

Expressive facial speech synthesis on a robotic platform

Xingyan Li and Bruce MacDonald and Catherine I. Watson

Department of Electrical and Computer Engineering

University of Auckland, New Zealand

Email: {xingyan.li, b.macdonald, c.watson}@auckland.ac.nz

Abstract—This paper presents our expressive facial speech synthesis system Eface, for a social or service robot. Eface aims at enabling a robot to deliver information clearly with empathetic speech and an expressive virtual face. The empathetic speech is built on the Festival speech synthesis system and provides robots the capability to speak with different voices and emotions. Two versions of a virtual face have been implemented to display the robot’s expressions. One with just over 100 polygons has a lower hardware requirement but looks less natural. The other has over 1000 polygons; it looks realistic, but costs more CPU resource and requires better video hardware. The whole system is incorporated into the popular open source robot interface Player, which makes client programs easy to write and debug. Also, it is convenient to use the same system with different robot platforms. We have implemented this system on a physical robot and tested it with a robotic nurse assistant scenario.

I. INTRODUCTION

With the rapid development of robotic technologies, more social and service robots are introduced into people’s everyday life. For example, a robot guide can show people around in a museum and describe the items on display; a robot waitress can take customers’ orders and serve coffee; a robot nurse assistant can measure patients’ blood pressure and assist doctors to deliver heavy objects; a robot companion can read books or newspapers for people. For these applications the robot has a direct communication with people, so the ability to recognize and express human emotions makes the robot more functional and effective. The robot should recognize human expression of emotion, and respond appropriately; that is the robot should be empathetic. For example, a robot nurse assistant should be able to greet people, sound happy to inform patients with good results and express sorrow or encouraging emotions when the test results are not satisfying. This paper focuses on the expressive component, rather than the recognition of emotion.

To communicate with people empathetically, the robot should speak with an empathetic voice and display expressive facial movements. Ideally, communication of emotion will happen through three channels, verbal, vocal nonverbal and facial, each complementing the other two. For the verbal channel, the human–robot dialogue should be designed carefully to express emotion. For example, a robot guide can express welcome by saying “It is very nice to meet you.” and a robot nurse assistant can ease the patients by saying “Your test result is normal, you have nothing to worry about.” For the vocal nonverbal channel, several speech features, such as pitch, duration and intensity can be altered to deliver different

moods. The Tones and Break Indices (ToBI [18]) can also be used for transcribing prosody and works very well in speech emotion expression.

Our goal is to provide expressive facial speech synthesis for robots based on Player [8], a widely used, open source robotic control interface, and using facial expressions, voice control, and ToBI. Player already has a speech synthesis driver based on Festival [19], but the driver provides only plain, machine–like voices and gives the user no ability to vary the voice. We improved the speech driver to offer more expressive, human–like voices. Movement of specific parts of the face can express different emotions. We use an earlier, 113 polygon, Candide-3 [3] model based virtual face and developed a second, 1000+ polygon, Xface [4] based virtual face. Thus, the robot can display diverse facial expressions with different face models.

Player’s robotic device interface acts as a hardware abstraction layer for robotic devices, which makes it is easy for robot developers to write, debug client programs and to use our system with other robot platforms.

We present related work in Section II. The implementation details of our system are presented in Section III, followed by the test results using a physical robot in Section IV.

II. RELATED WORK

Although social and service robots are a relatively recent innovation, much research has already been done in this area. For example, CMU developed the nurse robots Flo [15] and Pearl [13], and the interactive tour-guide robot Minerva [21]. MIT built a social robot Kismet [7] that has simulated human emotion and appearance.

Speech is an important communication medium between robot and humans. Nourbakhsh describes how emotions influence the synthesized speech in a tour guide robot [11]. Although the quality of synthesized speech is significantly poorer than synthesized facial expression and body language [5], it is still possible to generate empathetic speech. Another example of robot expressive speech is Kismets vocalization system [6]. It generates expressive utterances by assembling strings of phonemes with pitch accents, which is what we have done. There are various text to speech (TTS) systems available, such as IBM’s Naxpres, Microsoft’s Speech SDK, AT&T’s Natural Voices and OpenMARY developed by Saarland University [14]. We use the Festival speech synthesis system, a TTS research framework developed by the University of Edinburgh, to generate empathetic

speech for a robot. In Festival, synthesis is based on diphones while the pitch baseline, pitch range, and speech rate for the whole sentence can be controlled. It also provides ToBI [18] for transcribing prosody and control the intonation contour to some extent. We incorporated Festival’s various functions into Player, thus provide feature control and empathetic speech interfaces to the robot.

