Visualizing Human Pitch in Remote Voice Conversations

Dai Liu
University of Illinois at Urbana-Champaign

DREU 2010

Abstract

People's pitch changes often as they are engaged in a conversation. Most people do not pay attention to their pitch when talking with each other.

In this paper, we present an application that can visualize the human pitches grabbed from a Skype conversation in real time and keep a complete history view of the past conversation as well.

We utilized this visualization application to verify the hypothesis "if two people agree with each other in a conversation, their pitches will tend to approach each other".

1 Introduction

1.1 Background

In recent years, communications over the Internet have become more and more popular. People get used to communicating with each other by making free voice calls via an online conferencing software, such as Skype[7], for it allows people to participate in the same conversation at the same time even though they are not physically in the same place.

When having a remote audio conversation, two partners in the conversation cannot see each other's face such that they miss the opportunity of abstracting information from people's facial expressions and body gestures. They have to solely gain information from the audio space. Researches have been devoted into visualizing different aspects of an audio conversation, such as volume, interaction, dialog content, etc., so that more social cues can be provided through the audio space.

1.2 Motivation and Application

Every visualization has a purpose. Donath, Karahalios and Viegas coined the term "social visualization" for their approach of visualizing online conversations, for the purpose of creating such representation is to "highlight social information and help people make sense of the virtual social world"[2].

People's pitch changes often as they talk to each other, but most people aren't aware of these variances. We wonder what kind of social cues we can extract from the people's pitch in a conversation. There is one hypothesis, stating that, if two people agree with each other in a conversation, their pitches will tend to approach each other.

We would like to explore this hypothesis by creating visualizations of pitch in remote voice conversation through Skype. If our study result confirms this hypothesis, our visualization application can be easily turned into a tool that can facilitate examination of the dynamics in a remote audio conversation.

1.3 Related work

Before discussing our own project, we will briefly review the previous works in visualizing voice conversations.

1.3.1 Talking in Circles

In the Talking in Circles[6] project, each user in an audio conversation is visualized abstractly in a two-dimensional space as a circle of a unique color with the user's name labeled. When a user speaks, a brighter circle will appear inside that user's circle. While the brightness of the inner circle represents the instantaneous energy of that speaker, the size of the inner circle indicates the volume of that speaker. The inner circle will fade out gradually after the user stops talking, which creates a short speaking memory for that speaker. There is a distance threshold from an arbitrary circle C to any other circle, depending on the diameter of circle C, for the user of circle C to hear the audio of the user represented by the other circle. A user can move his circle closer to the other's so that he can hear the other user more clearly. In addition, users can draw on their circle, which creates "a

pictorial and gestural channel to complement the audio".

This system augments the users' audio conversation experience with additional social cues, such as "subgroup conversation", "who is speaking", through visualization.

1.3.2 Visiphone

The Visiphone[3] is "a communication object that opens a graphical as well as an audio portal" between two people in different places. Each user of the Visiphone is assigned a distinct color. When a user speaks, a circle of that color will be projected onto the center of a dome fixed upon a pedestal. The size of a circle is determined by the volume of the user it represents. If both users speak at the same time, the smaller circle will appear atop of the larger one. After every block of time, the circles will move outward for a distance along a spiral, until it disappears from the dome.

With the Visiphone, users are allowed to perceive from the audio visualization the conversational patterns that would otherwise go unnoticed in a regular voice communication interface.

1.3.3 Conversation Clock

The Conversation Clock project[1] visualizes a conversation in a shared display for a group of people at the same location. Each speaker in every single minute of the conversation is visualized by tick marks of different colors along a ring. A tick mark's location along the ring represents the number of seconds into the current minute. A tick mark's height represents the volume of the incoming sound signal. If more than one person speaks at the same time, tick marks will be placed above one another with the smallest on the top and the largest on the bottom. After every single minute, the tick marks along the ring will shrink to be along a smaller ring which is concentric rings with the original ring, and the Conversation Clock will start visualizing the new minute of the conversation along the original ring.

This visualization enables users to identify the "domination, interruption, and activity throughout the conversation"[1], which can facilitate further study of the dynamics in a conversation.

