et al. [21]) to at least partially take into account wall losses. Boundary conditions are enforced via $\mathbf{v}_w$, the velocity vector going to or coming from a wall:

$$\boldsymbol{v}_w = \frac{p_w}{Z_n}\boldsymbol{n} \tag{3}$$

$$Z_n = \rho c \frac{1+\sqrt{1-\alpha_n}}{1-\sqrt{1-\alpha_n}} \tag{4}$$

In equation (3), p_w is the current pressure value calculated on the air cell in front of the wall, while **n** is the unit vector normal to the wall and directed towards the wall itself. Pressure values inside wall cells are not utilized by the solver, since velocities on the boundaries depend only on p_w. Z_n is the normal acoustic impedance, as defined in equation (4). The normal acoustic impedance value depends on the reflection coefficient α, which is the actual input parameter used to define the wall's behavior. Equation (3) is then combined with equation (2) by means of setting $\mathbf{v}_b=\mathbf{v}_w$ for every velocity vector (v_x and v_y) that lies on an edge shared between an air cell and a wall cell. The equivalent boundary admittance can be obtained with:

$$\mu = \frac{\rho c}{Z_n} \tag{5}$$

This approach only takes into account the effects of the real part of the acoustic impedance (i.e., resistance), which affect the bandwidth of the transfer function formants [20].

D. EXCITATION MODELS

To simulate speech, articulatory synthesizers require a glottal excitation that acts as a source for the acoustic simulation. Initially, these models were based on the linear source-filter theory, which posits that speech production is a two-step feed-forward process: a sound source is generated from the glottis, and the articulators in the vocal tract shape this waveform similarly to a resonant filter [22]. However, this model is predicated on the underlying assumption that the

source and the filter are independent from each other, a premise that has been challenged in recent work [23]. For this reason, we chose to focus on self-oscillating glottal models of the vocal folds for our AV synthesizer over parametric models. The simulation of flow-induced oscillations in these models helps characterize the interaction between the glottal model and the resonant vocal tract, capturing the nonlinear coupling between the two components. Parametric glottal flow models, though computationally lightweight, do not allow for this direct coupling [24].

We developed two different self-oscillating glottal models to include in our synthesizer: a two mass model and a body-cover model. Both are based on lumped-elements, as they approximate the biomechanical structure of the vocal folds using discrete mass-spring elements, instead of solving it through continuum mechanics. This approach provides a fairly light computational load that allows the excitation of the 2D wave propagation solver even when running in real-time.

The two mass model was originally introduced by Ishizaka and Flanagan [25] and represents each vocal fold as a structure of two separate mass elements (i.e., superior and inferior mass), connected by a spring and moving in the medial-lateral direction only. Our implementation is based on the work of Sondhi et al [26], as it is computationally cheaper and has a more consistent discretization of the differential equations.

The body-cover model [27] introduces a separation between the core “body” of the vocal folds and the outer “cover” layers. An extra mass is added to represent the muscle body, with two coupled masses for the outer layers. This enables the model to represent the vertical phase difference of the surface wave more accurately than the two mass model, at the cost of using a more complicated solver based on a fourth-order Runge-Kutta method.

Both the two mass and the body-cover model are implemented on the GPU as a shader. They are coupled with the propagation model via the back propagating pressure coming from the simulation domain and via the size of the laryngeal cavity (i.e., the first segment of the vocal tract contour). The back propagating pressure is directly sampled from the texture where pressure is saved, while the size of the cavity is passed as a parameter every time the vocal tract geometry is changed.

3. 2D VOWEL AREA FUNCTIONS

The AV synthesizer has been employed to synthesize static vowel sounds, using the area functions published by Story in 2008 [28]. This dataset contains 11 different vocal tract shapes, each represented as an area function that describes a cylindrical tube with varying radii. In particular, we focused on the 3 corner vowels, /a/, /i/, /u/, since they provide a good variety in terms of tube shapes (i.e, size and position of constrictions) and lengths. Static vowels are the simplest challenge in AV synthesis, hence they represent a good first step to assess the performance of our system. Furthermore, Story's vowels have been studied by other researchers whose results are available for comparison. As described in this and the next section, our evaluation capitalizes on these useful data.

