Visually-Guided Beamforming For A Circular Microphone Array

by

Jianglin Fu

B.Sc., Simon Fraser University, 2018

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science in the School of Engineering Science Faculty of Applied Sciences

© Jianglin Fu 2020

SIMON FRASER UNIVERSITY
Spring 2020

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.
Approval

Name: Jianglin Fu

Degree: Master of Applied Science

Title: Visually-Guided Beamforming For A Circular Microphone Array

Examining Committee: Chair: Parvaneh Saeedi
Associate Professor

Ivan Bajić
Senior Supervisor
Professor

Rodney G. Vaughan
Supervisor
Professor

Bonnie L. Gray
Internal Examiner
Professor
School of Engineering Science

Date Defended: April 6, 2020
Abstract

Beamforming is a technique which can adaptively steer the pattern of a microphone array towards or away from a target direction. Three conventional beamforming techniques are reviewed and compared with a beamformer proposed here, called MVDR-2C.

Most acoustic beamformers selectively locate a single desired sound source, such as a speaker, and the beamforming performance drops significantly when two or more speakers are active. In order to deploy beamforming in a room, a circular microphone array is supplemented by a 360° camera comprising two fisheye lenses. The camera allows face detection to provide the speaker directions to the beamformer. In order to develop face/object detectors that operate directly on fisheye images, three annotated fisheye datasets are generated and used to re-train an existing face detector. Finally, several beamformers are evaluated and compared, demonstrating the clear performance advantage of the proposed one.

Keywords: Fisheye image, face detection, object detection, deep learning, beamforming, audio-visual speech processing, visually-guided beamforming
Dedication

I dedicate this thesis to my parents and friends. I couldn’t have done this without your support.
Acknowlegements

First and foremost, I would like to express my deepest gratitude to my senior supervisor Dr. Ivan Bajić, who always provides valuable advice, great support and extraordinary patience during my study at SFU. The wealth of knowledge and feedback he provided me has strengthened my work. I would not have been able to finish this work without him. I would like to thank my supervisor, Dr. Rodney Vaughan with his insightful guidance, encouragement, support and invaluable feedback on my work.

I am extremely grateful to my colleagues at the Multimedia Laboratory at SFU: Saeed Ranjbar Alvar, Lior Bragilevsky, Elham Ideli, Hyomin Choi, Alon Harell, Herath Gedara Chinthaka Pathum Dinesh and Stephen Makonin for their inspiration and encouragement to both my research and life. I also want to thank Alejandro Rodriguez Silva, Chris Hynes and Joe Steyn for their generous comments and advice.

Finally, I would like to acknowledge my beloved parents and friends for providing an endless source of support and encouragement.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval</td>
<td>ii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Acronyms</td>
<td>xiii</td>
</tr>
<tr>
<td>Nomenclatures</td>
<td>xv</td>
</tr>
<tr>
<td>1  Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Face/Object Detection Using 360° Cameras</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Beamforming</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Thesis Contribution and Outline</td>
<td>5</td>
</tr>
<tr>
<td>2  Face/Object Detection in Fisheye Images</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>7</td>
</tr>
<tr>
<td>2.2 FDDDB-360</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Image Extrapolation</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Fisheye-like Distortion</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 Annotation</td>
<td>10</td>
</tr>
<tr>
<td>2.2.4 Further Details of FDDDB-360</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Wider-360</td>
<td>12</td>
</tr>
<tr>
<td>2.4 VOC-360</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Transfer Learning for Face Detection in Fisheye Images</td>
<td>15</td>
</tr>
</tbody>
</table>
### 3 Circular Array Beamforming

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Literature Review</td>
<td>25</td>
</tr>
<tr>
<td>3.2.1 Delay-and-Sum (DAS) Beamformer</td>
<td>25</td>
</tr>
<tr>
<td>3.2.2 Generalized Sidelobe Canceller (GSC)</td>
<td>25</td>
</tr>
<tr>
<td>3.2.3 Minimum Variance Distortionless Response (MVDR) Beamformer</td>
<td>28</td>
</tr>
<tr>
<td>3.3 MVDR with Two Constraints (MVDR-2C)</td>
<td>30</td>
</tr>
<tr>
<td>3.4 Simulation of a Circular Array Beam Pattern</td>
<td>32</td>
</tr>
<tr>
<td>3.5 Beamformer Evaluation</td>
<td>35</td>
</tr>
<tr>
<td>3.5.1 Experiment Setup</td>
<td>37</td>
</tr>
<tr>
<td>3.5.2 Evaluation Metrics</td>
<td>38</td>
</tr>
<tr>
<td>3.5.3 Results and Analysis</td>
<td>39</td>
</tr>
<tr>
<td>3.6 Conclusions</td>
<td>44</td>
</tr>
</tbody>
</table>

