Singing Like a Tenor without a Real Voice

Jochen Feitsch, Marco Strobel, and Christian Geiger

University of Applied Sciences Düsseldorf, Department of Media, Germany
{jochen.feitsch,marco.strobel,geiger}@fh-duesseldorf.de

Abstract. We describe a multimedia installation that provides users with the experience to sing like a tenor from the early 20th century. The user defines vowels with her mouth but does not produce sound. The mouth shape is recognized and tracked by a depth-sensing camera and synthesized using a dedicated sound analysis using formants. Arm gestures are recognized and used to determine pitch and volume of an artificially generated voice. This synthesized voice is additionally modified by acoustic filters to sound like a singing voice from an old gramophone. The installation allows to scan the user's face and to create an individual 3D model of a tenor character that is used to visualize the user performance.

1 Introduction

The purpose of this project is to develop a system that allows users to act like a tenor by moving their arms and lips but without real singing. Shaping vowels with her mouth and getting corresponding multimodal feedback creates a believable user experience. It is difficult to synthesize the human voice based on purely visual analysis of the mouth's shape. A possible solution is to restrict the recognition and synthesis to trackable vowels and use additional cues like arm gestures for other vowels and to generate pitch and volume of the singing voice.

The origin of music, and thus singing as part of it, has been researched for a long time. Charles Darwin advanced the theory that the human antecessors learned musical notes and rhythms to charm the other sex. According to evolutionary biologist Geoffrey Miller, music and dancing emerged from rituals symbolizing combat and hunting [6]. A different approach arose from the observation that every human culture has lullabies used by mothers to calm their children. Due to anthropologist Dean Falk singing provided sort of a remote maintenance for the helpless baby, as long as the mother stayed within hearing distance [3]. Singing apparently had an important impact in human evolution and cultural development. Nevertheless, nowadays there is a strong tendency towards consumption of externally generated music and few people are able to really experience the positive feeling of their own successful singing voice production. This is the motivation of the project presented in this paper: to provide users with a believable user experience of performing an aria. Although we know that it is not possible to exactly simulate the performance of a professional opera singer we aim to create an entertaining user experience with our work.