1.3.4 VoiceSpace

In the VoiceSpace project[4], Mathur presented three different visualizations for remote voice conversations.

The first one uses a speech recognition program to capture the content of the conversation and display the English words explicitly, which is useful for archival.

The second one visualizes the volume values by drawing a square of a color assigned to a user every two second in a row from left to right, from bottom to top. The size of the square is determined by the highest volume over each two second period. If both users speak over the same two second time period, then two squares will be drawn in the same position with the smaller square on the top. This visualization gives a history view of the conversation.

In the third visualization, each user is assigned a color. When a user speaks, a circle of that color will be drawn. The coordinate of a circle is determined by the pitch and the volume value of the user. One can easily gain information from this visualization, such as which user in general has a higher volume, or which user in general has a higher pitch.

2 Descriptions

For our research project, we intend to develop an application to help test a social hypothesis that if two people agree with each other in a remote conversation, their pitches will tend to approach each other, and we decide to choose Skype as our remote conferencing tool. Hence, our application should be able to visualize the pitches of both ends in a Skype conversation synchronously while keeping a history of the pitch visualization of the past conversation.

My focus was on designing and implementing the application interface and the visualization graphics using Java and Processing.

After several design iterations, in my final version of visualization graphics, pitches of two users are displayed in parallel in a form similar to an EKG machine. Each user in the conversation is assigned a color - the local user is navy blue and the remote user is orange. When a user speaks, a dot of that user's color will be drawn for each of the incoming sound signals. The x-coordinate of the dot is determined by the number of seconds into the start time of the conversation, and the y-coordinate is determined by the pitch value of that sound signal. In addition, a line segment of that user's color will be drawn to connect the first pitch dot to the second, the second to the third, the third to the forth, so on and so forth. I chose

this design, because in this visualization, it's easy to observe the rise and fall of a user's pitch along time, and it's convenient to compare two users' pitches.

For pitch detection, I used the Fast Fourier Transform algorithm[5] to convert the sound signal to a frequency spectrum. Since graphically the mapping of pitch values in hertz wouldn't look right, I converted the pitch values into cents with a constant starting value of 55 hertz to make them uniform. I tested the accuracy of this pitch detec-

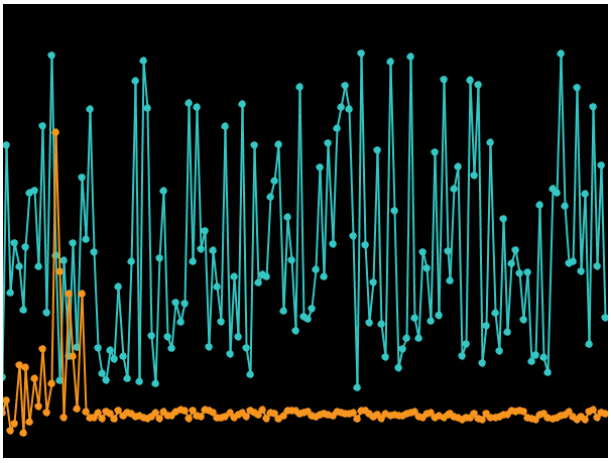

Figure 1: the horizontal part of the orange line visualizes the pitch of a sine wave

tion algorithm by inputting a sine wave created in Mathematica, and checking if the application draws a horizontal line for the pitches it has captured.

2.1 Application Interface

In this section, we will introduce the GUI of our application.

When initially launched, our application has four buttons, "start", "stop", "reset", and "help" on the top. There is a slider labeled "to smooth" below these four buttons. The bottom of the application lies a scrollbar which is grayed out for the moment. There is a horizontal axis labeled time and a vertical axis labeled pitch in the screen.