### 4 Visually-Guided Beamforming

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>46</td>
</tr>
<tr>
<td>4.2 Equipment Specification</td>
<td>47</td>
</tr>
<tr>
<td>4.2.1 MATRIX Creator</td>
<td>47</td>
</tr>
<tr>
<td>4.2.2 Ricoh Theta V</td>
<td>48</td>
</tr>
<tr>
<td>4.3 System Overview</td>
<td>49</td>
</tr>
<tr>
<td>4.4 Sensitivity to the Accuracy of Face Detection</td>
<td>54</td>
</tr>
<tr>
<td>4.4.1 Experimental Setup</td>
<td>55</td>
</tr>
<tr>
<td>4.4.2 Results Analysis</td>
<td>55</td>
</tr>
<tr>
<td>4.5 Conclusions</td>
<td>60</td>
</tr>
</tbody>
</table>

### 5 Conclusion and Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>62</td>
</tr>
</tbody>
</table>

**Bibliography**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Training-test split for 5-fold cross-validation.</td>
<td>16</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Result of DAS, GSC, MVDR with 0.1 diagonal element value and MVDR-2C with 0.1 diagonal element and \textit{infinite} target-to-interference constraint ratio, the angle $\alpha$ between target and interference is 180°.</td>
<td>40</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Vary diagonal loading value for conventional MVDR and MVDR-2C with \textit{infinite} $C_r$ constraint.</td>
<td>41</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Result of different constraint numbers of MVDR-2C, while diagonal loading values are 0.1 (top table) and 0.01 (bottom table), and female is treated as the primary target.</td>
<td>41</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>The lookup table to map pixel points in the image domain to the real-world coordinate system. The origin of the image coordinate system is the center of the image, and x-axis is towards the left. It is assumed that the origin of Cartesian coordinates is the center of the ellipse; the x-axis is the minor axis for ellipses. $a$ and $b$ are the height and width of ellipses, respectively. $\pm x$ shows whether the image point is located on the positive x-axis or negative x-axis. L/R represents the image captured from the left/right hemispherical lens.</td>
<td>52</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1 Ricoh Theta V camera. Front view (left), rear view (middle), and side view (right). .................................................. 3
Figure 1.2 Example of the fisheye image read directly from a hemispherical lens. .................................................. 4
Figure 1.3 Example of equirectangular panorama. .................................................. 5
Figure 2.1 Image extrapolation: (a) Final extrapolated image. (b) Extrapolation carried out as interpolation between two copies of the image. .................................................. 9
Figure 2.2 Six square patches, evenly spaced along the width, are extracted from an extended image. .................................................. 10
Figure 2.3 Square patch (left) converted to a circular patch with fisheye-like distortion (right). .................................................. 11
Figure 2.4 Sample image from our FDDB-360 dataset. .................................................. 12
Figure 2.5 Each image is sampled into six square patches, whose length of the side is equal to half-width of the image. .................................................. 13
Figure 2.6 Sample fisheye image with its segmentation mask from VOC-360. . .................................................. 15
Figure 2.7 Directory structures of Wider-360 and VOC-360. .................................................. 15
Figure 2.8 (a) Precision-recall curves on FDDB-360 for HR and HR-360. (b) True positive rate versus the number of false positives for HR and HR-360 on FDDB-360. .................................................. 17
Figure 2.9 Distribution of all false negatives for HR (right) and HR-360 (left). . .................................................. 18
Figure 2.10 Locations of all ground-truth faces in FDDB-360. .................................................. 18
Figure 2.11 Face detection results by HR (left) and HR-360 (right) on a real fisheye image with several faces along the perimeter. Detected faces are indicated by yellow rectangles. .................................................. 19
Figure 2.12 Difference between perspective projection for pinhole cameras and equidistance projection for fisheye cameras. \( p_c = (x_c, y_c) \) is the conventional image point of object while \( p_f = (x_f, y_f) \) is the fisheye image point. .................................................. 20
Figure 2.13 Fisheye image from the Ricoh Theta V 360-degree camera (left), image generated by our proposed method (middle), and an image generated by the method from [19]. .................................................. 21
Figure 3.1  Geometric distribution of different microphone arrays: Linear (left), Planar (middle), and Circular (right). Number 1 - N represent the microphone sensors, symbol P is the audio source. 24

Figure 3.2  Input source localization in far-field (left) and near-field (right). 24