D. Reidsma, H. Katayose, and A. Nijholt (Eds.): ACE 2013, LNCS 8253, pp. 258–269, 2013.

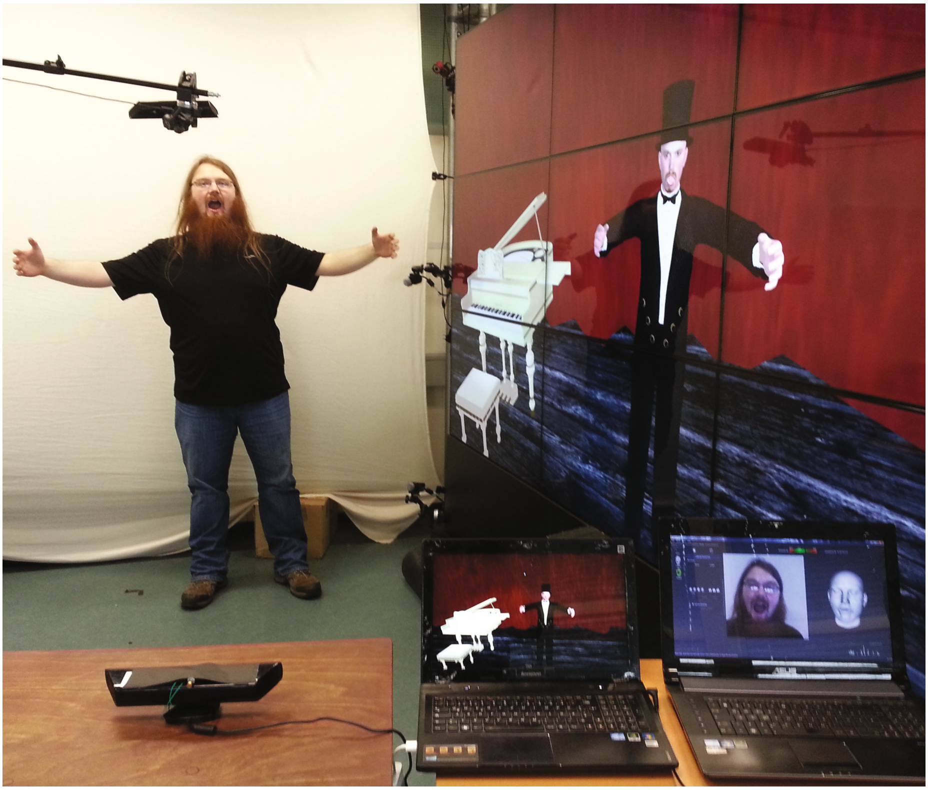

Fig. 1. System prototype

2 Related Work

Several projects studied the synthesis of sound with the mouth or with gestures. De Silva et al present a face tracking mouth controller [11]. The application example focused on a bioacoustics model of an avian syrinx that is simulated by a computer and controlled with this interface. The image-based recognition tracks the user's nostrils and mouth shape and maps this to parameters of a syrinx's model. This model is used to generate sound. At NIME 2003 Lyons et al presented a vision-based mouth interface that used facial action to control musical sound. A head-worn camera tracks mouth height, width and aspect ratio. These parameters are processed and used to control guitar effects, a keyboard and look sequences [5]. Closely related to our approach is the "Artificial Singing" project, a device that controls a singing synthesizer with mouth movements that are tracked by a web camera. The user's mouth is tracked and recognized mouth parameters like width, height, opening, rounding, etc. are mapped to a synthesizers parameters like pitch, loudness, vibrato, etc [4]. Recently, Cheng and Huang published an advanced mouth tracking approach that combines real-time mouth tracking and 3D reconstruction [2]. The synthesis of singing is an ambitious area of research with a long tradition and the human singing voice is a most complex instrument to be synthesized. A good overview is presented in [10], [12]. The creation and control of a 3D avatar has also been discussed in a number of projects. FaceGen (www.facegen.com) is a prominent base technology used in many AAA game productions. Many musical interfaces apply RGB-D cameras like Kinect for controlling sound synthesis (e.g., [7]) and we also chose this device for our purposes. While most projects focus on either body tracking or facial tracking, we combine both tracking methods to give the user a better overall experience. The body tracking used to control the movement of a virtual tenor character and aims at giving the user the impression of "being" that character. Thus, body movement is an essential part of our performance and also used for sound synthesis by controlling pitch and volume. Mouth gestures are used to control the sound synthesis by simply shaping the desired vowel with the user's mouth.

3 System Overview

The installation consists of a 3x3 video wall with 46" monitors, one Microsoft Kinect for full body skeleton tracking and a Primesense Carmine (or another Kinect) sensor for facial tracking. Figure 2 provides an overview of the system. The processing is done by two computers connected via Ethernet. One is responsible for operating the hardware system for facial tracking and the synthesizer modules while the other runs the main application with a software-based skeleton tracking system and the hardware sensors. The sensor for skeleton tracking is placed in about two to three meters distance to the user. The facial sensor is positioned and fixed hanging in the air in front of and above the user, looking at the user in about a 20°C angle. This minimizes the interference of the two depth cameras by reducing the overlapping region of both cameras. One computer processes all tracking data and uses this data also to fully animate a virtual 3D character on a theater stage. The character's head can be adapted towards the user's face by optionally using an integrated face generation module. To use this feature the user takes a picture of him/herself and the system creates a corresponding textured 3D face model. The user can also choose to modify several face model parameters to modify the final look of the 3D character. After this face generation step the user starts the performance mode and positions him/herself in front go the two depth cams (see fig 2). By moving the arms und shaping vowels with the mouth, the user can not only control the tenor's arms and facial expression, but also produce a singing voice and thus feel as being in an opera performance. The user's goal during the performance is to reproduce the original singing voice of an aria for a given background music as accurately as possible. This is similar to the well-known Karaoke music game but in our installation the user does not really sing with her own voice. With an increasing positive rating of the user's performance the 3D character's face morphs from the user's face to a virtual Enrico Caruso, a famous tenor from the 20th century.

