Probabilistic Interactive Installations

Constance G. Baltera and **Sara B. Smith** and **Judy A. Franklin**
Computer Science Department
Smith College
Northampton, MA 01063

Abstract

We present a description of two small audio/visual immersive installations. The main framework is an interactive structure that enables multiple participants to generate jazz improvisations, loosely speaking. The first uses a Bayesian Network to respond to sung or played pitches with machine pitches, in a kind of constrained harmonic way. The second uses Bayesian Networks and Hidden Markov Models to track human motion, play reactive chords, and to respond to pitches both aurally and visually.

Introduction

Multi-sensor audio/visual interactive installations are being developed by artists and computer scientists, together or independently. Human participants become art-makers by physically interacting with sensors and machines in a dedicated space. Dannenberg and Bernstein (Dannenberg & Bernstein 2006) designed and hosted an installation in which four microphones recorded human-generated sound as well as speaker feedback. Computers filtered the sound in various ways before sending it to speakers, then used the sound to filter projected photographs of art work and scenes. Mitchell and Lillios (Mitchell & Lillios 2006) built an installation that used many sensors to change images projected onto translucent fabric hanging in a space through which humans could move. The purpose was to create an environment that evokes emotional and psychological responses. Palmer (Palmer 2006) reports on an installation between two buildings on a street, at night, in which human motion is detected and sound and images are projected into the space and onto the concrete "floor."

The complex nature of the aesthetics as well as the fusion of different sensor streams, and the interaction with one or more humans, make these installations excellent testbeds for machine learning and artificial intelligence algorithms. An installation with many sensors and one or several participants experiencing it provides a challenge for design and coordination of such algorithms.

Our eventual goal is to devise and build interactive installations that enable public artistic explorations that change over time through experience. Currently we are in laboratory mode, exploring possibilities. We wanted to include uncertainty in the responses of the installation as a way to vary the machines responses. We are interested in having the installation react to human motion and sound patterns. In this pursuit, we investigated the use of Bayesian networks and Hidden Markov Models for these purposes.

Probabilistic graphical models such as these are used in related work in both machine vision of human motion tracking (Ramanan, Forsyth, & Zisserman June 2005; Wren *et al.* 1997) and in analysis of gestures and sound and relationships between them (Jensenius, Godϕy, & Wanderly 2005), such as piano playing gestures. Kolesnik and Wanderley (Kolesnik & Wanderley 2005) trained HMMs to recognize beat-patterns obtained by recording a conductor's motion. They used the trained HMMs to enable a conductor to conduct music generated by computer. The job of these HMMs was to precisely identify the patterns for beat indication. De Poli et al. (DePoli *et al.* 2005) have developed hierarchical multimodal interactive architectures where Hidden Markov Models and Bayesian networks at the mid-level recognize and classify gestures and their expressive intention. Their focus has been mainly analyzing musical gestures and providing an interactive space for a solo dance performance.

Our interest is in a kind of art-making space that is available for participants who are not necessarily arts specialists. Participants might sing a little, move through the space, wave their arms or hands, etc. and will interact with each other.

Our previous work explored the use of recurrent neural networks for interactive jazz playing and generating new jazz "compositions" or solos (Franklin 2004; 2006). A common theme in this work is to use domain knowledge in some way, and to derive pitch and duration representations that foster recurrent network learning 1) of existing melodies, 2) to generate new melodies, and 3) to use human-played improvisations to generate interactive responses. These systems were quite general in that any pitch could be played. This is a double-edged sword in that the systems could easily play many wrong notes. Jazz embraces the playing of many notes in interesting places, but a jazz player must know how to resolve them. In summary, we were researching systems that can produce complex solo improvisations in the jazz idiom over many possible harmonic chord progressions.

We thought it would be interesting to change our focus from this kind of system to installations that produce immersive improvisation using a few chords and a fixed set of pitches that work harmonically over those chords. To this end, we added a pair of interactive sound and graphics installations to a small sound and music laboratory. We based the initial work on interactive sound installations on a loose concept of jazz improvisation.