Figure 3.3  Block diagram of the generalized sidelobe canceller (GSC). 26

Figure 3.4  Jet colormap, the strength represented by the color is enhanced from left to right. Each point is an RGB vector which represents one color. This colormap is scaled to $[-1,1]$. 32

Figure 3.5  3D radiation patterns generated by Delay-and-Sum beamformer at frequency 400Hz, 1400Hz, 2000Hz, 4400Hz, 5600Hz and 8000Hz. The blue dot at $\left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right)$ represents the desired sound source with distance to the origin equals to 1, and the line connected the origin and the red dot at $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right)$ shows the direction of interference. 34

Figure 3.6  Comparison of 3D power patterns between DAS, MVDR and MVDR-2C at frequency 400Hz, 1400Hz, 2000Hz, 4400Hz and 5600Hz are shown in dB. 36

Figure 3.7  Signal-to-interference ratio between target and interference from frequency 400Hz to 8000Hz. Area under the curve (AUC) is shown in the legend. 37

Figure 3.8  Left is the simulation setup of virtual acoustic environment. Eight black dots represent microphones which uniformly distributed among a circular array with radius of 0.055 meters. The coordinate of the array center is $[2.5, 2.5, 1]$, with the unit in meter. Right figure gives the preview for the locations of target/interference relative towards the array center. 38

Figure 3.9  Experiment setup. Number 1 - 8 represents the location of eight microphones in our system coordinate. Female and male signals are one meter away from the coordinate origin. Two speeches are places in opposite direction ($180^\circ$ difference). 40

Figure 3.10  (a)-(d) show SDR, SINR, PESQ and STOI evaluated in a free space among different beamformers, respectively. The angle difference along x-axis represents the angle between target and interference relative to the center of microphone array. Results shown for MVDR and MVDR-2C here are associated with two diagonal loading values: 0.1 and 0.01. Infinite $C_r$ constraint of MVDR-2C is considered. 43

Figure 3.11  MVDR-2C beam patterns (in dB) at frequency 1400 Hz with a specific input signal when $\alpha$ is (a) $45^\circ$ and (b) $90^\circ$. 44
Figure 3.12 Results for objective evaluation in a **reverberant room**. α represents the angle between the target and interference. The MVDR and MVDR-2C are tested when the diagonal loading parameters are set to 0.1, and $C_r$ for MVDR-2C is infinite.

Figure 4.1 MATRIX Creator front view (**left**) and back view (**right**) with 8 microphones: M1 to M8.

Figure 4.2 **Left:** Rig for data collection. **Right:** Received max amplitude vs. frequency of eight Microphones on MATRIX Creator.

Figure 4.3 Visually guided beamforming system overview.

Figure 4.4 Experimental set-up.

Figure 4.5 One frame of dual-fisheye videos from Ricoh Theta V.

Figure 4.6 Cameras with two fisheye lenses can capture the entire surroundings; each lens covers the half-space. The whole $360^\circ$ area is split into 12 sub-regions. The shaded portion shown on the right represents the overlapped area due to the ultra-wide FOV.

Figure 4.7 (a) Fisheye FOV is split into 7 regions. (b) Ellipse model used for the lookup table.

Figure 4.8 Experimental setup for DOA estimation. α is the angle between target and interference; β is the angle of the estimated target (‘Female’) deviate from the ideal scenario (‘Female’).

Figure 4.9 SDR (**left**) and STOI (**right**) changed with different error angle β among four beamformers. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDDB-360 described in Section 2.5.

Figure 4.10 SDR (**left**) and STOI (**right**) with different error angle β among four beamformers. Solid lines represent the mean, and the semi-transparent region shows one standard deviation around the mean. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDDB-360 described in Section 2.5. The experiments were conducted under **free space**.

Figure 4.11 (a) SDR changes for MVDR-2C with various DOA estimation error for interference under a **free space** condition, corresponding to different target-and-interference position settings. (b) SDR for MVDR-2C with various DOA estimation error for interference in a **free space** environment.
Figure 4.12  SDR (left) and STOI (right) with different error angle $\beta$ among four beamformers. Solid lines represent the mean, and the semi-transparent region shows one standard deviation around the mean. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDBB-360 described in Section 2.5. The experiments were simulated in a reverberant room.