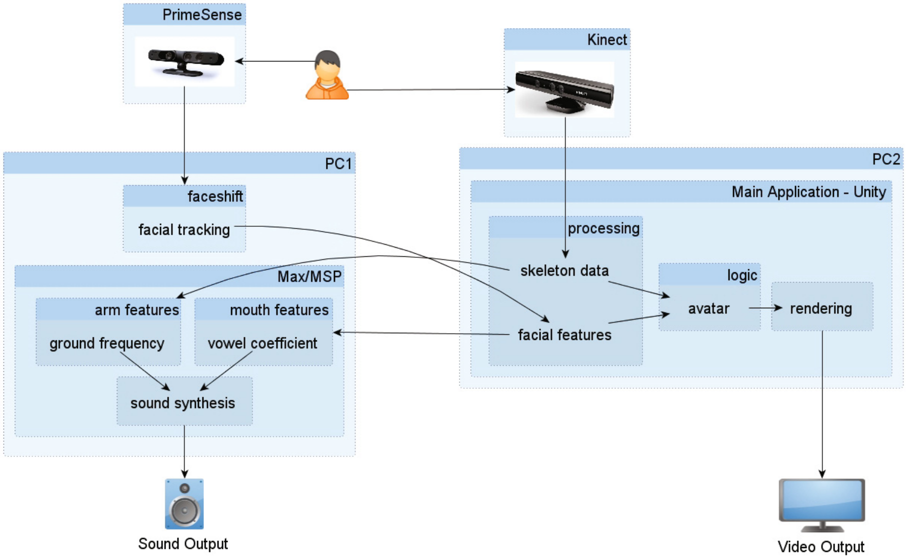

Fig. 2. System overview

4 Avatars Generation and Customization

To provide users with an individual performance experience one of the features of this project is to create a user-defined 3D avatar with the user's face. This was realized by integrating the FaceGen SDK (www.facegen.com) for the Unity3D Engine. To use this feature the application switches to "avatar customization" mode (3). The user takes a two-dimensional picture of his/her face using an HD camera. In a sequent step this image is mapped as texture on the user is represented with her image and asked to place eleven fiducial markers on the photograph to identify the facial structure. Attempts to automate this last step have failed to provide an acceptable result compared to manually set the markers (see fig 3, pic 2). After the last marker is set manually and the user finishes the manual mapping step the system will calculate a corresponding 3D mesh representation of the user's face in texture and shape. This model is used to create the user-defined tenor character (see fig 3, pic 3).

Once this process is completed the user has a number of options to further modify the avatar using a simple user interface with virtual sliders to adjust different facial parameters. These options include to make the avatar look older or younger, more male or female or to mix up racial features making the avatar look more afro-american, eastern asian, southern asian or european (see fig 3, pic 4). It may also be possible to modify the facial structure towards a more (or lesser) asymmetric face model or to modify characteristic face features from a "generic look"-level to a "caricature"-level.