The first installation started with the notion that a participant would interact with it and generate melodies based on a two-step time history. In the interest of trying a different approach, and backtracking a little in complexity, we decided to use a simple Bayesian network that could produce outputs in a particular scale, rather than any of the 12 chromatic pitches. The Bayesian network takes two input pitches, the current pitch and the previous pitch as evidence. The Bayesian network generates a discrete probability distribution representing the conditional probabilities of each of a predefined set of pitches, given the evidence. This distribution is then used to produce a pitch and is heard as a kind of harmony. The interaction patterns between the participant and the installation are probabilistic in this way. The human participant can make any kind of pitch-like sound and the machine will interpret it as a pitch, and respond with a note of its own from the scale it is constrained to use. Sounds that are not pitch-like, such as scraping, clapping, and banging, are ignored.

The second installation is more complex than the laptop project, involving both sound and graphics, and filling to one extent or another most of the room inside the lab. The framework of the installation is the loose interactive jazz chord progressions and improvisations that are caused by participants' motion through the room, participants' singing or creating pitches, and by physical interactions with two devices: a keyboard, and foot pressure sensors. The second installation incorporates both Bayesian networks and HMMs. We added projected graphics that are controlled through the same graphical algorithms that produce the sound.

We describe the installations in more detail, starting with the section **Interactive Door Installation** with a bit more detail on the laptop-driven door-mounted audio installation. A description of Bayesian networks follows in the Section **Bayesian Networks** tailored for this application (this section may be skipped). The next section is **Immersive Laboratory Installation** and after that a section called **Hidden Markov Models** gives algorithmic details (again, this section may be skipped). We summarize with some of our observations and experiences.

Interactive Door Installation

The physical manifestation of the interactive Bayesian network is a laptop that takes audio input from the microphone on the door of the lab, determines the pitch of the sound, and sends this data through the Bayesian network with the previous recorded pitch. The program then chooses an output pitch according to the resulting probability distribution. Speakers in the hallway just outside the door amplify both the computed harmony pitch and the observed pitch.

Figure 1 shows the front-end of the Bayesian network-driven audio installation. This is the door to the Computer Science Sound and Music Laboratory. It is in the foyer of one of the science buildings on campus. A microphone is taped to the door. The two small speakers are sitting on top of a glassed-in building directory mounted on the wall beside the door. The power and audio signal cables run under the door to a 833Mhz powerbook G4, running OSX 10.4, that is sitting on a shelf just inside the door. The laptop has the music graphical programming environment called Pure Data or just pd (Puckette & others 2006) continually processing the microphone input and choosing the next note to send to the speakers via the Bayesian network.

The conditional probabilities of the (machine) output given the evidence, $P(out|e)$, are computed, using the Python programming language, available as an external to pd. This combination allowed us to access the sound and signal manipulation and graphics capabilities of pd and its extensions and libraries, while still being able to implement the algorithms and computations necessary to create and use the Bayesian networks and HMMs in the more straightforward logic of a non-graphical language.

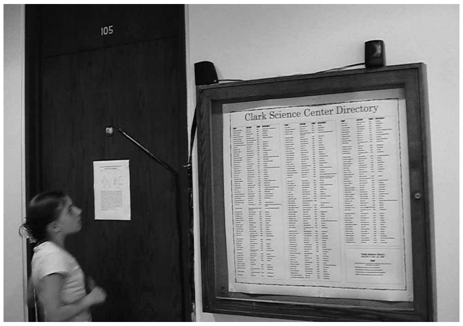

Figure 1: The door audio installation.

The human note is played back through a waveform oscillator in pd along with the machine output. The effect sounds like a harmonization using designated scale tones. Currently the door installation uses notes from the C major scale. As a reminder, the C major scale is C-D-E-F-G-A-B-C. People have been seen or heard singing, whistling, talking, and jingling keys in front of the microphone. The door's "singing" accompanies groups of students chatting as they move between classes as well as the custodian's boombox during foyer cleaning processes. This small installation has been running continuously since June of 2006.

Bayesian Networks

The Bayesian network that generates the machine's output pitches is shown in Figure 2. We include a small amount of theory here but refer the reader to tutorials by Charniak and Storkey (Charniak 1991; Storkey 2006), and to the Jensen text (Jensen 2001).