Figure 4.13  SDR changes for MVDR-2C with various DOA estimation error for interference in a reverberant environment, corresponding to different target-and-interference position settings.
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>Array Factor</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under The Curve</td>
</tr>
<tr>
<td>DAS</td>
<td>Delay-and-Sum</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DOA</td>
<td>Direction-of-Arrival</td>
</tr>
<tr>
<td>FDDB</td>
<td>Face Detection Data Set and Benchmark</td>
</tr>
<tr>
<td>FFT</td>
<td>fast Fourier transform</td>
</tr>
<tr>
<td>FN</td>
<td>False Negatives</td>
</tr>
<tr>
<td>FOV</td>
<td>Field-of-View</td>
</tr>
<tr>
<td>FP</td>
<td>False Positives</td>
</tr>
<tr>
<td>GSC</td>
<td>Generalized Sidelobe Canceller</td>
</tr>
<tr>
<td>HR</td>
<td>Hybrid-Resolution</td>
</tr>
<tr>
<td>ISM</td>
<td>Image Source Model</td>
</tr>
<tr>
<td>LISTEN</td>
<td>Locating Individual Speakers and Tracking ENvironment</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short Term Memory</td>
</tr>
<tr>
<td>MVDR</td>
<td>Minimum Variance Distortionless Response</td>
</tr>
<tr>
<td>MVDR-2C</td>
<td>Minimum Variance Distortionless Response with two Constraint</td>
</tr>
<tr>
<td>PESQ</td>
<td>Perceptual Evaluation of Speech Quality</td>
</tr>
<tr>
<td>RIR</td>
<td>Room Impulse Response</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristics</td>
</tr>
<tr>
<td>VGHA</td>
<td>Visually Guided Hearing Aid</td>
</tr>
<tr>
<td>VOC2012</td>
<td>Dataset PASCAL Visual Object Classes 2012</td>
</tr>
<tr>
<td>SDR</td>
<td>Signal-to-Distortion Ratio</td>
</tr>
<tr>
<td>STFT</td>
<td>Short-Time Fourier Transform</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal-to-Interference-and-Noise Ratio</td>
</tr>
<tr>
<td>SIR</td>
<td>Signal-to-Interference Ratio</td>
</tr>
<tr>
<td>STOI</td>
<td>Short-Time Objective Intelligibility</td>
</tr>
<tr>
<td>SSL</td>
<td>Sound Source Localization</td>
</tr>
</tbody>
</table>
TP  True Positives
WSJ  Wall Street Journal
Nomenclature

$(\theta_0, \phi_0)$ the direction of maximum radiation
$(w, h)$ the width and weight of face bounding box in FDDDB
$\alpha$ the angle between target and interference signal
$\beta$ the angle of estimated DOA error
$\gamma$ the scaling factor for SDR
$\kappa$ the wave number
$\lambda$ the wavelength
$\lambda_0$ the rate of change of the first constraint being optimized
$\lambda_1$ the rate of change of the second constraint being optimized
$\mathbb{E} [\cdot]$ the expected value
$\tilde{C}$ the regularized covariance matrix
$i$ the single-channel interference
$L(\cdot)$ the Lagrange multiplier
$\mathcal{N}(0, \sigma^2)$ the normal distribution with mean 0 and standard deviation $\sigma$
$\mu$ the diagonal loading value
$\|\cdot\|$ the L2 norm
$\odot$ the element-wise multiplication
$\phi$ the azimuth angle
$\phi_n$ the angular position of the $n^{th}$ microphone sensor
$\sigma$ the standard deviation for $\beta$
$A$ the steering matrix
$B$ the blocking matrix in GSC
$C_{nn}$ the covariance matrix of noise and interference
$C_{ss}$ the covariance matrix of the desired signal
$C_{xx}$ the covariance matrix of the input signal
$h_0$ the steering vector towards the direction of desired signal
$h_1$ the steering vector towards the direction of the specific interference
$I$ the identity matrix
$n$ the ambient noise
$S$ the multi-channel input signals without noise
$s_0$ the single-channel desired signal
the considered single-channel interference in MVDR-2C
the weight matrix for the beamformers
the weights for the adaptive noise canceller in GSC
the adaptive weight at n\(^{th}\) microphone and index k
the fixed weights in GSC
the optimal weights of the beamformer
the multi-channel signals from the microphone array
the matrix of input signals in frequency domain
the output signal of the beamformer
the output of the adaptive noise canceller in GSC
the output of the blocking matrix in GSC
the output of the blocking matrix at n\(^{th}\) microphone and index k
the output signal from the fixed beamformer in GSC
the elevation angle
the frequency of the signal
the speed of sound in air
the constraint number for desired signal
the constraint number for the specific interference
the target-to-interference constraint ratio
the exponential function
the camera focal length
the Hermitian transpose
the number of total samples in input signal
the number of speakers
the number of microphone sensors
the index of microphone sensors
the radius of the circular array
the radial distance
the weight at n\(^{th}\) microphone
Chapter 1

Introduction

1.1 Motivation

The fusion of visual and acoustic information contributes to better modeling and understanding of a scene, which can be used profitably in different applications, such as telecommunication, hearing aids, speaker diarization and speaker tracking. Recent literature suggests that incorporating auditory information shows remarkable improvement in computer vision tasks. Audio-visual systems described in [43, 52] demonstrate the possibility to track multiple speakers in a room, while the speakers are relatively unconstrained and the need for lapel microphones is avoided.