A. FROM 3D TO 2D

A direct way of representing an area function as a 2D domain is composed of 2 steps: first, a 3D tube composed of juxtaposed cylindrical segments is built from the area function; then a midsaggital plane is intersected with the tube to extract a 2D contour. The distance between the two contour lines represents the varying diameter of the 3D tube, which remains constant within

each segment. The resulting shape is open at both the extremes, with one end separated by the diameter of the glottal opening and the other by the size of the mouth opening.

A discretization step is then needed when contours are fed into a 2D acoustic model, along both the x axis (tube's length) and the y axis (tube's width). In general, the discretization step is the same for both the axes and it is chosen according to the size of the constrictions included in the simulated vocal tracts (i.e., smallest diameters). In the case of Story's vowels, the area functions are characterized by constrictions that reach the order of 4 mm (4.37 mm for /a/, 6.67 mm for /i/, 4.22 mm for /u/) and lengths between 16.9 cm for /i/ to 19.34 cm for /u/. These values allow for real-time synthesis of the vowels using our AV synthesizer. The system is equipped with an Nvidia GeForce GTX Titan X video card, that supports real-time simulations in domains sized up to 88 x 30 cells, at a sample rate of 220.5 kHz; the resulting spatial resolution is 2.24 mm per cell along both x and y directions, when the speed of sound (c) is set to 350 m/s (standard speed of sound within the vocal tract). This is enough to fit the entire length of the 2D contours and represent the tube constrictions with at least 2 cells.

However, such a direct 2D representation of area functions is inadequate, since it does not preserve the impedance mismatch between consecutive tube segments of the related 3D tubes. A solution to recover the original 3D acoustics has been presented by Arnela and Guasch [2] and is based on the modification of the diameters of the direct 2D contours. In particular, in the new contours the ratio between the diameters of two consecutive segments is equal to the ratio between the two tube's areas in the same position. The expansion ratio mechanism uses as reference the diameter corresponding to the largest area of the tube, whose size is set to match the biggest transverse mode found in the tube itself. Given the quality of the results obtained by Arnela and Guasch when used with a 2D FEM [2], we chose this technique to represent area functions as contours in our 2D FDTD acoustic model.

Although mathematically very simple, this approach comes with a drawback. The new expansion ratios determine a significant reduction of the size of the diameters, in some cases close to an order of magnitude; when applied to Story's vowels, the smallest constrictions shrink for /a/ from 4.37 mm to 0.47 mm, for /i/ from 6.67 mm to 1.47 mm and for /u/ from 4.22 mm to 0.56 mm. This requires running the simulation at much higher spatio-temporal resolutions, which currently make it infeasible to achieve real-time performance.

B. SYMMETRIC AND ASYMMETRIC DISCRETIZATION

The discretization step may introduce issues regarding tube symmetry and length. The original area functions represent a set of 3D tubes characterized by perfect radial symmetry that translates into specular 2D contours. However, the spatial resolution of our acoustic model is constant (i.e., each cell has the same size); this means that, when discretized, the best approximation of some diameters may consist of an even number of cells, breaking the specularity of the shape. This artifact causes unwanted acoustic effects, in terms of spurious antiresonances in the high part of the tube's transfer function. Furthermore, each asymmetric segment produces an artificial extension of the midline length of the contour and a consequent shift of the formants. While the lengthening can be adjusted by uniformly diminishing the length of each segment, the transfer function of an asymmetric domain will always display antiresonances.

An alternative discretization approach consists of forcing the symmetry of the 2D contour lines. In this case, the diameter of each segment is approximated with an odd number of cells. This prevents the onset of spurious antiresonances, but produces a doubling of the maximum

discretization error (i.e., up to the chosen spatial resolution), which may affect the position of the formants.