After connected to the other end in a Skype voice call, the user can click on the "start" button to start visualizations. The user can scroll the slider to average the pitch values we captured such that we can get a smoother line of pitches. The visualization will automatically progress as it goes off the screen. Once the "start" is clicked, two new buttons "agree" and "disagree" will appear in the

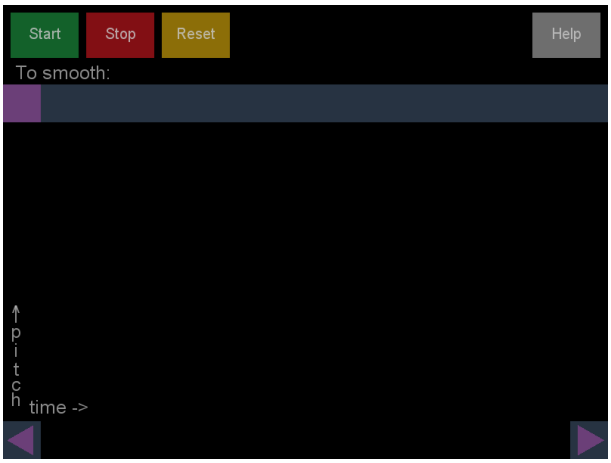

Figure 2: initial interface

screen. Clicking "agree", a green vertical line y = current time will be drawn to represent agreement; clicking "disagree", a red vertical line y = current time will be drawn to represent disagreement

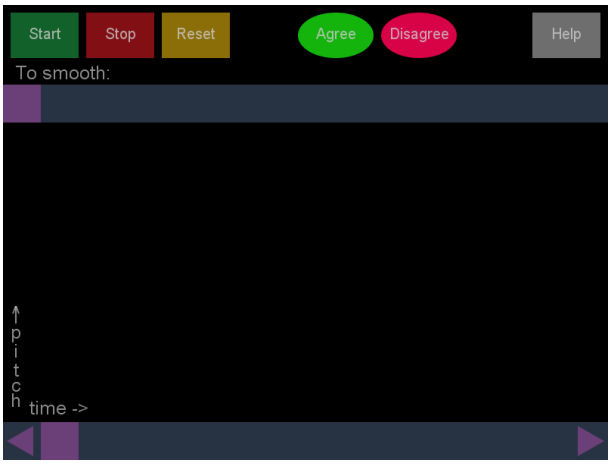

Figure 3: interface after clicking "start"

The user can click on the "stop" button to stop visualizing. Once the visualization is stopped, the bottom scrollbar will no longer be in gray. We can scroll it to get a comprehensive view of the pitch visualizations. In addition, the two pitch lines will be aligned according to its normal, i.e., the average of all the pitch values.

The user can click on the "reset" button to clear all the visualizations in the screen. Thus, the user can start visualizing a new conference call without closing and reopening the application.

When the user clicks on the "help" button, a help menu will appear in the screen with keyboard shortcuts listed. User can choose to save the cur-

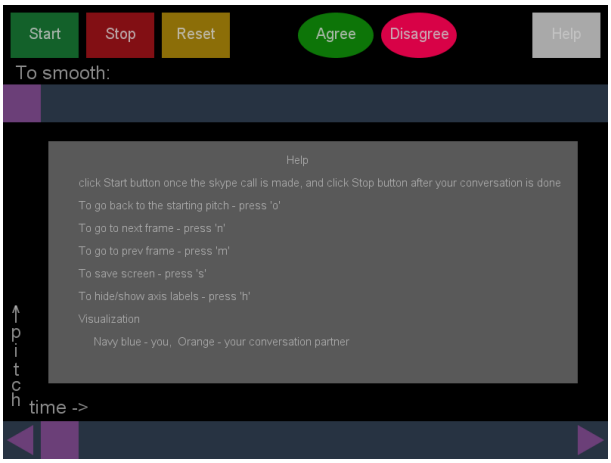

Figure 4: interface after clicking "help"

rent view of visualizations as an image by pressing "s" in the keyboard. The user can choose to log visualizations in text format as a sequence of pitch circle coordinates by pressing "g". The user can load the data log to visualize the past conversations by pressing "l".

2.2 User Study and Reflection

The user study is designed to be done in sessions that each lasts 20 minutes.

We performed two sessions of user studies with two different plans.

For the first session, we recruited three volunteers: two would have a Skype conversation and one would act as the "agreement" judge for them.

Before we started our user studies, the pair of conversation participants was asked to come up with a topic about which they could comfortable talk for 20 minutes.