Compared with the limited performance of a traditional microphone array in a multi-talker scenario, microphone arrays coupled with vision sensors deliver steadily enhanced beamforming performance. For beamforming techniques with a conventional microphone array, the orientation of the desired signal is usually assumed to be known in advance or could be estimated by algorithms like sound source localization (SSL). In a multi-speaker environment or noisy area, the array tends to choose the loudest sound source as the steering target. The array is treated as ‘blind’ since it cannot accurately identify the desired speaker due to such a ‘cocktail party effect’ [16].

Attracted by the benefits derived from multi-sensor networks, there are already some research studies and applications that talk about the combination of vision and audio processing. A laboratory-based prototype ‘visually guided hearing aid (VGHA)’ is implemented in [36], which outperforms the sophisticated hearing aid systems that do not use visual information. VGHA exploits the direction of eye gaze as cues to selectively amplify the desired talker in a multi-speaker situation. One common application to combine camera sensors and microphone array is video conferencing. Applications such as [62] make use of microphone array to localize an active speaker in a conference, and then provide a high-resolution video of the located speaker. However, the SSL method is computationally expensive and has poor performance when the distance between microphones in the array is small. Work from [9] illustrates that beamforming with more precise knowledge of sound source location
gained visually boosts the word recognition accuracy of moving speakers. [32] offers another example of improved speech enhancement assisted by visual component. A LISTEN (Locating Individual Speakers and Tracking ENvironment) system proposed in [17] exploits skin color detector, movement detector and face detector to locate and track a speaker and perform beamforming immediately. However, it uses a harmonically nested linear array and is limited to one active speaker in a room of small to medium size. A patent recently issued to Microsoft [57] is another embodiment of a visually-guided beamforming system, which demonstrates a combination of a microphone array and a single fisheye camera. The microphone array is arranged in a hexagon shape with an additional microphone placed in the center. This embodiment uses a long short term memory (LSTM) model to fuse audio and vision information, which aims at improving perceptual content for artificial intelligence (AI) applications.

In this thesis, we wish to simulate and examine an application that uses a 360° camera to provide cues for beamforming in a circular microphone array. Specifically, the camera is used to locate the talker, instead of considering the loudest speaker as the target. Many studies assume only one speaker active at a time. Here we will look for enhancing beamforming ability towards the wanted orientation in the presence of multiple interfering signals/speakers and noise.

1.2 Face/Object Detection Using 360° Cameras

Face/object detection is a poster problem of computer vision, with applications in surveillance, security, biometrics, human-computer interaction, and other areas. With the recent advances in deep learning, the problem of face/object detection in conventional 2D images has largely been solved. For example, on the well-known Face Detection Data Set and Benchmark (FDDB) [35], recent models such as [40, 63] reach near 100% true positive rate with very few false positives. However, detection in other signal domains, such as 360° images, point clouds [49], or compressed bitstreams [3, 4], has been less explored.

Conventional cameras only have a single lens and limited field-of-view (FOV). In other words, such a camera can only observe the direction it is aimed at. Due to the limited visual coverage of single-lens cameras, there is an increasing interest in cameras with the ability to capture the entire surroundings simultaneously. Those images are called panoramic images or 360° images. One way to realize the 360° capturing is to use multiple single-lens cameras as a system, and multi-lenses cameras are the other option. The most common panoramic cameras comprise two fisheye-lenses with wide FOV back-to-back, see Fig. 1.1.

One of our goals is applying face/object detection on 360° images. One way to accomplish the detection, in this case, would be to project the 360° image onto a set of 2D images and then employ one of the conventional detection models on these 2D images. An approach similar to this, where an equirectangular panorama image (a post-processed form of a 360°
Another recent work [38] proposes to generate a spherical image and adjust convolution kernels and pooling operators to work in spherical coordinates, so that Convolutional Neural Network (CNN) - based detectors could operate directly on spherical coordinates.