The use of facial expressions to enhance human-computer interaction has also been previously researched and implemented. Hara’s 3-dimensional robotic face [9] is capable of displaying seven different facial expressions; Thalmann’s virtual face [20] provides face-to-virtualface communication in a virtual world; Bouchra’s synthetic faces [1] are reproduced with appearance parameters extracted from a natural image or video sequence and Sheng’s 3D face [17] can be synthesized from an arbitrary head-and-shoulder image with the complex background. Most virtual face models comply to the MPEG-4 standard [2], a widely known open framework for media encoding and decoding. We have used two versions of virtual face that comply to MPEG-4 standard. The first one was developed by Christopher Bertram and Richard Paul as an undergraduate project, and is based on the Candide-3 model [3] with only 113 polygons. The second one is based on Xface [4] with over 1000 polygons. We incorporated both virtual faces into Player and provide the robot programmer with expressive facial display interfaces to the robot.

In contrast to other empathetic speech or expressive virtual faces, we built our expressive facial speech synthesis system with Player. So the client program is easy to write and debug, and can be easily transferred to other robot platforms. In addition, we focus on providing various control interfaces instead of defining specific expressions. Thus, users can build and use their own speech and face display.

III. DETAILS IMPLEMENTATION

Our system is incorporated into the Player structure as shown in Fig. 1.

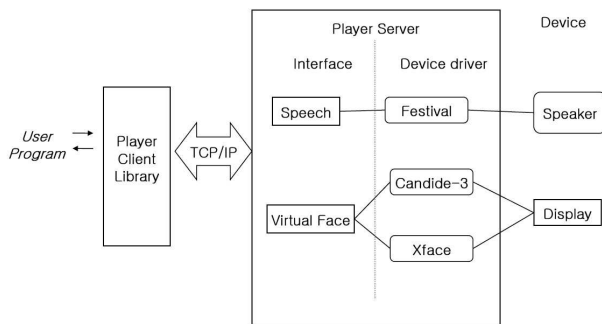


Fig. 1. Our Eface system with the Player structure

A. Expressive speech synthesis

To interact well with users, the robot should have a large vocabulary and full sentence connected speech that is recognizable by any normally hearing person with a good

command of the language. In a healthcare environment, it should be recognizable by people with reduced hearing. For our system, we want the robot to generate clear and expressive speech so that human users can recognize both the meaning and expression. For example, the guide robot should express a happy voice with a welcome speech such as “It is a nice day and I am very happy to meet you.”

The emotion of the speech is not only affected by the words it uses, but also by the way it is said. The vocal non-verbal component is more efficient than the verbal content for communicating information about the speaker’s state or attitude [16]. To realize a more efficient and pleasant human-robot communication, vocal cues should be included in the synthetic speech, especially for robots in social situations, such as guide robots and nursing assistant robots.

Although previous research showed that different speech features might have different emotional effects in different languages, the following specific features of speech may contribute to convey emotional information [14]:

- Pitch and duration play an important role in speech emotion. In particular, the interaction of pitch with loudness and with the grammatical features of the text seems to be critical. In some conditions, pitch and duration are sufficient to distinguish between neutral speech, joy, boredom, anger, sadness, fear and indignation.
- Loudness alone may not be important but the correct synthesis of loudness can help to deliver emotional information.
- Spectral energy distribution and spectral structure can carry much of the affective information.
- Voice quality is also significant in showing the affective information.

There is some disagreement as to how much each of the aspects mentioned above contributes to expressing emotion [14]. In addition, different sentences or different situations require different speech features. Thus, a platform that is able to control all of these parameters is necessary for a general purpose robot.