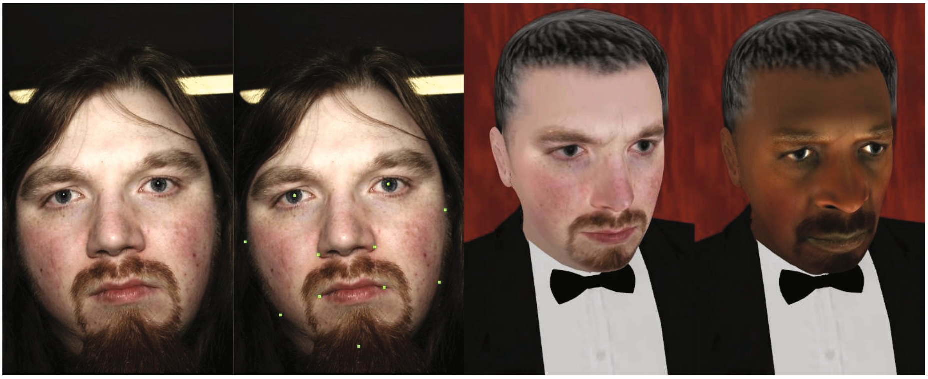

Fig. 3. Face customization process

5 User Tracking

Once the user decided that she likes her avatar look she switches to "performance mode". In this mode she is being tracked by two depth sensors (e.g. Microsoft Kinect) for full body skeleton tracking for facial motion tracking. The tracking quality of two sensors decreases significantly if the sensors are positioned with a larger the overlap of the camera's structured infra-red pattern. This problem was solved by shielding the field of view of both sensors from each other as well as possible by having the facial tracker hang in the air and only aim at the face while the full body tracker is further away capturing the full picture. In addition to solve the last bit of interference the "Shake'n'Sense" [1] approach was applied. We attached a small vibrating servo motor to one Kinect sensor and introduces artificial motion blur through vibration and eliminates crosstalk.

5.1 Full Body Tracking

The full body skeleton tracking takes all the provided joint data from the Kinect and maps it to the avatar's skeleton, either with full body tracking enabled or only upper body tracking. In addition to this hand, elbow and shoulder joints are processed to control the volume and pitch parameters of the generated singing voice. The joints' position values are used to calculate the arm stretch by dividing the shoulder-to-hand length (current stretch) to the sum of shoulder-to-elbow and elbow-to-hand length (maximum stretch), providing a normalized volume range from approximately 0 (no stretch) to 1 (full stretch). Furthermore, the height value of the shoulder's position (y-axis) is subtracted from the height value of the hand's position and this sum is divided by the maximum arm stretch to get a normalized range of approximately -1 (lowest height value of the hand) to 1 (highest possible height) for the pitch value. These calculations are done separately for each side of the body and the largest value is selected.

5.2 Facial Tracking

The current prototype of our facial motion tracking system utilizes faceshift (http://www.faceshift.com), a software originally used for marker-less facial motion capturing and 3D character face animation. Previous prototypes used a 2D face tracking using OpenCV and the face tracking provided by the Microsofts Kinect SDK 1.5+ but were too inaccurate for our purposes. The integration of faceshift resulted in a much better performance. To get reliable data from the faceshift tracking module the user has to create a profile that is efficiently used during the performance. This profile has to be created in the software's training mode by capturing a few default facial expressions in order to create a suitable 3D representation of the user's facial structure. Although an existing profile can be used for new user's the tracking performance may be less accurate without an individual profile.

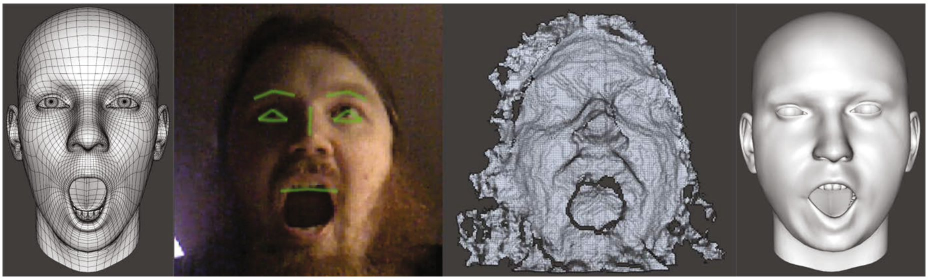

Fig. 4. Training of a user-specific profile for face tracking using face shift

After the calibration step is completed the user proceeds to "performance mode". If the user's face is recognized in this mode the camera image, the corresponding 3D representation and all the facial parameters are captured and tracked. During this mode a network server provides a streaming service for clients that can connect via the network. The sound synthesis module described in section 6 connects to this service, streams the tracking data from a custom protocol consisting of head pose information, blend shapes (also called coefficients), eye gaze and additional manually set virtual markers. The coefficients and marker positions are sent directly to the audio synthesizer where they are processed for sound generation. Furthermore the head pose, blend shapes and eye gaze are used to animate the avatar's facial features in real time. The head pose is used to rotate the neck bone and the eye gaze to rotate the specific eyes while the blend shapes are used to change the look of the avatar's face using the morph capabilities of the FaceGen SDK. To make this work correctly with our avatar customization we adjusted our tenor model to work well with both the facial tracking data from faceshift and the fitting and morphing system of FaceGen. This was done by adjusting the basic face shape to the base model used by FaceGen, creating custom blend shapes that mimic the blend shapes of faceshift and finally converting this into a suitable model base. To give the user feedback