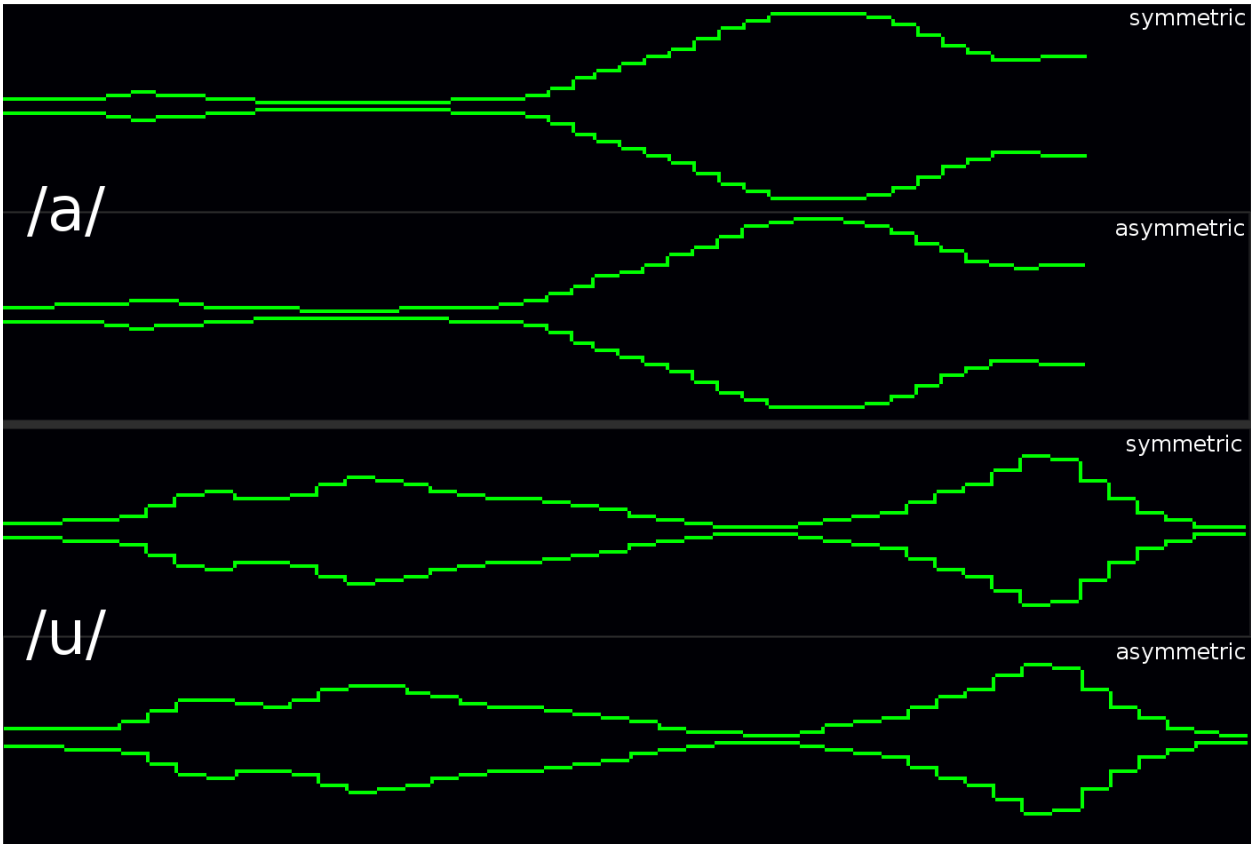

Figure 1. Symmetric and asymmetric 2D tube discretization examples for vowel /a/ (top) and vowel /u/ (bottom) (glottal opening on the left, mouth on the right). At the spatial resolution chosen for these examples (0.56 mm), it is possible to see how differently constrictions are affected by the two discretization approaches.