Festival provides the functionality to change parameters so that the speech emotional features can be altered. For example, the pitch can be altered by the Festival parameters *Current_Toplevel*, *Default_Start_Baseline* and *Default_End_Baseline*; duration can be altered by the Festival parameter *Duration_Stretch*. Our speech synthesis driver in Player provides an interface to control the robot’s speech parameters related to pitch and duration.

Although Festival global parameters can be altered to express simple emotional speech, it turns out that changes to global parameters led to only minor improvements in the output emotion heard and are insufficient to make the speech empathetic. An alternative method to generate more empathetic speech is to use ToBI labels, for which Festival has an interpreter. ToBI is a framework for describing prosody, using annotations to the text input to the TTS. With the ToBI marks, the user can control the intonation contour to some extent. For example with the accent *High* (H) and *Low* (L), sentences can express different emotions:

- Plain accent: “I am very happy to meet you”
- With emotion: “(I ((accent H*)) (am ((accent L*)) (very ((accent H*)) (happy ((accent L*)) (to ()) (meet ((accent H*)) (you ((tone L-H%)))))”
- Plain accent: “Now I am very sad you don’t like me”
- With emotion: “(Now ((accent H*)) (I ((accent L*)) (am ()) (very ((accent H*)) (sad ((accent H*)) (you ((accent L*)) (dont ((accent L*)) (like ((accent H*)) (me ((accent L*)(tone L-L%)))))”

where H(L)* describe the pitch in accented words in a phrase, H(L)– and H(L)% describe the behavior of the pitch contour in the last syllable of an end of phrase, so for L–H% the pitch contour goes down first and then finishes with a short rise. Each phrase has at least 1 pitch accent, and an end of phrase tone. We added the capability to send ToBI-annotated sentences to the rudimentary Festival driver of the Player architecture. Note though, the annotation is not derived automatically via a set of rules or the standard intonation which interprets punctuation marks. The placement of the ToBI labels is done by hand. In addition, other than the typical American and British accent English voices, we also built the New Zealand accent English voice and plan to include Maori language into our system. To switch between different voices, a “set voice” function was added into the original Festival driver and related speech interface.

In summary, our speech synthesis Player interface provides the following functions to the client programs. With these functions, the robot client program can speak empathetically, change speech features dynamically and switch voices easily.

<i>Say:</i>	Speak with plain sentence
<i>SayToBI:</i>	Speak with ToBI marked sentence
<i>GetVoices:</i>	Return all possible voices
<i>RequestVoice:</i>	Return current voice
<i>SetVoice:</i>	Set current voice
<i>GetPitch:</i>	Return current pitch parameter values
<i>SetPitch:</i>	Set current pitch parameter values
<i>GetDuration:</i>	Return current duration parameter values
<i>SetDuration:</i>	Set current duration parameter values
<i>IsPlaying:</i>	Return the current voice playing status

B. Expressive virtual face synthesis

In addition to empathetic speech, expressive display is another important factor for human-robot communication. Facial expressions have the ability to portray a person’s emotional state, temper and mood as well as being able to emphasize and aid in the understanding of spoken text. Our physical robot has a display to show a 3D virtual face which is capable of expressing several emotions as well as rendering the correct lip movements for speech. Two different versions of virtual face have been designed, implemented and incorporated into the Player structure. Version one has only 113 polygons and requires less computation while version two has over 1000 polygons and looks more human-like.

1) *Virtual face version one:* Our first face was developed based on the Candide-3 model [12]. It complies to the MPEG-4 standard [2], which has devised a coding method for graphical models and the transmission of their animation parameters that specifically allows the modelling, manipula-

tion and animation of human facial models. The facial animation coding consists of two standardised sets of parameters: Facial Definition Parameters (FDPs) and Facial Animation Parameters (FAPs). FDPs consist of a set of 3-dimensional feature points that can be used to define the basic geometry of the face, and optionally associate a texture with these points, it makes each individual facial model unique. FAPs represent a set of atomic facial movements relating to key facial features such as the eyebrows, eyes, nose, mouth, ears and tongue. By explicitly separating animation parameters from unique facial models it is possible to apply the same set of FAP movements to different models defined by a unique set of FDPs. This allows the use of one set of FAPs (expression movements) to be applied to different facial models.