Two computers were setup in two different rooms. One conversation participant stayed in one room, and the other conversation participant as well as the "agreement" judge stayed in the other room. For the purpose of distinguishing the two participant partners, let's denote the one who stayed in the same room as the "agreement" judge local user, and the other one remote user.

All of the three volunteers were required to use microphones and headsets during the Skype conversation. Since the "agreement" judge needed to listen in the SKYPE conversation and record the point when he felt the two users agreed or disagreed with each other, an audio splitter was used in order to connect two headsets into one computer.

For the second session, we only recruited two volunteers. This time, the local user would act as the "agreement" judge himself. Whenever he felt he agreed or disagreed with the remote user, he would record it himself.

When performing the two user study sessions, we found both user study plans had many flaws.

For the first user study session, it was difficult for the "agreement" judge to determine if the local user and the remote user agree with each other only by listening to the conversations. In a short interview right after the first user study session, the "agreement" judge confessed that he only labeled the "agreement" or "disagreement" when at least one of the users explicitly uttered "I agree" or something equivalent. Besides, the "agreement" judge's judge was largely constrained by his unfamiliarity with the two users.

For the second user study session, since the local could watch the visualization of his own pitches, he might unconsciously adjust his pitches as a response. In addition, our hypothesis emphasizes the agreement with each other, that is, only the moment when the local user agrees with the remote user and the remote user also agrees with the local user should be marked.

We also realized our approach of labeling "agreement" and "disagreement" in the visualization was not suitable for recording "agreement" and "disagreement". The agreement between two users can last a period of time. However, our user can only label their agreement at a discrete point. Our application doesn't appropriately reflect the users' change in agreement along time, which directly adds difficulties for us to analyze the relation between convergence of pitch and users' agreement.

2.3 Results and Analysis

Figure 5 - 14 are screen shots of the visualizations we obtained from the first study session after slightly smoothing. From this set of visualizations, we cannot find a pattern for the relation between pitch convergence and agreement.

One reason is that we cannot tell the trends of a user's pitch by looking at the visualization, for it changes sharply and frequently. Another reason has already been discussed in the previous section: the "agreement" judge didn't associate people's agreement with the corresponding visualization accurately.

Figure 5: screenshot no.1, study session 1

Figure 6: screenshot no.2, study session 1

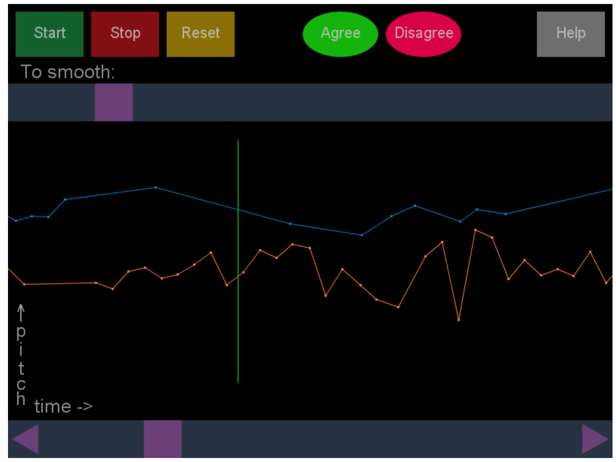

Figure 7: screenshot no.3, study session 1

Figure 8: screenshot no.4, study session 1

3 Conclusions

Since the user studies didn't achieve the desired results, we can hardly draw a conclusion as to whether our hypothesis "two people's pitches converge to each other if they're in agreement in a conversation" holds or not.

3.1 Future Work

We suggest that next step in this project is to develop a new user study plan.

We need a clear definition of two people agreeing with each other. We need to make strict rules for how to determine two people's agreement in a conversation, and, more importantly, we need to come up with methods to identify the part of our visualization that corresponds to the agreement.

After this, a user study on a large scale will be conducted such that enough data is gathered to verify our hypothesis.

Extending this application to be applicable for visualizing a multiple-person conversation is also worth consideration.

4 Acknowledgments

We would like to thank the CRA-W Distributed Research Experiences for Undergraduates (DREU) Program for providing the opportunity, funding and supporting my research experience. We would like to thank Prof Kyratso Karahalios and Prof John Hart at the Department of Computer Science at University of Illinois at Urbana - Champaign for advising me on the project. We would also like to thank the SocialSpace group for the continuous help throughout the project.