Compared to traditional cameras and multi-camera systems, the compact dual-fisheye apparatus can be easily located at any position and provide a panoramic view at the same time. Many 360-degree cameras are released in the market already. Popular dual-lens 360° cameras (e.g., Ricoh Theta, Samsung Gear 360, Insta 360, etc.) make 360-degree image/video capturing more convenient and portable. These dual-lens cameras have two hemispherical ("fisheye") lenses, each projecting a 180° view onto an imaging sensor. The images read directly from the sensors are circular in appearance and have strong barrel distortion towards the perimeter of the circle, as depicted in Fig. 1.2. Equirectangular image is the most popular format to represent the whole spherical scene in a single figure, usually stitched from dual-fisheye images [29]. The field-of-view of this kind of image is 360° horizontally and 180° vertically, as shown in Fig. 1.3. Both the equirectangular panorama used in [60] and the spherical-coordinate image used in [38] are obtained from these fisheye images via post-processing. Normally, the post-processing includes the image projection, dual-fisheye image-stitching, and distortion correction [29, 44]. Equirectangular and Mercator projection are two common methods for displaying spherical objects onto a 2D plane; they define the rule to assign pixel points on a spherical surface to a flat image. Specifically, equirectangular projection converts both meridians and circles of latitude on the sphere into straight lines with constant spacing, which is also known as equidistant cylindrical projection, while Mercator is a conformal projection.

Due to the nature of the fisheye lens, curved straight lines and distorted objects make object detection in 360° panoramic images more challenging. The images in equirectangular format tend to have high resolution, and the distortion in the image varies among the different regions of the image. As depicted in Fig. 1.3, the hand on the south-east/west
region has been severely distorted compared to the objects and the face in the middle part of the image. Besides, various camera lenses also introduce lens-specific distortion.

### 1.3 Beamforming

Multi-sensor systems are being developed to satisfy the increasing demand for monitoring the streets, public spaces, and the environment, which is driving many research fields, such as vision, sonar, radar, and wireless communication [26][56]. In the audio world, beamforming is a versatile solution to spatial filtering the sound information gathered from microphone array systems. Beamforming coupled with an acoustic microphone array shows significant improvement in applications like automatic speech recognition (ASR) [51], hearing aids [27], audio conferencing [37][46], sound source localization (SSL) [8][45], etc. The core of beamforming is enhancing acoustic signals from the desired direction with minimum distortion while suppressing environmental noise and interference from other directions.

The following challenges are well-known in the beamforming research:

- Coherent target and interference: beamforming performance degrades when the desired signal and interference are coherent or correlated. For example, when a beamformer amplifies the signal from a specific wanted direction, it also enhances the noise and the interference from that direction.

- Reverberation: sound reflection happens everywhere in the real world; walls, furniture, people, and even air absorbs and reflects sound. According to [15], the reverberation can result in an error in direction-of-arrival (DOA) estimation and sound source local-
ization, since in some cases the direction of the dominant reflected path has a stronger signal than the direct sound propagation path.

- Sensor calibration: A real multi-microphone sensor array may need to be calibrated in terms of phase shift and amplitude gain. Research from Microsoft [54] concludes that microphone array is more sensitive to the variations in magnitude gain than the variations in phase shift. Besides, some microphones also have direct current (DC) offset (a mean amplitude displacement from zero) due to the circuit design.

Smart speakers with circular microphone array (e.g. Amazon Echo, Apple HomePod) available in the market offers the solution to hands-free speech interaction. The circular and planar array has the ability to pick up all the acoustical signal in a room, while a linear array presents the front-and-back ambiguity. In other words, the linear microphone array can only steer the radiation beam in a range of \((-\frac{\pi}{2}, \frac{\pi}{2})\), and cannot distinguish the signal arrived from front or back. [33] also presents that the circular array for beamforming outperforms the linear array in terms of speech quality and intelligibility. In consequence, the circular array with omnidirectional listening ability presents an excellent match to 360° cameras.

1.4 Thesis Contribution and Outline

This thesis is organized as follows.

The problem of face/object detection in fisheye images, along with three new fisheye-looking image datasets are discussed in Chapter 2. In order to examine the benefits from the new datasets, a detector trained on regular images has been re-trained and tested on fisheye images. In addition, the comparison between our proposed fisheye conversion model and an existing fisheye model is presented.
In Chapter 3, a literature review of beamforming technologies is given, and the detailed algorithms of three chosen well-known beamformers, Delay-and-Sum (DAS), Generalized Sidelobe Canceller (GSC) and Minimum Variance Distortionless Response (MVDR) are studied. Based on the conventional MVDR, an improved MVDR with an additional constraint is proposed in Section 3.3, which aims at maximizing the power ratio between a specific target signal and one interfering signal. To examine the performance of different beamformers on a circular array, and to find out the factors which influence their behaviors, several computer-simulated experiments have been carried out with results listed and compared in Section 3.5.3.