Neither of these approaches can provide an optimal representation of the tubes. However, in both cases the acoustic effects of the reported imprecisions are smaller and reduce further with an increase in the resolution of the simulation. For the preliminary evaluation of our synthesizer, we compare two versions of 2D contours for each vowel: one *symmetric case* and one *asymmetric case* (Figure 1). The acoustic outcomes of the two cases are compared in the next section, including investigation of different spatio-temporal resolutions.

4. TRANSFER FUNCTIONS

We focused on the analysis of the transfer functions of the 2D vocal tracts obtained from the area functions. To do so, we first computed the impulse response of the system for each corner vowel, by means of replacing the glottal excitation with a short pulse generator. We then applied an FFT to switch to a frequency-domain representation of the original signal. The computed transfer functions were compared with the results of a 3D FEM simulation of the same area functions, previously presented by Arnela and Guasch [2]. This model is extremely accurate and its results can be considered a good approximation of the real acoustics of the tubes. In the following subsections, the 2D synthesis parameters chosen to match those of the reference 3D simulations will be described, then the vocal tract transfer functions will be presented and briefly analyzed.

A. SIMULATION PARAMETERS

Three different resolutions were chosen to be included in our tests: 0.56 mm (882 kHz), 0.28 mm (1764 kHz) and 0.18 mm (2646 kHz). These values (from now on referred to as *low*, *medium* and *high resolution*) produce interesting discretization cases, regarding in particular the number of cells used to represent the width of the constrictions. At low resolution, the 0.47 mm pharyngeal constriction of /a/ (the narrowest constriction of the dataset, showed in Figure 1-top) is roughly approximated by only one cell. At medium resolution and when asymmetries are enabled, the same constriction is discretized by two cells. Finally, the high resolution case allows for a three cell discretization. A three cell discretization coincides with a theoretically good value for the smallest geometrical features of the model. This is also the upper limit of the available simulation resolutions, due to current implementation limitations of the system.

Following the procedure described in [2], a wall of excitation cells was used to seal the glottal opening at the leftmost end of the tube. A virtual microphone was placed around 3 mm down the mouth opening to record the impulse response; the exact position depends on the chosen resolution. The excitation pulse consisted of a band-passed impulse, with a bandwidth between 2 and 22050 Hz, to remove possible DC offsets as well as spurious ultrasonic oscillations. The passband filter is a combination of 2 sinc functions, with normalized transition bands equal to 0.001 and smoothed out by Hamming windows. An open-end termination was chosen for the tubes to correspond to the parameter configuration used by Arnela and Guasch in their simulations [2]. We did this by imposing a Dirichlet homogeneous boundary condition ($p=0$) at the mouth opening. Finally, the reflection coefficient was set to $\alpha=0.008$ for all the wall boundaries; this constant value was empirically calculated and, coupled with a speed of sound $c=350$ m/s and air density $\rho=1.14$ kg/m^3, determined a normal acoustic wall impedance $Z_n=199500$ kg/m^{2}s. The equivalent boundary admittance is $\mu=0.002$.

The impulse responses obtained using this set of parameters were recorded, keeping track of the computational times for all three resolutions. The measured time values to produce 50 ms of audio are: 3.0 s for the low resolution simulation (domain of 341 x 43 pixel cells); 12.4 s for medium resolution (690 x 97 cells); 29.8 s for high resolution (1049 x 146 cells). The numbers of pixel cells were optimized to fit any of the three vowels in the same domain. PML was discarded in the tests, since it is not needed in the open-end condition.

Table 1. Positional errors of the first 8 2D formants computed for /a/, with respect to 3D values.

