Audio for Kinect: pushing it to the limit

Dr. Ivan Tashev
Principal Architect
Microsoft Research

CREST Symposium on Human-Harmonized Information Technology
Kyoto University, April 2012

Agenda

• Kinect overview
• Acoustical design
• Audio pipeline
• Kinect overall
• Speech enabled user interfaces
• Takeaways
**Start point**

- Initial customer base
  - Mostly young male adults
- Successes and trends
  - Nintendo Wii (2006)
  - Xbox Lips (2008)
- Vision
  - Stand from the couch!
  - No controller required
  - Talk and be understood
Kinect sensors

- **Kinect (2010)**
- **Depth camera**
  - 640x480 depth image, 11 bits resolution
- **RGB camera**
  - 640x480 color, 8 bits/pixel
- **Microphone array**
  - 4 supercardioid microphones, 24 bits ADCs
- **Motorized pivot**
  - Vertical tilt, $\pm 27^\circ$

Depth camera in Kinect

- Scattered infrared light based
- Range set from 0.8 to 4.0 m
- 1 rad FOV, tilt assisted
- Creates 640x480 pixels depth image, 11 bits resolution
- The processing is done in the device
Skeleton tracking

- Executed on the console
- Recognize and track in 3D (x, y, z) all major joints
- Allows posture and body biometrics tracking

Scenarios

- Gesture recognition
- Person tracking and identification
- Gaming (dance, fitness)
Problems to overcome in audio

- Sound coming from the loudspeakers
- Reverberation in the room
- Noise: comparable to voice level at 3.5 meters
- Dynamic range of the sounds to handle
- Building a manufacturable and robust device
- End-to-end system integration and optimization

Acoustical design

From the microscopic changes in the air pressure to electrical signal
Kinect microphones and enclosure

- Desired: acoustically inexistent
- In general introduces complexity
  - Worsens directivity patterns
  - DI loses 2.3 dB
- The enclosure shape can be used to increase the microphones directivity
  - Modeling the sound wave propagation around the enclosure

Kinect microphones and enclosure (2)

- Optimization of the microphones placement and vents
  - Four optimization parameters
  - $Q = w_1 * DI + \ldots$
- Results
  - Final DI improved 1.1 dB
  - Experimentally confirmed in the anechoic chamber
Array geometry

- Every pair of microphones
  - Optimal for one frequency
- N-microphones array
  - Has $N(N-1)/2$ unique pairs
- Given microphone array geometry:
  - We can design optimal in sense of noise suppression array
  - Given design we can analyze and compute DI
  - If we can analyze – we can optimize!

Array geometry optimization

- Optimization parameters:
- Optimization criterion:
  - $Q = w_1 \cdot \text{mean}(\text{DI}) + w_2 \cdot \text{std}(\text{DI})$
- Results
Audio pipeline architecture

We have the acoustical design. Now what?
Acoustic echo reduction systems

- Acoustic echo cancellation
- Acoustic echo suppression
- Mono AEC – part of each speakerphone
More rendering channels?

- Highly correlated channels
  - Ill conditioned matrix
  - Multiple solutions
- The approach on the left
  - Doesn't converge well
  - Has to re-converge after change
- Bell Labs, 1991: “You can’t do stereo echo cancellation” ... with this architecture

Multichannel AEC

- Our multichannel AEC
  - Use calibration pulses, compute mixing filters
  - Lock mixing filters, use one adaptive filter
  - In Kinect implemented for surround sound and four element microphone array
**Microphone arrays: terminology**

- **Beamforming**: making the microphone array to listen to given look-up direction
- **Beamsteering**: electronically change the look-up direction the microphone array listens to
- **Nullforming**: suppressing the sounds coming from given direction
- **Nullsteering**: electronically move the suppression direction
- **Sound source localization**: techniques to detect, localize and track one or multiple sound sources using microphone array

**Beamforming: time invariant**

- **Beamforming**
  - \( Y^{(m)}(k) = W(k)X^{(m)}(k) \)
- **Time invariant beamformer**
  - Isotropic noise assumption
  - Off-line design
  - Set of pre-computed beams
- **Design criterion**: minimize the noise, keep desired
  - \( W_c(k) = \arg\min_{W_c(k)} W_c(k)N(k)W_c^H(k) \)
  - subject to \( W_c(k)D_c(k) = 1 \)
Beamforming: adaptive

- Adaptive beamformer
  - On the fly computation of the weights
  - Higher CPU requirements
  - Does nullsteering
- MVDR beamformer
  - \( W_{\text{MVDR}}(k) = \frac{\Phi_{kk}(k)}{\Phi_{kk}(k) \Phi_{kk}^H(k) \Phi_{kk}(k) \Phi_{kk}^H(k)} \)
- Demos