The Candide-3 model defines a number of the required FDPs and a Facial Animation Table (FAT), which specifies exactly how the FDPs on the Candide-3 model are manipulated by FAPs. However, it is only a mask and contains only some elements of a real human face. For example it does not include ears, teeth or a tongue. Without using advanced graphic techniques, the face is made up of straight lines and plane surfaces and does not look life-like. However, the CPU usage for computation to manipulate and display the facial model is little and it is still possible to display a large range of expressions using this facial model.

While the Candide-3 model provided the 3-dimensional coordinates of all the required facial definition points and the FAPs needed for facial expression, a texture is applied to the model to realistically represent a human face. Texture mapping is performed by matching a point in 2D on the texture with a facial definition point in 3D on the facial mesh for all the 113 polygons.

Due to the low number of polygons in the face, when rendered in 3 dimensions it appears blocky and sharp edged. A shading technique is applied to make the face appear smoother. The Gouraud Shading algorithm, also known as smooth shading within OpenGL [10], is a method for linearly interpolating shading across a flat polygon. This algorithm calculates the intensity of the light source at each of the 3 vertices of the polygon, then linearly interpolates between the points to calculate a light intensity for every pixel shown on the screen. This allows for smooth shading between the vertices and entirely removes the sharp region borders.

To display different expressions, a set of FAPs that correlate to the movement of the facial definition points are combined to form facial movements that correspond to recognizable expressions. Having expressions in terms of their component FAPs, it is possible to manipulate a facial model by altering the positions of the FDPs. Currently the virtual face can display nine different expressions: *happy, sad, angry, afraid, disgusted, surprised, ecstatic, confused and thoughtful*. For each expression, it associates each related FAP to a magnitude value to specify the amount of FDP movement in the defined direction. Thus, we can have different levels of expression by altering the magnitude value, such as “very happy” and “a little sad.” Other than this static expression, we use linear interpolation based animation to perform a realistic

change from one facial expression to another over time. An initial and a destination facial expression are defined, then a number of in between expressions are interpolated, such that when played one after another they give the appearance of a fluid movement.

In addition to different expressions, there are some other movements to make the face appear more natural. A free moving head is implemented using the OpenGL rotation functions. During the rotation, each time the virtual face is drawn, all vertices and polygons are rotated. The virtual face also allows the robot to express visemes in a similar manner to the facial expressions. The viseme is a generic facial image, particular shaping of the lips and mouth features, that can be used to describe a particular sound and is the visual equivalent of a phoneme or unit of sound in spoken language. Finally, there may be periods of inactivity between changing expressions that would make the face display a static expression for a long period of time, which is unnatural for humans. To overcome this problem, the virtual face will perform some idle movements, such as blinking, eye rotation or head rotation, when the inactive time reaches a limit.

2) *Virtual face version two*: The second, more human-like virtual face is based on Xface, an open source 3D talking head based on the MPEG-4 standard. The Xface toolkit relies on OpenGL and is optimized enough to achieve at least 25 frames per second with a polygon count up to 12000, using modest hardware. This virtual face can show eight expressions: *anger*, *disgust*, *fear*, *rest*, *sad*, *smile(closed)*, *smile(open)* and *surprise*. The 3D virtual face can also perform 16 face movements such as blinking, brow movement, eye squint and looking around. It also able to alter the lips and mouth to express visemes.

With Xface, we can test different face models generated by the FaceGen Modeller from Singular Inversions to determine the most accepted face for a user group. FaceGen can generate realistic 3D faces for any race, gender and adult age group with 36 expressions, phonemes and modifiers. The user can also control the texture color, symmetric shape and add extra parts such as eyeglasses or hats. Combining Facegen’s realistic face images and Xface, we can achieve expressive facial models that look human-like.

Since only Windows version of Xface is available, Alan Yang did some modification to make it run under Linux as part of his Masters project. Both virtual face version one and version two are added into the player structure as drivers. The virtual face player interface provides the following functions to the client programs:

<i>GetExpressions</i> :	return all possible facial expression
<i>RequestExpressions</i> :	return current facial expression
<i>SetExpressions</i> :	set current facial expression
<i>GetVisemes</i> :	return all possible visemes
<i>RequestVisemes</i> :	return current viseme
<i>SetVisemes</i> :	set current viseme
<i>SetSubtitle</i> :	set text subtitle

C. Synchronization between speech and virtual face

An important topic in synthetic speech is the generation of lip movements corresponding to recorded speech. The

classic way to do this is shown as in Fig. 2. First, the TTS software, Festival in our project, converts the input sentence into phonemes. For the speech thread, the phonemes are then used to create a speech waveform. For the virtual face thread, the main thread, the phonemes are mapped into visemes and sent to the virtual face model to realize lip movements. As the movements finish, the next input sentence is processed in the same way.