Figure 9: screenshot no.5, study session 1

Figure 11: screenshot no.7, study session 1

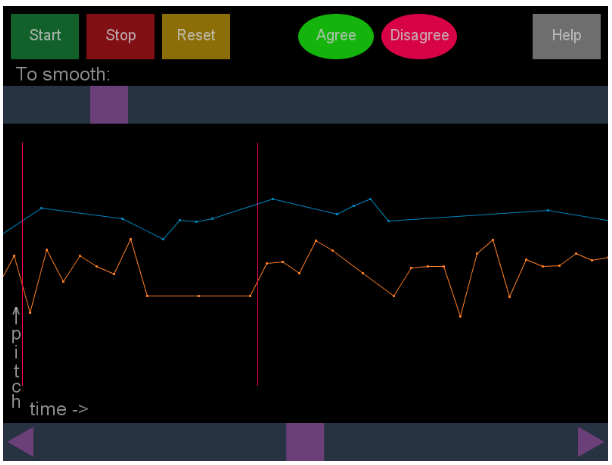

Figure 10: screenshot no.6, study session 1

Figure 12: screenshot no.8, study session 1

References

[1] Tony Bergstrom and Karrie Karahalios. Seeing more: Visualizing audio cues. In *INTERACT'07*, 2007.

[2] Judith Donath, Karrie Karahalios, and Fernanda Viegas. Visualizing conversation. In *HICSS'99*, 1999.

[3] Judith Donath, Karrie Karahalios, and Fernanda Viegas. Visiphone. In *ICAD'00*, 2000.

[4] Pooja Mathur. Visualizing remote voice conversations: Uses from artifacts to archival. Master's thesis, University of Illinois Urbana-Champaign, May, 2009.

[5] A. M. Noll. Cepstrum pitch determination. *Journal of the Acoustical Society of America*, 41(2):23, 1967.

[6] R Rodenstein and Judith Donath. Talking in circles: Designing a spatially-grounded audio-conferencing environment. In *CHI'00*, page 349, 2000.

[7] Skype. http://www.skype.com/.

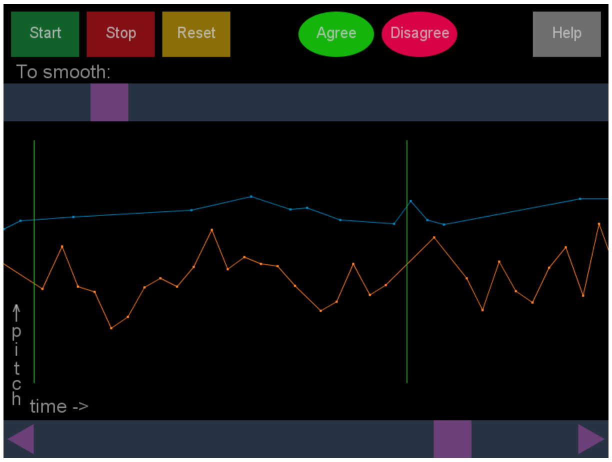

Figure 13: screenshot no.9, study session 1

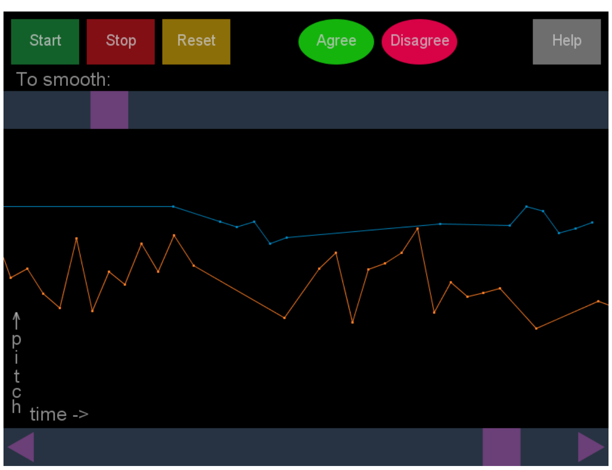

Figure 14: screenshot no.10, study session 1