A Bayesian network is a graph with nodes influencing each other in a probabilistic way. Theoretically we can reason about the joint probability of all possible combinations of events, where an event is one possible outcome for a node. If a node represents an input pitch, an event may be e-flat. One can imagine the space of all possible event combinations becoming large in the number of nodes. In our case, there are just a few nodes, and each node has twelve or fewer event values. The nodes labelled machine output, machine previous output, human input and previous human input all have as events pitches from the same scale. For example, if the possible pitches are from the C major scale, in one octave, there are seven possible pitch events for each of the four nodes just listed, and there are twelve possible intervals. The interval between notes is constrained by the possible notes that can be played by the human. If the human plays a note or sings a pitch that is not in the required scale, the highest possible scale-pitch is assumed. Octaves are ignored so if the human input is 2 octaves above the c at MIDI pitch 60 (middle c), pitch 60 is assumed.

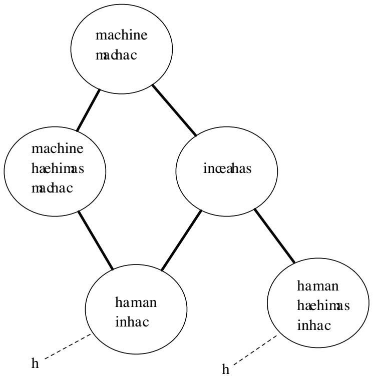

Figure 2: The Bayesian network that models human-computer two-note interactive improvisation, with evidence contraints h and p.

When the Bayesian network is set up, there are prior probability tables set up for all nodes. These values are obtained in general from experts, or from counting the frequency of the outcomes over a large set of experiments, or by other means. All of the individual probabilities must be assigned before any computation occurs. We call the set of prior probabilities of the Machine Output $P(Out)$. We set $P(Out)$ to have uniform probability across all possible notes (for example the seven notes of the C-major pentatonic scale). We improvised on our own instruments and obtained qualitative directions of movement say to and from the tonic from other scale tones in order to set the discrete set of prior probabilities for the other nodes. The prior probabilities for the machine output nodes mimicked those set for the human nodes. Once these probabilities are set, the Bayesian network is ready to use the prior probabilities to generate conditional probabilities, given evidence.

In generative mode, the network is given evidence, as shown in Figure 2. The evidence is one pitch, h, for the human input, and another, p, for the human previous input. The network generates the probability of the (machine) output given the evidence, $P(Out|e)$, and a machine pitch is obtained from that. The human pitch and the machine pitch are sent in succession to the speakers.

To compute $P(Out|e)$ we used the chain rule for Bayesian networks. That is, the joint probability for the whole network is the product of the conditional probabilities of nodes given their parents. Using $U = PrevOut, Out, Interval, HumInput, HumPrevInput$, we have

$$\begin{aligned} P(U) = \quad & P(Out)P(PrevOut|Out)P(Interval|Out)* \\ & P(HumInput|Interval, PrevOut)* \\ & P(HumPrevInput|Interval) \end{aligned} \tag{1}$$

Our evidence $e = \{HumInput = h, HumPrevInput = p\}$ gives us

$$\begin{aligned} P(U|e) = \quad & P(Out)P(PrevOut|Out)P(Interval|Out)* \\ & P(h|Interval, PrevOut)P(p|Interval) \end{aligned} \tag{2}$$

Once we have the fairly succinct expression for $P(U|e)$, the marginalization process is used to obtain the distribution values (Jensen 2001).

Immersive Laboratory Installation

The immersive installation takes place in a small 20 feet by 8 feet lab When beginning this research we decided to make the lab one immersive installation. There are three cameras that track a human motion across the room and back - about all that can be done by a human moving in such a small space. Associated with each of three cameras are three fully connected Hidden Markov Models (HMMs) that have three states each. The three HMMs in a set are given a sequence of nine observations that are retrieved from their camera, using a modification of an example motion detection patch provided in pd. Each observation is the difference between horizontal values of the center of the motion detected by the camera. Observations are taken at half-second intervals. Movement detected by the cameras is identified by the HMMs as being mostly right-oriented, left-oriented, or stationary. This identification triggers the computer to play either a I, IV, or V chord in C-major. Thus, as one moves through the room, stands in the room, or moves back through the room, one is surrounded by a chord progression. Figure 3 shows the installation data flow diagram.