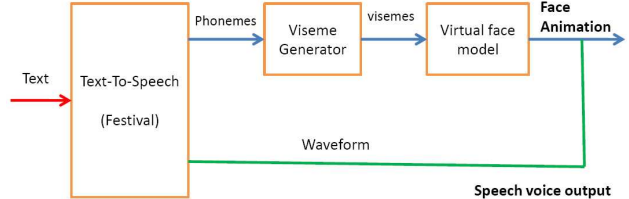


Fig. 2. Speech and virtual face synchronization model

Festival is able to output a text-based list of phonemes for any given block of text spoken, using the utterance command *utt.segs*. The list contains both the specific phonemes spoken and the duration times that each phoneme is spoken for. We then associate the phonemes with their respective viseme output through a comparison process that is implemented using an XML file. The file contains all 42 phonemes in the English language (with another phoneme standing for silence) and relates them to their corresponding viseme values that represent a set of FAPs to move all the required definition points around the mouth area into the correct position. The time to display each viseme is taken directly from the phoneme duration time in Festival so the total time to display the entire viseme animation should be the same as the total time of the audio file being played.

However, in real applications, the phoneme duration time calculated by Festival is not always be exactly the same as the actual audio playing time. This may due to the audio hardware access time, viseme generation time or inaccurate Festival prediction. Therefore, these two threads should be synchronized, for example, the main thread (the virtual face) can wait until the child thread (the audio playback) is finished playing and has joined the main thread. However, the Player structure is a server-client model, which means that the audio playing thread may be on a different computer and has feedback from the Player server only once the server accepts the command. Thus, the speech thread is not able to know if the audio playing is finished or not. We added another function *isplaying()* to the speech interface of Player, which tells whether the audio is still playing, by checking the audio device status. With this new function, the speech and virtual face can be synchronized correctly and the robot shows lip-synchronized speech.

IV. EXPERIMENTAL RESULTS

We have implemented our expressive speech animation synthesis system with Player and tested it with a robot

nursing assistant scenario. In this scenario, a peoplebot robot with an onboard Intel Pentium 1300 MHz processor is used to act the part of a helper for a human nurse. The robot is equipped with a speaker to talk, a display to show its face and a cuff blood pressure monitor connected to the robot vis USB to measure the patient’s blood pressure (shown in Fig. 3). Dialogues are designed for the robot to instruct the patient to use the blood pressure monitor, the result is then recorded and the patient is informed about the results.



Fig. 3. Left: peoplebot; Right: blood pressure monitor

We have tested different empathetic speech for a nursing assistant. It is relatively difficult to express different emotions just by altering the global speech features and can cause side effects. For example, increasing the pitch and decreasing the duration can express a “happy” emotion but it is hard for people to hear the command clearly. By improving, the intonation model the speech becomes more empathetic, and the differences in the emotions more apparent.

For example, the sentence “I am very happy to meet you” would be marked as “(I ((accent H*)) (am ((accent L*)) (very ((accent H*)) (happy ((accent L*)) (to () (meet ((accent H*)) (you ((tone L-H%))).)” Fig. 4 compares the F0 contour of this sentence with and without ToBI annotations.