Resolution	Discretization	F1	F2	F3	F4	F5	F6	F7	F8
Low	symmetric	-43.6 Hz -6.27 %	-33.6 Hz -3.15 %	60.0 Hz 1.98 %	-40.6 Hz -1.00 %	-126.7 Hz -2.51 %	-184.2 Hz -3.23 %	-589.5 Hz -8.34 %	-27.4 Hz -0.36 %
Low	asymmetric	-19.8 Hz -2.84%	18.7 Hz 1.75 %	2.6 Hz 0.08 %	-43.0 Hz -1.05 %	-206.3 Hz -4.09 %	-56.5 Hz -0.99 %	-329.6 Hz -4.66 %	-74.9 Hz -0.98 %
Medium	symmetric	-82.7 Hz -11.89 %	-308.2 Hz -28.86 %	-30.9 Hz -1.02 %	48.1 Hz 1.18 %	-30.2 Hz -0.59 %	-143.7 Hz -2.52 %	-26.9 Hz -0.38 %	-162.6 Hz -2.14 %
Medium	asymmetric	-5.1 Hz -0.74 %	18.3 Hz 1.71 %	-47.8 Hz -1.57 %	-75.2Hz -1.85 %	-107.4 Hz -2.13 %	-76.8Hz -1.34 %	-245.4 Hz -3.47 %	-38.9 Hz -0.51 %
High	symmetric	0.7 Hz 0.10 %	-26.6 Hz -2.49 %	-2.5 Hz -0.08 %	-16.4 Hz -0.40 %	-78.2 Hz -1.55 %	-68.9 Hz -1.20 %	-273.1 Hz -3.86 %	-138.8 Hz -1.82 %
High	asymmetric	5.6 Hz 0.37 %	-17.96 Hz 1.98 %	-37.5 Hz -0.80 %	-52.18 Hz -2.10 %	-68.8 Hz -2.61 %	-112.7 Hz -1.14 %	-193.0 Hz -4.20 %	-108.0 Hz -0.91 %

Table2. Positional errors of the first 8 2D formants computed for /i/, with respect to 3D values.

Resolution	Discretization	F1	F2	F3	F4	F5	F6	F7	F8
Low	symmetric	-0.2 Hz -0.09 %	34.7 Hz 1.64 %	-5.4 Hz -0.18 %	-107.6 Hz -2.60 %	-82.7 Hz -1.64 %	-94.1 Hz -1.63 %	12.3 Hz 0.18 %	-110.9 Hz -1.45 %
Low	asymmetric	4.8 Hz 1.84 %	-16.8 Hz -0.79 %	-21.8 Hz -0.72 %	-66.8 Hz -1.61 %	-134.9 Hz -2.68 %	-215.8 Hz -3.75 %	80.9 Hz 1.23 %	127.3 Hz 1.66 %
Medium	symmetric	-6.0 Hz -2.30 %	26.5 Hz 1.25 %	15.5 Hz 0.51 %	-47.5 Hz -1.14 %	-125.6 Hz -2.50 %	-173.5 Hz -3.01 %	-73.3 Hz -1.11 %	-54.1 Hz -0.70 %
Medium	asymmetric	1.5 Hz 0.60 %	15.4 Hz 0.73 %	8.0 Hz 0.26 %	-28.5 Hz -0.69 %	-145.4 Hz -2.89 %	-211.2 Hz -3.67 %	-73.9 Hz -1.12 %	0.3 Hz 0.00 %
High	symmetric	5.6 Hz -1.43 %	-17.96 Hz 1.03 %	-37.5 Hz 0.85 %	-52.18 Hz -0.99 %	-68.8 Hz -3.52 %	-112.7 Hz -4.16 %	-193.0 Hz -1.82 %	-108.0 Hz -0.72 %
High	asymmetric	3.3 Hz 1.26 %	14.9 Hz 0.70 %	7.2 Hz 0.23 %	-40.8 Hz -0.98 %	-129.9 Hz -2.58 %	-172.4 Hz -2.99 %	-62.0 Hz -0.94 %	-28.3 Hz -0.37 %

Table 3. Positional errors of the first 8 2D formants computed for /u/, with respect to 3D values.