We note that we do not require and do not necessarily want a precise motion identification in the installation. Several participants' motions in the room trigger the chords in different, overlapping combinations. This became part of the improvisation. A Mac G5 runs all HMM programs. A PC uses the motion detection patch to obtain the differences in horizontal motion from the two cameras attached to the PC. These results are sent via the Open Sound Control (OSC) protocol (OSC 2006) to the G5 for use by the HMMs. The lab installation makes use of a C minor pentatonic Bayesian network not only to harmonize with audio input but also to control the colors of a rotating image that is projected onto a lace screen in the corner. The C minor pentatonic is C-Eflat-F-G-Bflat-C. Our interest in the minor pentatonic is that it works well for improvising on chord progressions based on minor chords. It can also be used over certain major chord

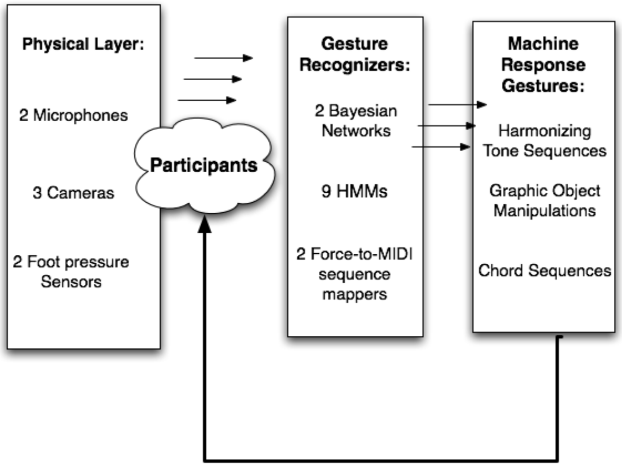

Figure 3: The data flow and algorithms in the current large installation. Gestures are sung or played pitches, motion through the room or arm motions, and foot pressures.

progressions to achieve a "bluesy" sound (Lineberry & Keur 2001). A keyboard at this end of the room enables a human to improvise over the surrounding chords and/or to the Bayesian network on the G5. Small stickers are placed on the piano keys that correspond to C-minor pentatonic scale tones, to enable everyone to improvise. We note that both the C minor pentatonic scale and the C major scale work well over the immersive I-IV-V chord progression. The capitalized Roman numerals refer to chords relative to the tonic (I) which is C here. The IV and V are chords that start on the C-scales's fourth and fifth notes, (f and g). It also works well over the V-IV-I-V (Lineberry & Keur 2001). The motion of more than one person in the room can trigger many combinations of chord progressions and the C minor pentatonic sounds bluesy throughout this motion. Again, this is a bit of loose jazz theory that inspires the choices of music for the interaction. The G5 additionally controls the graphics projected onto the translucent screen. The color of projected/animated objects is determined by the input and output notes of the Bayesian network. The image background is also influenced by these sounds, being textured by a GEM (Graphics Environment for Multimedia) program that maps signal streams to images. GEM is an extension library of pure data (pd), the software used as the installation basis.

In a station of the installation near the door, there are two floor sensors in the room that measure foot pressure. In response to one sensor, one of 12 pitches of the C major scale is emitted. In response to the second sensor, the audio emission is either a two-notes or three-note chord harmony, with C as the tonic (i.e. variants of the I chord).. A human participant may stand on the sensors and add to the surrounding chord sound, or generate scale tones by shifting weight back and forth over the sensors. The pressure values are also used in combination to change the background color of screen-based graphics objects. Since it is by the door, it also causes the door installation to interact with it.

Hidden-Markov Models

HMMs are probabilistic graphs that are often employed to recognize sequences (Rabiner February 1989). The nodes of the HMM correspond to states. Figure 4 shows a fully-connected N-state HMM; a set of state-transition probabilities a_{ij} shapes the transitions of the model:

$$a_{ij} = P(x_t = j | x_{t-1} = i), \;\; 1 \leq i, j \leq N \tag{3}$$

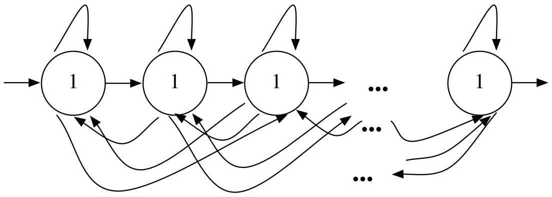

Figure 4: A fully connected HMM.