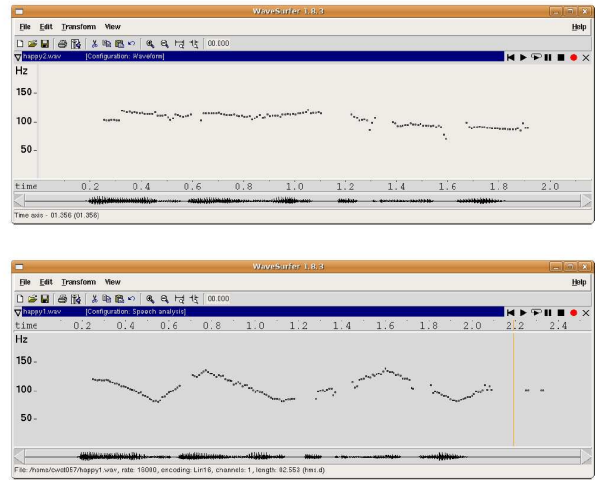


Fig. 4. Top: The F0 contour plot of an utterance without ToBI annotations; bottom: The F0 contour plot of an utterance with ToBI annotations

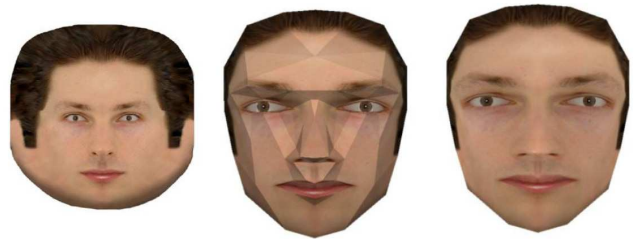


Fig. 5. Left: 2D texture image; Middle: texture mapped virtual face; Right: Gouraud shaded virtual face



Fig. 6. Left: happy face; Middle: surprised face; Right: sad face

With the virtual face version one, a 2D texture image was created by taking a digital photography of a human face and then stretching using Adobe Photoshop to compensate for the curved nature of the face. Texture mapping is then performed to map this 2D texture to the 3D Candide model. Finally, the Gouraud shading algorithm is applied to obtain a smooth face. Fig. 5 shows the texture image, the virtual face after texture mapping and the virtual face after shading.

The virtual face version one can display nine different expressions, three of them are shown in Fig. 6. Due to the limited polygon number, we can see the straight lines and plane surfaces of the face, especially around the edge. The missing ear, tongue and teeth also make the face less human-like.

With more polygons (1000-2000), virtual face version two displays a more realistic human face image. Like version one, it can display different expressions, phonemes and modifiers. We also used Facegen to generate several face models with different race, gender and age. Fig. 7 shows different face models with expressions or phonemes. It provides a more realistic human face than version one. With the default 50 frames per second, the CPU usage of first virtual face is less than 20% while the CPU usage of the second virtual face is around 50% on the Peoplebot.

Both virtual faces have good lip synchronization with the speech. For each of the sentences, there exists an average 0.15 second, maximum 0.5 second, time difference between the audio voice and the visual display. For most users, this

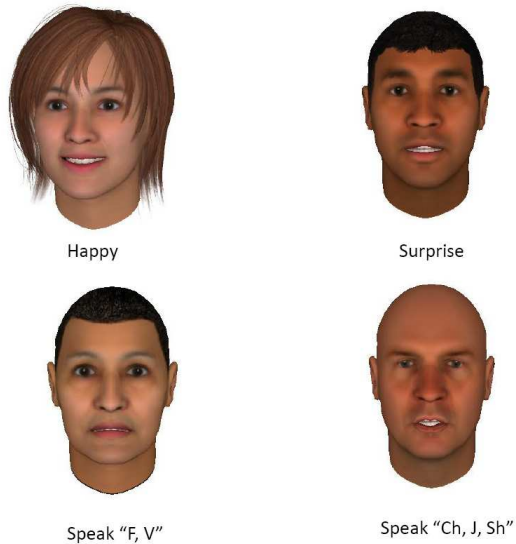


Fig. 7. Virtual face version two with different expressions and visemes.

is acceptable. However, without the new synchronization method, the time difference will be accumulated and the user would have noticed the asynchronization after a few sentences.

A more professional test will be given under the direction of a psychologist in the near future. That test will involve different voice parameters, ToBI marks, face models with different age, sex and race group. Through the test, we will discover what kind of voice/face model is most acceptable for our target user group in a healthcare scenario.

The expressive facial speech synthesis system can be used in many other social and service robot applications, in addition to the nursing assistant application. Since we incorporated it within the Player structure, it is relatively easy and convenient for other applications to use.