Resolution	Discretization	F1	F2	F3	F4	F5	F6	F7	F8
Low	symmetric	-32.0 Hz -12.37 %	-124.2 Hz -16.41 %	16.7 Hz 0.73 %	16.7 Hz 0.46 %	12.5 Hz 0.29 %	-97.3 Hz -1.92 %	-39.5 Hz -0.64 %	-75.6 Hz -1.14 %
Low	asymmetric	-16.3 Hz -6.30 %	-70.0 Hz -9.25 %	9.5 Hz 0.42 %	-40.7 Hz -1.13 %	39.6 Hz 0.94 %	-192.9 Hz -3.81 %	-61.8 Hz -1.00 %	-17.6 Hz -0.26 %
Medium	symmetric	-50.0 Hz -19.31 %	-181.9 Hz -24.03 %	-0.9 Hz -0.04 %	-23.0 Hz -0.63 %	-110.6 Hz -2.65 %	-162.1 Hz -3.20 %	-55.0 Hz -0.89 %	-232.7 Hz -3.53 %
Medium	asymmetric	-12.3 Hz -4.77 %	-62.7 Hz -8.28 %	7.7 Hz 0.34 %	-26.3 Hz -0.73 %	-28.3 Hz -0.68 %	-172.9 Hz -3.41 %	-20.4 Hz -0.33 %	-69.3 Hz -1.05 %
High	symmetric	-18.9 Hz -7.30 %	-83.2 Hz -10.99 %	9.0 Hz 0.39 %	-34.9 Hz -0.96 %	-3.5 Hz -0.08 %	-196.4 Hz -3.88 %	-26.4 Hz -0.43 %	-52.5 Hz -0.79 %
High	asymmetric	-12.4 Hz -4.79 %	-63.9 Hz -8.45 %	7.6 Hz 0.33 %	-33.6 Hz -0.93 %	-21.2 Hz -0.50 %	-167.5 Hz -3.30 %	-28.4 Hz -0.46 %	-71.0 Hz -1.07 %

B. FORMANT ANALYSIS

We extracted the positions of the first 8 formants and compared them with the results from the 3D FEM using the vocal tract transfer functions obtained from the impulse responses. The exact 3D formant values can be found in Arnela's Ph.D dissertation [29]. Note, formant bandwidths were not taken into account. Table 1, 2 and 3 show the errors between the 2D and 3D computed vowel formants for /a/, /i/ and /u/, respectively. The tables present the results obtained using low, medium and high resolution combined with the 2 discretization approaches; each combination (i.e., row) includes both absolute and percent errors.

Overall, results show good agreement with errors that tend to stay below 4%. Vowels /a/ and /i/ display the best matches, while the first two formants of /u/ are always characterized by larger shifts of the position of the formants. As expected, increasing the resolution of the tubes shows a general improvement. This is especially true for the asymmetric cases; all errors go gradually down and for several formants get lower than 1%. This constant trend is particularly evident for /i/, while for /a/ and /u/, medium and high resolution produce very similar values, more accurate than the low resolution ones. The beneficial effects of increasing the resolution are less evident when using the symmetric discretization. All the symmetric tubes at medium

resolution display larger errors in the first two formants and, more generally, do not always provide an improvement compared to the low resolution cases. Furthermore, the low resolution symmetric case produces the most accurate formants for vowel /i/, with errors below 1%.

Even when choosing a fixed resolution, the asymmetric case generally seems to be a better representation of the original tube; formant positions are enhanced for all vowels, particularly in the low and medium resolution cases for /a/, as well as for /u/, across all 3 resolutions. The only 2 exceptions are the high resolution /a/ and the low resolution /i/. The former loses precision in F1, F3 and F4 when represented asymmetrically. The latter, as previously mentioned, produces exceptionally good results in the symmetric case. These are much more accurate than the ones from its asymmetric version, in particular for F1, F3 and F7.

5. CONCLUSIONS AND FUTURE WORK

The results of the tests suggest that the presented AV synthesizer is a promising tool, both in terms of precision and speed of computation. The reported formant's values are close to realistic 3D data, even when running at relatively coarse resolution. The measured computational times are extremely low, between 3 and 29.8s bringing them closer to 1D model simulation performance than to other 3D or 2D approaches (reference times can be found in [2]).

