Microphone Array project in MSR: approach and results

Ivan Tashev
Microsoft Research
June 2004
Agenda

- Microphone Array project
- Beamformer design algorithm
- Implementation and hardware designs
- Demo
Motivation

- PCs today have pretty bad “ears”; audio captured or recorded from PCs sounds terrible (especially with laptops) – unless a good headset is used.
- Sound will play more and more important role in human-computer interaction, especially in devices without keyboard (tablets, handhelds)
- Increases using computers in collaboration and communication
- Users don’t like headsets or other tethered microphones, especially in a video call.
- Existing wireless solutions do not provide enough good sound quality, you have to wear them
Microphone array project: goals

- **Far goal:** sound capturing quality for untethered user the same as with close-up microphone
- **Near goal:** Create technology for OS support and devices so cheap to become commodity on the market
- Beamforming is ability to make the microphone array to listen to given location, suppressing the signals coming from other locations
Target scenarios

● Real-time communications
  – Providing good sound capturing for Windows Messenger, MSN Messenger, other applications built on top of the RTC stack
  – New applications for VoIP and enhanced telephony

● Collaboration and groupware
  – High quality sound from meeting rooms for recording and broadcasting purposes (OneNote)
  – Voice messaging

● Speech recognition
  – Voice commands for Tablet PCs and handhelds
  – Voice control and dictation for PCs and laptops
Problems

- “Wear nothing” approach requires using separate microphones: connected or integrated
- These microphones deliver poor sound capturing quality:
  - Too much ambient and electronic noises
  - Reverberation and reflections – poor user experience and bad speech recognition results
- Noise suppression and de-reverberation are difficult with a single microphone channel
The solution

- Using microphone arrays for capturing the sound
  - A set of close positioned microphones
  - Synchronous capturing of the signals
- Microphone Array acts as an acoustic antenna
  - This is called spatial filtering or beamforming
  - Listens only to the direction of the speaker
  - Reduces the noises from other directions
  - Reduces the reverberation
Beamforming: known approaches

- **Fixed beam formation**
  - Delay and sum – most intuitive, irregular beam shape
  - Parametric solutions: very complex
  - Fast real-time execution

- **Adaptive beamformers**
  - Generalized side lobe canceller
  - Vary with the target criteria (MVDR, etc.)
  - Slow adaptation, CPU time intensive
Beamforming: known approaches

- **Fixed beam formation**
  - Delay and sum – most intuitive, irregular beam shape
  - Parametric solutions:
  - Fast real-time execution

- **Adaptive beamformers**
  - Generalized side lobe canceller
  - Vary with the target criteria (MVDR, etc.)
  - Slow adaptation, CPU time intensive
Beamforming: known approaches

- Fixed beam formation
  - Delay and sum – most intuitive, irregular beam shape
  - Parametric solutions: very complex
  - Fast real-time execution

- Adaptive beamformers
  - Generalized side lobe canceller
  - Vary with the target criteria (MVDR, etc.)
  - Slow adaptation, CPU time intensive
Beamformer: canonical form

- Canonical form of the beamformer:

\[ Y(f) = \sum_{i=0}^{M-1} W(f,i) X_i(f) \]

- For each weight matrix we have corresponding shape of the beam \( B(\phi, \theta, f) \) - the array gain as function of direction

- The goal is to find weight matrix to satisfy certain criteria
Beamformer: Array parameters

- Noise = ambient + non-correlated + correlated (jammers and reverberation)
- Ambient noise gain
- Non-correlated noise:
- Correlated (from given direction):
- The total noise gain is the combination of the first two
Weights calculation

- Weights calculation as optimization process
- Minimization criterion: the total noise gain
- Multidimensional optimization
  - Slow, especially in real time (adaptive beamformers)
  - Can’t follow the changes
- Multimodal 2M dimensional hypersurface – local minima
- In all cases the starting point is critical
Weights calculation (2)

Our approach:

- Deterministic beam formation
- Use as much prior info as possible
- Do your homework: calculate the weights in advance
- Calculate set of beams to cover the work volume
- Fast real-time engine: switches the beams on the fly
Beamformer: Prior Info

- **Prerequisites:**
  - Microphone array geometry – microphones coordinates and orientation
  - Directivity response of the microphones \( U_m(f,c) \)
  - Hardware noise model \( N_i(f) \)
  - Ambient noise model \( N_A(f) \)
Beamformer: Prior Info

- Prerequisites:
  - Microphone array geometry – microphones coordinates and orientation
  - Directivity response of
  - Hardware noise model
  - Ambient noise model
Beamformer: Prior Info