V. CONCLUSION

In this paper, we have introduced our expressive facial speech synthesis system which can be used by social and service robots. We focused on the robot's ability to produce expressive facial speech so that clear help and commands will be delivered to users. The empathetic speech, expressive virtual face and the association between them are presented. The advantages and disadvantages of two versions of virtual face are discussed so that users can choose the appropriate one for their own level of computation. Additionally, all of the work has been incorporated into the robotics interface Player. We believe that, as a fundamental function, this work can be applied to many other social or service robot applications.

VI. ACKNOWLEDGMENTS

We would like to thank Liz Broadbent and Tony Kuo for their help in designing and performing test experiments; Richard Paul and Christopher Bertram for creating virtual face version one; Alan Yang for developing the virtual

face two structure; Simon Fong and Nicholas Hart for their visemes and ToBI intonation work; Sigrid Roehling for speech synthesizing and Tony Kuo for porting the system to Peoplebot.

REFERENCES

- [1] Bouchra Abboud, Franck Davoine, and Mo Dang. Expressive face recognition and synthesis. *Computer Vision and Pattern Recognition Workshop*, 5:54, 2003.
- [2] Gabriel A. Abrantes and Fernando Pereira. Mpeg-4 facial animation technology: survey, implementation, and results, 1999.
- [3] Jrgen Ahlberg. Candide-3 - an updated parameterised face. Technical report, Linkoping University, Sweden, 2001.
- [4] Koray Balci. Xface: Mpeg-4 based open source toolkit for 3d facial animation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 399–402, 2004.
- [5] Christoph Bartneck. Interacting with an embodied emotional character. In *Proceedings of the International Conference on Designing pleasurable products and interfaces*, pages 55–60, 2003.
- [6] Cynthia Breazeal. Designing sociable robots. *Robotics and Autonomous Systems*, 2002.
- [7] Cynthia Breazeal and Brian Scassellati. How to build robots that make friends and influence people. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pages 858–863, 1999.
- [8] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, 2003.
- [9] F. Hara and H. Kobayashi. Use of face robot for human-computer communication. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1515–1520, 1995.
- [10] Francis J.Hill. *Computer Graphics Using OpenGL*. Prentice Hall, 2000.
- [11] Ila Nourbakhsh, Judith Bobenage, and Sebastien Grangec. An affective mobile robot educator with a full-time job. *Artificial Intelligence*, 114:95–124, 1999.
- [12] Richard Paul and Christopher Bertram. Virtual face for a robot guide. Technical report, Department of Electrical and Computer Engineering, University of Auckland, New Zealand, 2004.
- [13] Joelle Pineau, Michael Montemerlo, and M. Pollack. Towards robotic assistants in nursing homes: Challenges and results. *Special issue on Socially Interactive Robots, Robotics and Autonomous Systems*, 42:271 – 281, 2003.
- [14] Sigrid Roehling, Catherine Watson, and Bruce MacDonald. Towards expressive speech synthesis in english on a robotic platform. In *Proceedings of the Australasian International Conference on Speech Science and Technology*, pages 130–135, 2006.
- [15] Nicholas Roy, Gregory Baltus, and Dieter Fox. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment*, 2000.
- [16] Klaus Scherer, Robert Ladd, and Kim Silverman. Vocal cues to speaker affect: Testing two models. *Journal of the Acoustical Society of America*, page 13461356, 1984.
- [17] Yun Sheng, Abdul H. Sadka, and Ahmet M. Kondo. Automatic single view-based 3-d face synthesis for unsupervised multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:961–974, 2008.
- [18] Kim Silverman, Mary Beckman, and John Pitrelli. Tobi: a standard for labeling english prosody. In *Proceedings of International Conference on Spoken Language*, pages 867–870, 1992.
- [19] Paul Taylor, Alan Black, and Richard Caley. The architecture of the festival speech synthesis system. In *Proceedings of the ESCA Workshop in Speech Synthesis*, pages 147–151, 1998.
- [20] Magnat Thalman, Kalra P, and Escher M. Face to virtual face. In *Proceedings of IEEE special issue on multimedia*, 1998.
- [21] Sebastian Thrun, Maren Bennewitz, and Wolfram Burgard. Minerva: A second generation mobile tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.