Chapter 4 reports how the face detection from Chapter 2 and beamformer from Chapter 3 can be combined. The platform called MATRIX Creator is used as the array of eight microphones, while Ricoh Theta V is exploited as the ‘eye’ with a panoramic view of this system. The full system overview and its details are explained. Finally, Section 4.4 discussed the impact of the imprecise face detection on our visually-guided beamforming system. In other words, the feasibility of this system has been examined.

In the end, Chapter 5 provides a conclusion of this work and suggests possible future directions.

This thesis research has resulted in the following publications:

**Conference papers**


**Journal papers**

Chapter 2

Face/Object Detection in Fisheye Images

2.1 Introduction

Topics such as object detection, segmentation and classification are fundamental to computer vision, and most of the state-of-the-art algorithms are based on deep neural networks trained on large-scale datasets collected by the regular cameras. These days, an increasing number of applications in the area like autonomous driving and surveillance need a wide field-of-view (FOV) to perceive the surroundings. Modern 360° cameras that use wide FOVs offer the possibility to cover a large area, for example an entire room, without using multiple distributed vision sensors. This kind of camera is usually formed by two or more ‘fisheye’ camera modules; each has a wider FOV than a standard lens. Most popular 360° cameras released in the market are made with two fisheye lenses placed back-to-back. The images captured by such a camera are output as a stitched panorama by internal processing, and the actual raw data presented as a pair of images in a dual-fisheye format. There are usually some artifacts along the edge of stitching, but they are subtle enough to be ignored for many viewing applications.

Fisheye lenses tend to have strong radial barrel distortion, which generates displacements between the ideal pixel position from a rectilinear pinhole camera and the actual points on the fisheye image plane. The displacements are growing non-linearly towards the perimeter of the image. It is evident that the geometric distortions introduced by fisheye lenses make computer vision problems more challenging. Many applications related to computer vision problems choose to un-distort/un-wrap the fisheye images before the subsequent tasks [10, 39, 61]. The estimated un-distorted models have limitations. For example, those designed models require to estimate the camera’s intrinsic parameters and distortion matrix; the corrected images degrade in image quality and resolution.

Clearly, it would be much more efficient to perform face/object detection directly on fisheye images; this would enable the detector to operate closer to the sensor and circumvent
unnecessary processing steps. However, training detectors on fisheye images is challenging due to the lack of annotated datasets for fisheye images; models trained with conventional images have an unsatisfactory performance on fisheye images. Few datasets in equirectangular panorama, like SUN360 [58] for scene recognition, are publicly available. Yet until now, none of the datasets in the fisheye format are published with labelled ground-truth for face and object detection problems. To exploit existing conventional datasets and avoid the high cost of human effort on new dataset collection and ground-truth labelling from scratch, we choose to manipulate some original datasets and apply our proposed fisheye distortion model to them. A comparison between our proposed fisheye model and an existing fisheye model is presented in Section 2.6. The material in this chapter has also been presented in [23, 24].

In this chapter, we introduce three new fisheye-like dataset for face/object detection, which we call FDDB-360, Wider-360 and VOC-360. We also show how a detector trained on regular images can be re-trained for face/object detection in fisheye images. All three 360° datasets are available online at http://multimedia.fas.sfu.ca/data/

2.2 FDDB-360

The purpose of FDDB-360 is to enable the training of models for detecting faces in fisheye images. To this end, we started from Face Detection Data Set and Benchmark (FDDB) [35], a well-known dataset for face detection. According to [35], this dataset contains annotations for 5171 faces in a set of 2845 images taken from the Faces in the Wild [7] dataset.

We used the annotated images from the original FDDB dataset, and from them create a number of images that have the appearance of fisheye images. The face locations in the original FDDB are specified by ellipses, but we first converted them into rectangles to enable easier processing.

Fisheye images have the least geometric distortions near the center. As we move towards the perimeter, the degree of distortions increases. Face detectors trained on conventional 2D images would likely be able to detect faces near the center of a fisheye image, but detection becomes more challenging away from the center. For this reason, we wanted to create fisheye-looking images that would have a sufficient number of faces away from the center, so that the detector can learn the appearance of distorted faces. The steps taken to create the new dataset are described in the following subsections.

2.2.1 Image Extrapolation

A number of images in the original FDDB have faces near the center. To facilitate sampling the images with patches where the face locations could be arbitrary, we widened all images by 40%, by extending it 20% on both left and right as shown in Fig. 2.1(a). This requires
image extrapolation, which is inherently difficult due to the absence of boundary conditions on three sides of the extensions.