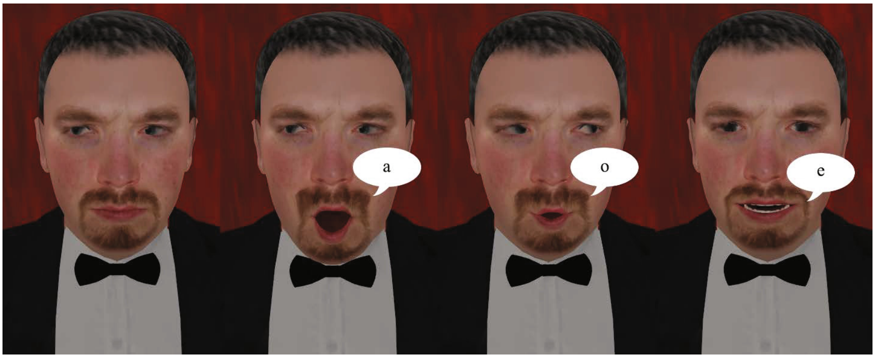

Fig. 5. Facial animation shots of vowel synthesis

on the quality of her performance we've also created functionality to morph between the user's customized 3D head and a 3D head created from a real opera singer's photograph (e.g. Enrico Caruso). The better she performs singing an pre-selected song like ”Ave Maria” the more she will turn into a virtual tenor character like Caruso. This is done using the same base structure for both heads and simply interpolating between the 3D mesh data of the two models.

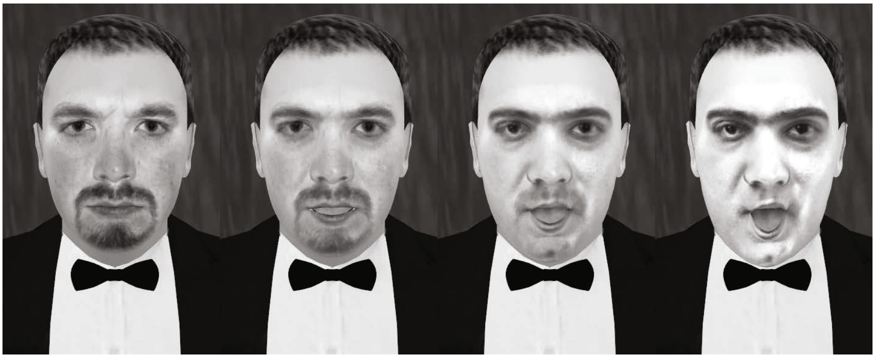

Fig. 6. Morphing between custom user head and Caruso

To make it easier to calibrate the user's body and face gestures we developed a small Android app that functions as a control panel for the different calibration modules. This makes it possible to send networked commands to start or stop tracking, calibrate the neutral position and angle and whether to treat the head position relative or absolute simply from or nearby the user's place on the stage and not directly in front of the laptops.

6 Sound Synthesis

The sound synthesis of this work is based on formants which we introduce in this section.

6.1 Preface

The analysis technique used for transforming signals from the time domain to the frequency domain to generate the illustrations for this paper is somewhat inaccurate. This is due to the fact, that most illustrations are snapshots taken while performing with the installation, and thus the transformations had to be operated in real time. A sample is shown in Fig. 7 b): only the peaks of the "hills" represent contained sine waves, the remainder is to be seen as "overshoot". In Fig. 7 a) the corresponding correct analysis is shown schematically. Only contained frequencies display a bar in the graph. Both figures represent an analysis of the same signal and are to been seen equal in this paper.