HMMs represent processes as a set of hidden states that produce observable outputs. Knowledge of the state sequence of a process is determined through observation, where $b_i(o_t)$ is the probability that observation o_t is "produced" by state i. A process is represented by a sequence of events or observations $o_1, o_2, \cdots o_K$.

A single HMM models a human motion in the lab. For each HMM, let λ represent: 1) its state transition probabilities a_{ij}; 2) its observation probabilities b_i; and 3) the set of initial state probabilities π_i. The π_i govern how likely it is that state i is the first state in the state sequence.

Given an observation sequence $O = o_1, o_2, \cdots o_K$, each HMM can generate $P(O|\lambda)$, the probability of the observation sequence, given the values of the HMM's λ. We choose as the recognizer the HMM with the largest $P(O|\lambda)$.

The Baum-Welch iterative algorithm uses sets of example observations to train an HMM in order to find feasible values for λ (Jackson 2006). Training ensues by adjusting the λ values in order to maximize $P(O|\lambda)$ over all possible state sequences. Say one HMM is designated to recognize motion from one side of a room to another in one direction. Examples of sequences of observations from the cameras that fit this behavior are provided to the Baum-Welch algorithm.

A general HMM class was implemented in python, with functions to perform the three basic tasks of an HMM: training of probabilities, recognition of likely observation sequences, and prediction of state sequences. We used three-state, three-observation HMM for simple motion detection in the second installation.

Results

Before the installation occured, the HMMs had to be trained. To train the HMMs, the cameras were enabled and 30 sequences of 9 observations each were collected. However, we tested the HMM recognition properties simply by having three different people move through the room and back, and monitoring the chord progressions that were generated.

Figure 5: A view of the installation in action, with the door behind the viewer.

Under these controlled conditions, the results were consistently correct.

We invited 15 people to a luncheon/installation. Several were from the sciences: geology, neuroscience, engineering, and computer science. A sound artist attended, as well as students and children. Viewers could enter at any time. Figures 5, 6, and 7 are movie stills of the installation.

We did not explain anything beforehand, aiming for discovery. Many viewers wanted to know how everything worked at a technical level. Others were interested in figuring out the cause and effect. A few were listening to the music interaction, especially those who had a background in jazz. Some were happy to make sounds and watch images. Some of the remarks made by participants to each other were:“This is fun, fun!”; “Ho ho!”; “Oh, I see, this one makes the teapot spin.”; “It’s just changing the color.”; “Yah, the foot is the bass.”; “We couldn’t figure out what was doing what.”; “Are all the systems unaware of each other?”; “So in that way they are communicating because whatever is played on the floor sensor is heard by the microphone.”; “What do you think of the experience of it?”; “Wait a minute it shouldn’t be doing that!”; and “OOhhhhh”, in unison.

Figure 6: Another view showing hand motions.

We found the installation to be a nice venue for discussing computing and the arts, and we expect it to be a springboard for many other projects. In particular, we are investigating distributed installations that are not co-located, but in which participants are aware of distant participants. We are also working on collaborations with sound and image artists in orer to gain artistic insight into the installation process. Managing the group dynamics in one or more installations is a rich area for AI research, as is the art-making.

Acknowledgements

We acknowledge Miller Puckette and Thomas Grill for developing Pure Data, flext, and pyext, and the two reviewers who spent time making thoughtful suggestions. This material is based on work supported by the National Science Foundation (Grant No. IIS-0222541) and by Smith College. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and not necessarily those of the National Science Foundation.

References

Charniak, E. 1991. Bayesian networks without tears. *AI Magazine* 50–63.

Dannenberg, R., and Bernstein, B. 2006. Origin, direction, location: An installation. *Proceedings of the Tenth Biennial Symposium on Arts and Technology* 65–68.

DePoli, G.; Avanzini, K.; Rodà, A.; Mion, L.; D’Incà, G.; Trestino, C.; Pirrò, D.; Luciani, A.; and Castagne, N. 2005. Towards a multi-layer architecture for multi-modal rendering of expressive actions. In *Proceedings of 2nd International Conference on Enactive Interfaces.*

Franklin, J. 2004. Computational models for learning pitch and duration using lstm recurrent neural networks. In *Proceedings of the 8th International Conference on Music Perception and Cognition (ICMPC8).*

Franklin, J. 2006. Recurrent neural networks for music computation. *INFORMS Journal on Computing, Special Cluster on Music and Computation.*

Jackson, P. 2006. *HMM tutorials.* `http://www.ee.surrey.ac.uk/Personal/P.Jackson/tutorial/index.html#tip`.