End-to-end optimization

- Mean Opinion Score (MOS) and Perceptual Evaluation of Sound Quality (PESQ)
- 75 parameters for optimization
- Optimization criterion:
  - \( Q = \text{PESQ} + 0.05 \times \text{ERLE} + 0.001 \times \text{SNR} - 0.001 \times \text{LSD} - 0.01 \times \text{MSE} \)
- Optimization algorithm
  - Gaussian minimization
- Data corpus with various distance, levels, reverberation
- Parallelized processing on computing cluster
End-to-end optimization: results

Multi-volume recognition results

Improvements in WER, %

0 10 20 30 40 50 60 70 80 90 100

Loudspeaker volume, %

- Do nothing
- Before optim
- After optim
- After training

Speech NUI good up to here

Supported levels up to here

Results: demo

Audio pipeline output

Surround sound output

Direction, confidence

To all blocks

Audio for Kinect: pushing it to the limits
Kinect overall

- All together: amazing new way to play
  - 10 millions sold for four months
  - The best selling consumer electronics product ever
- Kinect is an HMI sensor!
  - Goes way beyond gaming
  - Voice modality preferred for media selection by 54%
- Other applications of such technologies
  - Speech and gesture modalities to the HMI
  - People recognition and tracking
Kinect for Windows

- Kinect for Windows Development Kit (KDK)
  - Beta version released from MSR June 2011
  - Commercial version 1.0 released February 1st 2012
- Contains
  - Drivers for the cameras and microphones
  - Depth image processor and skeletal tracking
  - Audio pipeline with mono AEC
  - Speech recognizer with acoustic models
- Applicable in education and industry

Speech enabled UIs
We have the sound. How to use it?
Automatic Speech Recognition

- ASR is a statistical pattern matching problem
  - Find most likely word sequence to explain input

\[ W_{\text{hyp}} = \arg \max_W P(W|S) = \arg \max_W P(W)P(S|W) \]

- Customizing your system you focus mostly on \( P(W) \)
  - Context Free Grammar (CFG)
  - Statistical Language Model (SLM)

Classic CFG based system

Speech \rightarrow \text{Recognizer} \rightarrow \text{Map to query to the database}

Phrases \rightarrow CFG \rightarrow P(W)

\[ P(S|W) \]
ASR with fixed grammars (CFG)

- Defines items user can say at a given time
  - Easy to author (XML format)
  - Compact

“Please say flight information, new reservation, existing reservation, or customer service”

Is this a problem for music selection?

- MSR Intern data collection
  - > 50% of the queries contain multiple fields from meta data
  - “Play Yesterday by the Beatles”
  - Users do not know or use the exact names of the titles or names
- Which one you prefer:
  - “Play track Serenade No. 13 for strings in G major K.525”
  - “Play Little Night Music by Beethoven” (even if it is composed by Mozart 😊)
SLM and parser based processing

Speech → Recognizer → SLU Parser → Structured query to the database

Training data → SLM → HMM → Sampling & Annotation

Demo: music selection and SMS reply

Natural Language Input

- Non-hierarchical menu, single free phrase query
- Support for non-strict queries with fuzzy matching names:
  - More than 60% of the queries have two fields (i.e. “Play Yesterday from Beatles” – title and artist)
  - Most of the queries are with non-complete names (i.e. “Play the Myth of Fingerprints” instead “Play the All Around the World or the Myth of Fingerprints”)
- Minimizing the dialog turns
  - Confirmation only when necessary, or low confident
  - Full query support, but with clarifications if necessary

Multimodal User Interface and NUI

- Speech only = phone call 😊
- Multimodal Interface: voice, touch, gesture, GUI, buttons
  - The goal is increasing the usability
- Voice is good for long lists, GUI/gesture – for short:
  - The combination requires less time and attention: query by voice, confirm from the short list by buttons.
  - Support “voice only” and “GUI/gesture only” modes
  - Gradually move the boundary between them based on the situation
- With the right proportion of the modalities and proper design we can achieve so-called Natural User Interface that doesn’t require training to operate the system
Conclusions

- Several breakthrough achievements in Kinect audio
  - The only product with surround sound echo cancellation
  - Hands free sound capture system for speech recognition
  - First open microphone speech recognition system
- A chain of optimal blocks is suboptimal!
  - If one of the blocks underperforms – the entire pipeline does!
  - End-to-end optimization played critical role
- Kinect is a HMI sensor!
Future work

- Fusion of the sensors:
  - Vision, depth, audio for better results
- More advanced audio processing techniques
  - Sound source separation
  - Speaker ID
- Demo: sound source separation
  - Two persons, 2.4 meters, 26° separation
- And this is just the beginning of the journey ...

Shameless plug


Contains algorithms, a lot of figures, and sample Matlab code
Finally ...

Thank you for your attention!

Questions?

Contact info: ivantash@microsoft.com