However, these tests should be considered as a preliminary evaluation of the system. The final goal of this work is fast human speech synthesis, capable of resembling natural voice more closely than a 1D synthesizer can. The idea is to aim for a better match of resonances and antiresonances found in vocal tracts above 5 kHz by means of simulating more realistic geometries (i.e., bent non-cylindrical tubes) with the ability to compute transverse modes in the added dimension. The testing of static area functions is a fundamental first step, but the radial symmetry of their geometrical representations rules out the possibility of testing the interaction between fundamental and transverse modes. In other words, in this scenario the system behaves similarly to a 1D model and does not take advantage of the higher dimensionality.

The next step for this research consists of assessing the performance of the system using more realistic vocal tracts. This posits the problem of how to represent arbitrary 3D geometries in 2D beyond a series of straight cylindrical tubes. While the curvature of a 3D vocal tract might be directly ported into a 2D contour, the two-dimensional representation of irregular cross-sections of non-symmetric tubes is still an open problem.

Our system shows very good time performance, but a boost is still needed to achieve real-time vowel synthesis. A first way to get closer to real-time performance consists of leveraging more advanced GPU technologies—in particular cards with higher memory bandwidth as that is our main bottleneck. Although a new generation of more powerful cards has already been released since the beginning of this study, none of these devices features a substantial upgrade in bandwidth specifications. A second approach to speed up the computation requires a different approach to process the 3D to 2D mapping of the vocal tract models to decrease the precision required by the 2D simulation. The geometrical transformation applied in this study to adjust the impedance mismatch of the 2D circular tube segments resulted in a consistent shrinking of vowels' constrictions, which in turn, made real-time performance challenging in the current study. In the search for more generic 3D to 2D geometrical mappings, it is worth targeting solutions that combine structural morphing with a parametric impedance representation, similar to 1D approaches, to avoid extreme deformation of the original mid-sagittal contours.

ACKNOWLEDGMENTS

This work is supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme and by the Natural Sciences and Engineering Research Council (NSERC) of Canada. We would also like to thank Nvidia Corporation for donating the graphics card used in this work.

REFERENCES

[1] Takemoto, Hironori, Parham Mokhtari, and Tatsuya Kitamura. "Acoustic analysis of the vocal tract during vowel production by finite-difference time-domain method." The Journal of the Acoustical Society of America 128.6 (2010): 3724-3738.

[2] Arnela, Marc, and Oriol Guasch. "Two-dimensional vocal tracts with three-dimensional behavior in the numerical generation of vowels." The Journal of the Acoustical Society of America 135.1 (2014): 369-379.

[3] Stevens, Kenneth N. Acoustic phonetics. Vol. 30. MIT press, 2000.

[4] Story, Brad H., Ingo R. Titze, and Eric A. Hoffman. "Vocal tract area functions from magnetic resonance imaging." The Journal of the Acoustical Society of America 100.1 (1996): 537-554.

[5] van den Doel, Kees, and Uri M. Ascher. "Real-time numerical solution of Webster's equation on a nonuniform grid." IEEE transactions on audio, speech, and language processing 16.6 (2008): 1163-1172.

[6] Birkholz, Peter, Dietmar Jackèl, and Bernd J. Kroger. "Simulation of losses due to turbulence in the time-varying vocal system." IEEE Transactions on Audio, Speech, and Language Processing 15.4 (2007): 1218-1226.

[7] Arnela, Marc, et al. "Influence of vocal tract geometry simplifications on the numerical simulation of vowel sounds." The Journal of the Acoustical Society of America 140.3 (2016): 1707-1718.

[8] Sondhi, M. M. "An improved vocal tract model." Proceedings of the 11th ICA. Paris, France (1983): 167-170.

[9] Mokhtari, Parham, Hironori Takemoto, and Tatsuya Kitamura. "Single-matrix formulation of a time domain acoustic model of the vocal tract with side branches." Speech Communication 50.3 (2008): 179-190.