- Prerequisites:
  - Microphone array geometry – microphones coordinates and orientation
  - Directivity response of the microphones $U_m(f,c)$
  - Hardware noise model $N_i(f)$
  - Ambient noise model $N_A(f)$
Beamformer: Prior Info

- Prerequisites:
  - Microphone array geometry – microphones coordinates and orientation
  - Directivity response
  - Hardware noise model
  - Ambient noise model
Beamformer: Prior Info

- Prerequisites:
  - Microphone array geometry – microphones coordinates and orientation
  - Directivity response of the microphones $U_m(f, c)$
  - Hardware noise model $N_h(f)$
  - Ambient noise model $N_A(f)$
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right)\cos\left(\frac{\pi(\varphi_T - \varphi)}{\delta}\right)\cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} W_{1xM} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho)}{\rho_k}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} \]
- Do the design

![Graph showing beams at 1250 Hz]
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \phi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \cos\left(\frac{\pi(\phi_T - \phi)}{\delta}\right) \cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} W_{1xM} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \]
- Define the weight
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \cos\left(\frac{\pi(\varphi_T - \varphi)}{\delta}\right) \cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1\times L} = V_{1\times L} D_{M\times L} M_{M\times L} W_{1\times M} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target,
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE,
  \[ T_{1xL} = V_{1xL}D_{MxL}M_M \]
- Do the design in 3D.
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \cos\left(\frac{\pi(\varphi_T - \varphi)}{\delta}\right) \cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} W_{1xM} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos \left( \frac{\pi(\rho_T - \rho)}{k \delta} \right) \cos \left( \frac{\pi(\varphi_T - \varphi)}{\delta} \right) \cos \left( \frac{\pi(\theta_T - \theta)}{\delta} \right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} W \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos \left( \frac{\pi (\rho_T - \rho)}{k \delta} \right) \cos \left( \frac{\pi (\varphi_T - \varphi)}{\delta} \right) \cos \left( \frac{\pi (\theta_T - \theta)}{\delta} \right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1 \times L} = V_{1 \times L} D_{M \times L} M_{M \times L} W_{1 \times M} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target function:
  \[ T(\rho, \phi, \theta, \delta) = \cos \left( \frac{\pi (\rho_T - \rho)}{k\delta} \right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE:
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{Mx} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right)\cos\left(\frac{\pi(\varphi_T - \varphi)}{\delta}\right)\cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1\times L} = V_{1\times L} D_{M\times L} M_{M\times L} W_{1\times M} \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \phi, \theta) = \cos \left( \frac{\pi (\rho_T - \rho)}{k\delta} \right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL}D_{MxL}M_N \]
- Do the design in 3D
Pattern synthesis

- Design in the beamspace
  - Define the target beam shape:
    \[
    t \text{ beam shape:} = \cos \left( \pi \rho \right) \cos \left( \pi \theta \right) \cos \left( \pi \phi \right)
    \]
  - Do the design in 3D

\[
I_{1xL} = V_{1xL} D_{MxL} W_{MxL} W_{1xL}
\]
Pattern synthesis

- Design in the beamspace
- Define the target beam shape:
  \[ T(\rho, \varphi, \theta, \delta) = \cos\left(\frac{\pi(\rho_T - \rho)}{k\delta}\right) \cos\left(\frac{\pi(\varphi_T - \varphi)}{\delta}\right) \cos\left(\frac{\pi(\theta_T - \theta)}{\delta}\right) \]
- Define the weight function
- Combine the microphone directivity patterns using weighted MMSE
  \[ T_{1xL} = V_{1xL} D_{MxL} M_{MxL} W_{1xM} \]
- Do the design in 3D
Dimensions reduction

- Dimensions reduction: from 2M to 1
- Two controversial processes:
  - Narrow beam: better ambient noise reduction
  - Wide beam: better internal noise reduction
- One dimensional search: beam width
- Cover the whole frequency band
- Calculate set of beams
On next charts:

- **Z-axis**: noise gain in dB
- **X-axis**: frequency, logarithmic, 1-100Hz, 2-200 Hz, 3-400Hz, …7-6400Hz
- **Y-axis**: beam width, linear, 0 – 180°, every 5°, 33-15°.
Ambient noise gain
Non-correlated noise gain
Total noise gain

![3D graph showing noise gain, frequency, and beam width]
Dimensions reduction

- Dimensions reduction: from 2M to 1
- Two controversial processes:
  - Narrow beam: better ambient noise reduction
  - Wide beam: better internal noise reduction
- One dimensional search: beam width
- Cover the whole frequency band
- Calculate set of beams
Implementation: overall