Jensen, F. V. 2001. New York, NY: Springer-Verlag.

Jensenius, A. R.; Godϕy, R. I.; and Wanderly, M. M. 2005. Developing tools for studying musical gestures within the max/msp/jitter environment. In *Proceedings of the 2005 International Computer Music Conference*.

Kolesnik, P., and Wanderley, M. M. 2005. Implementation of the discrete hidden markov model in max/msp environment. *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2005), eds. I. Russell and Z. Markov* 1:68–73.

Lineberry, F., and Keur, D. 2001. Inside the music with stone dragon. In *Web URL*. `http://www.zentao.com/guitar/theory/pentatonic.html`.

Mitchell, B., and Lillios, E. 2006. Merging minds: Collaborative exploration, teaching, and artistic expression. *Proceedings of the Tenth Biennial Symposium on Arts and Technology* 115–120.

2006. Open sound control home page. In *Web URL*. `http://www.cnmat.berkeley.edu/OpenSoundControl/`.

Palmer, S. 2006. Dance and interactive scenography. *Proceedings of the Tenth Biennial Symposium on Arts and Technology* 125–136.

Puckette, M., et al. 2006. Pure data (pd). In *Web URL*. `http://puredata.info/`.

Rabiner, L. R. February 1989. A tutorial in hidden markov models and selected applications in speech. *Proceedings of the IEEE* 77(21):257–286.

Ramanan, D.; Forsyth, D. A.; and Zisserman, A. June 2005. Strike a pose: Tracking people by finding stylized poses. *Computer Vision and Pattern Recognition (CVPR)*.

Storkey, A. 2006. Tutorial: Introduction to belief networks. In *Web URL*. `http://www.anc.ed.ac.uk/~amos/belief.html`.

Wren, C.; Azarbayejani, A.; Darrell, T.; and Pentland, A. 1997. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7):780–785.

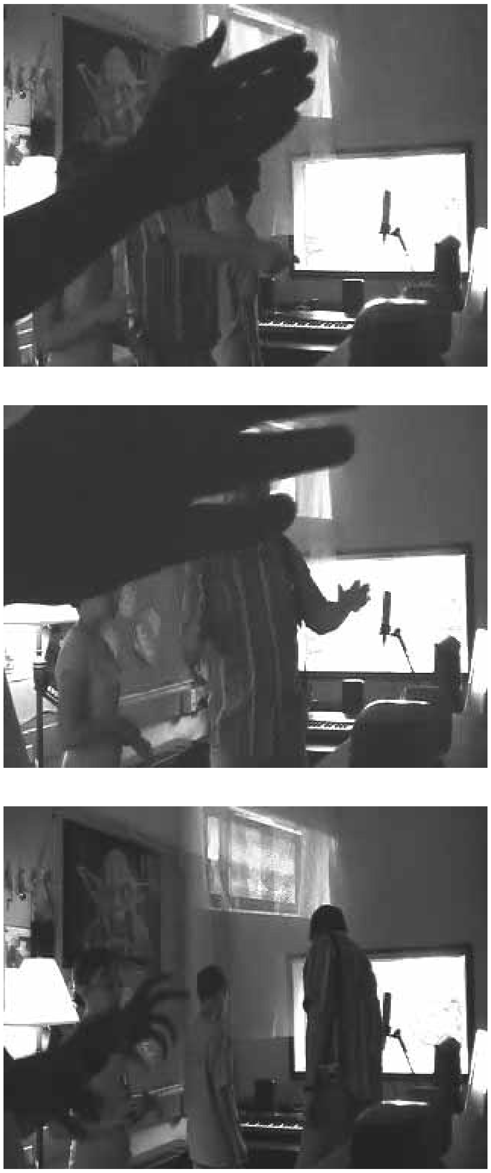

Figure 7: A sequence of views of the installation in action. Many are views with hand motions.