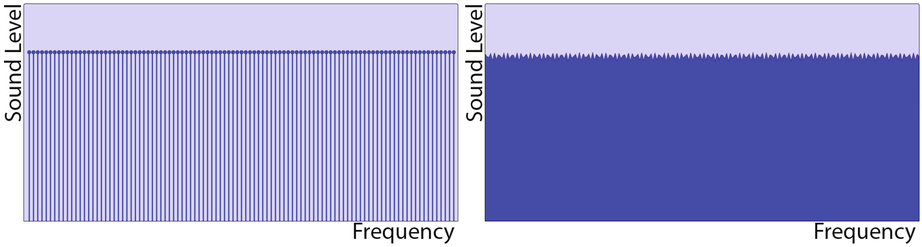

Fig. 7. Fundamental frequency 110Hz a) schematic, b) perfomance snapshot

6.2 Formants

A formant is a concentration of acoustical energy in a specific frequency range, that means these frequency ranges are emphasized in relation to other frequency ranges. The middle of this frequency range contains the most energy, the frequencies to both sides have gradually decreasing energy levels. Talking about a formant at 730 Hz refers to the formant with the mid-frequency of 730 Hz. Fig. 8 displays a signal in the frequency domain that features three formants, 8 a) serves for better understanding of the allocation of the frequencies. The middle of these formants is marked with the blue arrows (in this case at 730 Hz, 1090 Hz and 2440 Hz). The signal is based on the signal shown in Fig. 7. The relevant filters used to generate the formants are outlined by the colored curves.

Formants are an essential part of the characteristic sound of an instrument (there are other factors as well). Even though two sounds might have the same fundamental frequency, they can sound differently. If a piano and a guitar play the same note, they are still clearly distinguishable. In this sense, the human voice is an instrument as well, and its different vowels can be distinguished because of their different formants.

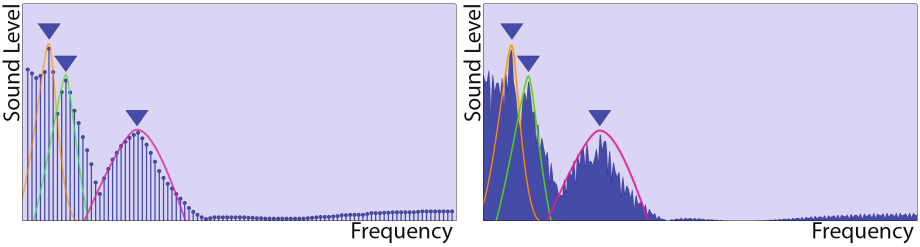

Fig. 8. Vowel "A" formants at 110Hz a) schematic, b) perfomance snapshot

6.3 Synthesis

The tenor's voice is synthesized by using a vowel synthesis via formants. Max/MSP is used for audio processing and calculations, while data is sent as OSC packages from one program to another. The user determines the target fundamental pitch of the singing with her arms (see chapter Mapping). Based on the target pitch of the tenor's voice, multiple sine waves are added up. These sine waves have pitches that are integer multiples of the target pitch, that means if the target pitch is 110 Hz, the added sine waves are at 220 Hz, 330 Hz etc. In this synthesis, the upper limit was defined at 12 kHz, as higher frequencies are not relevant for the characteristics of the human voice. Next, the signal is modulated by several signals that also depend on the fundamental pitch, but are randomized to a certain amount to create a preferably non-artificial impression of the end result. This modulation adds a vibrato effect, that means a periodical small change of the target pitch, and slightly manipulates the amplitude of the signal to simulate a human singing voice, assuming that a singer can't produce a perfectly constant tone. Three band pass filters are used to create the target vowel's characteristic formants. These formants have the property of being more or less independent of the generated fundamental pitch. Fig. 7 shows the unfiltered signal with a fundamental frequency of 110 Hz and it's harmonic frequencies at 220 Hz, 330 Hz etc. In Fig. 8 this signal was filtered by three bandpass filters (outlined by the colored curves) to "cut out" the characteristic formants and thus create the apparent envelope curve with three summits (in this example at 730 Hz, 1090 Hz and 2440 Hz marked by the blue arrows to form the vowel "A"). The example in Fig. 10 a) shows the result of the same process with a fundamental frequency of 220 Hz and it's harmonic frequencies at 440 Hz, 660 Hz etc. The resulting formants are the same as the one at a fundamental frequency of 110 Hz, so the resulting vowel is still "A". This shows that formants are more or less independent from the fundamental frequency. As an evaluation of our synthesized voice, we separated the vowel "A" from a recorded aria in Fig. 10 b). The real singing has three apparent formants at the beginning of the frequency spectrum that have similar middle frequencies to those that are created in our system. The difference in sound level and the other amplified regions form the singer's personal timbre. The selected formant frequencies were taken from [8]. Most other sources give only two formants, as two of them are sufficient for the perception of a vowel. However, a third formant turned out to be useful to make the result more human. The synthesis actually uses an additional fourth formant at 2,7 kHz. That formant is both independent of the fundamental frequency and the target vowel. It's called "singers' formant" and is only visible in the frequency spectrum of trained singers. This formant is essential when singing with an orchestra and allows professional singers to be heard without further amplification, as the peak frequency range of an orchestra is much lower than 2,7 kHz. After the formant filtering process, several additional filters are used to achieve a 20th century gramophone like sound. In a final step, vowel-dependent amplitude variations are reduced by normalizing and compressing the signal. Additionally, an optional reverb effect can be added to simulate the acoustic of an opera.