Offline – Design the weights

Real time – just use pre-calculated weights

MASynthesis.exe

MicArr.INI

Weights.dat

AEC

MABeamformer

Noise Suppression
Implementation: Real-time engine

- **SSL**
  - Beam selection

- **Gain calibration**
  - Gains correction

- **Beamformer**
  - Geometry
  - Weights
  - N-channels input stream
  - Mono output stream
Hardware designs

- USB MicArray Prototypes
  - 4-mic desktop
  - 8-mic conference tabletop
  - Bus-powered (no power grid)
  - Compatible with USB audio (no device drivers to install)

- Integrated in laptops/monitors

Microphones of the L-shaped microphone array for Tablet PC
Results: noise suppression

- Microphone Array noise suppression
  - Provides itself 14-18 dB ambient noise suppression
  - Helps the noise suppressor to do better job
  - More at http://micarray

- One of the best technologies on the market

<table>
<thead>
<tr>
<th>Device</th>
<th>Noise</th>
<th>Signal</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omni-directional Microphone</td>
<td>-45.53</td>
<td>-40.64</td>
<td>4.89</td>
</tr>
<tr>
<td>Unidirectional Microphone</td>
<td>-44.51</td>
<td>-33.91</td>
<td>10.6</td>
</tr>
<tr>
<td>Close-Up Microphone</td>
<td>-64.46</td>
<td>-30.04</td>
<td>34.42</td>
</tr>
<tr>
<td>Andrea DA 400 2.0, 4 el. MA, $135</td>
<td>-51.72</td>
<td>-26.19</td>
<td>25.53</td>
</tr>
<tr>
<td>Acoustic Magic, 8 element MA, $250</td>
<td>-62.39</td>
<td>-32.6</td>
<td>28.79</td>
</tr>
<tr>
<td><strong>MSR 4 elements + WinXP NS</strong></td>
<td><strong>-61.68</strong></td>
<td><strong>-33.86</strong></td>
<td><strong>27.82</strong></td>
</tr>
<tr>
<td><strong>MSR 4 elements + New NS</strong></td>
<td><strong>-64.41</strong></td>
<td><strong>-32.14</strong></td>
<td><strong>33.27</strong></td>
</tr>
</tbody>
</table>
Results: speech recognition

- Microphone Arrays for speech recognition
  - Linear processing, speech recognition friendly
  - Reduces ambient noises
  - Partial de-reverberation

- Results

<table>
<thead>
<tr>
<th>Device</th>
<th>Error rate, %</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC Mic</td>
<td>20.391</td>
<td>3:25</td>
</tr>
<tr>
<td>VoiceTracker</td>
<td>17.9</td>
<td>3:17</td>
</tr>
<tr>
<td>MSR MicArray</td>
<td>14.22</td>
<td>4:03</td>
</tr>
<tr>
<td>MSR MicArray+NS</td>
<td>13.683</td>
<td>3:34</td>
</tr>
<tr>
<td>Close-up</td>
<td>6.171</td>
<td>2:35</td>
</tr>
</tbody>
</table>

4 element array, Yakima SAPI 5.2
374 utterances, 7 speakers
(4 male, 3 female), age 25-53
Results: conclusions

- Ambient noise suppression
  - The current technology provides good noise suppression under the quality requirements constrains
  - Telecommunication scenario has good quality sound
  - Meetings recording for listening purposes – OK.

- Speech recognition results
  - Need improvement
  - Reverberation as major reason
  - Important for recorded meetings search technology
Microphone Array - Example

- Person speaking at 3 ft from microphones

<table>
<thead>
<tr>
<th>Microphone Type</th>
<th>SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical $10$ PC microphone</td>
<td>$10.3$</td>
</tr>
<tr>
<td>PC mic + WinXP noise reduction</td>
<td>$18.4$</td>
</tr>
<tr>
<td>Competitor (HW DSP)</td>
<td>$34.4$</td>
</tr>
<tr>
<td>MSR USB desktop array</td>
<td>$42.5$</td>
</tr>
</tbody>
</table>

Example waveforms showing the signal-to-noise ratio (SNR) for different microphone configurations.
Microphone array - demo

● First demo:
  – Records in parallel the output of the microphone array and a regular PC microphone.
  – After this merges both WAV files to one file …
  – … and plays it with CoolEdit.

● Second demo: ClearMessage application
Take outs

Most of our projects are optimization in one way or another:

- Try carefully to define the optimization criterion
- Reduce the number of dimensions as much as possible
- Choose the method, especially if there are too many papers and no definite answer
Finally

Questions?

Contact: ivantash@microsoft.com
See: http://research.microsoft.com/users/ivantash/