[10] Ho, Julio C., Matías Zañartu, and George R. Wodicka. "An anatomically based, time-domain acoustic model of the subglottal system for speech production." The Journal of the Acoustical Society of America 129.3 (2011): 1531-1547.

[11] Howard, David M., Sten Ternström, and Matt Speed. "NATURAL VOICE SYNTHESIS: The potential relevance of high-frequency components." Proceedings of AVFA 9 (2009): 18-20.

[12] Mullen, Jack, David M. Howard, and Damian T. Murphy. "Waveguide physical modeling of vocal tract acoustics: flexible formant bandwidth control from increased model dimensionality." IEEE Transactions on Audio, Speech, and Language Processing 1 4.3 (2006): 964-971.

[13] Speed, Matt, Damian T. Murphy, and David M. Howard. "Characteristics of two-dimensional finite difference techniques for vocal tract analysis and voice synthesis." INTERSPEECH. 2009.

[14] Wang, Yuguang, et al. "Mandarin vowel synthesis based on 2D and 3D vocal tract model by finite-difference time-domain method." Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific. IEEE, 2012.

[15] Arnela, Marc, et al. "Finite element computation of diphthong sounds using tuned two-dimensional vocal tracts." Proc. of 7th Forum Acousticum (Kraków, Poland). 2014.

[16] Mullen, Jack, David M. Howard, and Damian T. Murphy. "Real-time dynamic a rticulations in the 2-D waveguide mesh vocal tract model." IEEE Transactions on Audio, Speech, and Language Processing 15.2 (2007): 577-585.

[17] Allen, Andrew, and Nikunj Raghuvanshi. "Aerophones in flatland: interactive wave simulation of wind instruments." ACM Transactions on Graphics (TOG) 34.4 (2015): 134.

[18] Bossart, R., N. Joly, and M. Bruneau. "Hybrid numerical and analytical solutions for acoustic boundary problems in thermo-viscous fluids." Journal of Sound and Vibration 263.1 (2003): 69-84.

[19] Fleischer, Mario, et al. "Formant frequencies and bandwidths of the vocal tract transfer function are affected by the mechanical impedance of the vocal tract wall." Biomechanics and modeling in mechanobiology 14.4 (2015): 719-733.

[20] Takemoto, Hironori, Parham Mokhtari, and Tatsuya Kitamura. "Acoustic analysis of the vocal tract during vowel production by finite-difference time-domain method." The Journal of the Acoustical Society of America 128.6 (2010): 3724-3738.

[21] Yokota, Takatoshi, Shinichi Sakamoto, and Hideki Tachibana. "Visualization of sound propagation and scattering in rooms." Acoustical science and technology 23.1 (2002): 40-46.

[22] Fant, Gunnar. Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations. Vol. 2. Walter de Gruyter (1971).

[23] Nonlinear Source–filter Coupling in Phonation: Theory.

[24] Fant, Gunnar, Johan Liljencrants, and Qi-guang Lin. "A four-parameter model of glottal flow." STL-QPSR 4.1985 (1985): 1-13.

[25] Ishizaka, Kenzo, and James L. Flanagan. "Synthesis of voiced sounds from a two‐mass model of the vocal cords." Bell system technical journal 51.6 (1972): 1233-1268.

[26] Sondhi, Man, and Juergen Schroeter. "A hybrid time-frequency domain articulatory speech synthesizer." IEEE Transactions on Acoustics, Speech, and Signal Processing 35.7 (1987): 955-967.

[27] Story, Brad H., and Ingo R. Titze. "Voice simulation with a body‐cover model of the vocal folds." The Journal of the Acoustical Society of America 97.2 (1995): 1249-1260.

[28] Story, Brad H. "Comparison of magnetic resonance imaging-based vocal tract area functions obtained from the same speaker in 1994 and 2002." The Journal of the Acoustical Society of America 123.1 (2008): 327-335.

[29] Arnela Coll, Marc. "Numerical production of vowels and diphthongs using finite element methods." Ph.D. Thesis. Universitat Ramon Llull (2015): 138.