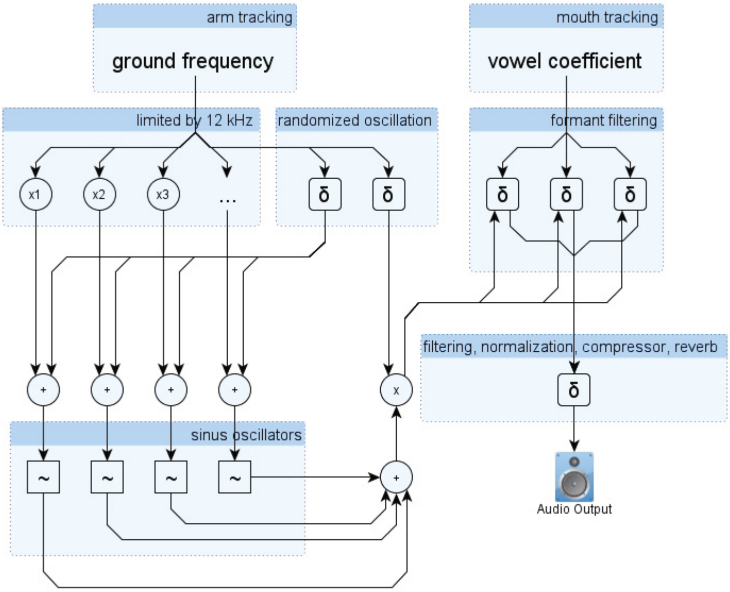

Fig. 9. Sound synthesis using Max/MSP

7 Mapping

The user controls pitch and volume of the synthesized singing voice with her arms. The volume is determined by the user's arm stretch and is directly mapped to the system's gain control. The arm-height is used to calculate the target pitch by using the change of arm-height over time. This method was selected to make it easier and more enjoyable to use the system (previous methods used fixed positions for determining the pitch). The simplest implementation uses only three commands: "no change", "change upwards" and "change downwards" (finer steps can be chosen to make playing more precisely, for example "big change upwards", "small change upwards" etc). If the melody's next note is above the current pitch, the user has to move her arms upward, if the next note is below, she has to move her arms downwards. The system then automatically chooses the right note to play the correct melody. If the user makes a ”wrong” move, the system either stays at the currently sung pitch (”no change”), or uses a mapping-table to determine the next higher respectively next lower pitch. There are a total of 25 MIDI-pitches that can be sung, from MIDI-pitch 41 (≈ 87 Hz) to MIDI-pitch 65 (≈ 349 Hz). The actual singable pitches are limited by the song’s current key (all keys need to be predefined for the whole song). The program triggers events in accordance to the song’s beat to change the currently used mapping table. The basis of these mapping tables has always the keynote ”C”, that is transposed later, with a variety of scales: C-Major, C-Minor, C diminished and C augmented, as well with optionally added seventh, major seventh, sixth or diminished fifth. In addition, the system distinguishes whether the song’s measure is in beat 1 or 4, or in beat 2 or 3. In case one, only pitch values from the current key’s chord can be played, in case two the whole scale can be selected. These limitations help the user to create a melody that sounds always more or less suitable for the currently chosen song’s background music, even if she is not performing the correct arm movements for singing the song’s melody. This allows for more or less harmonic improvisation if the user does not want to sing the original song. The desired vowel is given via the user’s mouth shape. 23 of the 46 parameters provided by the face tracking module are relevant for the area around the user’s mouth. Therefore they are used as input nodes for a neural network trained to identify vowels by a given mouth shape. The neural network uses resilient back-propagation. This is a fast back-propagation algorithm that allows to train new data to the network in a matter of seconds [9]. The output of the neural network consists of 4 probability parameters, ”Closed Mouth”, ”A” [a:], ”E” [e:] and ”O” [o:]. These four values are monitored, and the highest determines the current vowel, sending the frequency values to the representing formant filters. To give the user the opportunity of choosing another set of vowels ”Ä [ε:]”, ”I” [i:] and ”U” [u:], the option to switch between the two sets is activated by moving the eyebrows up and down or by means of a dedicated hand / finger gesture.

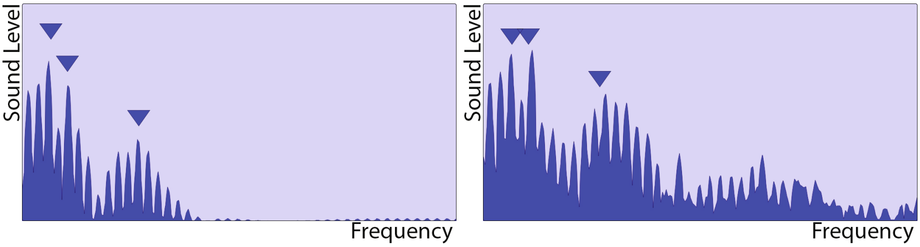

Fig. 10. Vowel ”A” at approximately 220Hz a) synthesized, b) sung by real tenor

8 Conclusion

The current prototype is functional, i.e., the individual components work but are still a bit limited concerning the tracking quality and the range of expressively synthesized vocals. The integration into a final system prototype is complete but we are still working on an expressive performance mode. Currently we use an old gramophone induced by an exciter to produce an enjoyable auditive experience for both the user and the audience. We also evaluate the combination of vibrational feedback and bone conduction headphones to enhance the user experience and support the immersion of the user during the performance. This should further increase the intended believability and enjoyment of being an opera tenor during a performance.

References

1. Butler, A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., Kim, D.: Shake'n'Sense: Reducing Interference for Overlapping Structured Light Depth Cameras. In: ACM SIGCHI Conference on Human Factors in Computing Systems (2012)
2. Cheng, J., Huang, P.: Real-Time Mouth Tracking and 3D reconstruction. In: Int. Conf. on Image and Signal Processing (2010)
3. Falk, D.: Finding our Tongues: Mothers, Infants, and the Origins of Language. Basic Books (2009)
4. Hapipis, A., Miranda, E.R.: Artificial Singing with a webcam mouth-controller. In: Int Conf. on Sound and Music Computing (2005)
5. Lyons, M., Haehnel, M., Tetsutani, N.: Designing, Playing, and Performing with a Vision-based Mouth Interface. In: Proc. of NIME 2003 (2003)
6. Miller, G.: The mating mind, how sexual choice shaped the evolution of human nature. Anchor (2001)
7. Odowichuk, G., Trail, S., Driessen, P., Nie, W., Page, W.: Sensor fusion: Towards a fully expressive 3D music control interface. In: Pacific Rim Conference on Communications, Computers and Signal Processing (2011)
8. Peterson, G.E., Barney, H.L.: Control methods used in a study of the vowels. J. of the Acoustical Society of America 24, 183 (1952)
9. Riedmiller, M., Braun, H.: Rprop - A Fast Adaptive Learning Algorithm. In: Proc. of the International Symposium on Computer and Information Science VII (1992)
10. Rodet, X.: Synthesis and Processing of the Singing Voice. In: 1st IEEE Workshop on Model Based Processing and Coding of Audio (2002)
11. de Silva, C., Smyth, T., Lyons, M.J.: A novel face-tracking mouth controller and its application to interacting with bioacoustic models. In: Proc. of NIME (2004)
12. Sundberg, J.: The KTH Synthesis of Singing. J. on Advances in Cognitive Psychology 2(2-3) (2006)