To overcome this challenge, we used the strategy illustrated in Fig. 2.1(b). We created a copy of the image on the right side of the original, spaced away by 40% of the original width, and then interpolated between the two copies of the image. Though still challenging, this is an easier problem than extrapolation in Fig. 2.1(a) because of a larger area where boundary conditions exist. After interpolation, the interpolated area is split in half, and the right half of it (shown in red in Fig. 2.1) is moved to the left side of the image, to complete the extrapolated image.

Interpolation is carried out using the inpainting algorithm from [5], which is an extension of the well-known Criminisi et al. method [18]. While performing inpainting, we exclude face and skin regions from being used for inpainting. This is because, if the inpainted region ends up with patches containing partial human face, it might confuse the model during training, whether or not these partial faces are actually annotated as faces. To avoid this, while performing inpainting, we increase the cost [5, 18] of a patch for any patch overlapping with an annotated face, or where skin color is detected [11].

The approach mentioned above was applied to the majority of images in FDDB. However, about 34% of FDDB images have a width-to-height ratio of less than 3:4. For such images, even 40% width extension still gives a relatively narrow image. In these cases, we did not rearrange the extended image as shown in Fig. 2.1(a), but left it in the format shown in Fig. 2.1(b), with a copy of the original image on the right side.

2.2.2 Fisheye-like Distortion

Each extended image is sampled using square patches, which are evenly distributed along its width as shown in Fig. 2.2. In total, six square patches are extracted from each extended image. Subsequently, fisheye-like distortion is applied to each square patch.
Fisheye distortion models usually involve intrinsic camera parameters\(^1\) and various lens distortion parameters [20]. To avoid making distortions camera- and lens-specific, we adopted a simpler approach.

Consider a square patch extracted from an extended image, as shown in Fig. 2.2. We first map this square patch to a circular patch. Let \((x, y)\) be the normalized coordinates of the square patch, such that the patch center is \((0, 0)\) and the four corners have coordinates \((\pm 1, \pm 1)\). The square patch is mapped to a circular patch using the following coordinate mapping:\(^2\)

\[
(x', y') = \left(x\sqrt{1 - \frac{y^2}{2}}, y\sqrt{1 - \frac{x^2}{2}}\right) .
\]  

Such mapping introduces radial distortion, where straight lines get bent towards the perimeter of the circle. To add barrel distortion, which manifests itself as “squeezing” towards the perimeter, we further scale the coordinates by a factor that decreases towards the perimeter. Specifically, the mapping is

\[
(x'', y'') = \left(x' e^{-r^2/4}, y' e^{-r^2/4}\right),
\]  

where \(r = \sqrt{(x')^2 + (y')^2}\) is the radial distance from the center of the patch. This form of exponential squeezing was chosen empirically to visually approximate the appearance of fisheye images. An example of a square patch converted to a circular patch with fisheye-like distortion is shown in Fig. 2.3.

### 2.2.3 Annotation

Once square patches are converted to circular patches, face locations have to be appropriately mapped to the new coordinates. As mentioned earlier, the location of each face in the

---

\(^1\)https://docs.opencv.org/3.4/db/d58/group__calib3d__fisheye.html

\(^2\)https://www.xarg.org/2017/07/how-to-map-a-square-to-a-circle/
Figure 2.3: Square patch (left) converted to a circular patch with fisheye-like distortion (right).

original FDDB is specified by an ellipse, but we converted those locations to rectangles to simplify further mapping to the circular patch.

We selected eight points from the bounding box for each face (four corners and four edge midpoints), as illustrated in red in the left part of Fig. 2.3. If a part of the bounding box fell out of the boundaries of the square patch, we trimmed its coordinates to coincide with the patch boundary. Once the mapping (2.1)-(2.2) is applied to these eight points, they are mapped to the squeezed circular coordinates as shown in the left part of Fig. 2.3. While it is possible to store these eight points (or more, if higher precision is needed) as the bounding polygon for the face in circular coordinates, we decided to simplify the annotations and again use bounding boxes. Hence, we selected the minimum bounding rectangle of the polygon as the annotation for the face location. This is illustrated by a green rectangle in the right part of Fig. 2.3.

When extracting square patches from the extended image, it is possible that faces are cropped and that only a part of the original face falls into the square patch. If the overlap between the original bounding box and the part that is inside the square patch is over 50%, we kept that annotation and mapped it to the circular patch as explained above. Otherwise, we treated the face as incomplete and did not convert the corresponding annotation to the circular patch.

2.2.4 Further Details of FDDB-360

After applying the procedures described above, we ended up with 16,788 fisheye-looking images and a total of 26,811 annotated faces. A sample image from FDDB-360 is shown in Fig. 2.4. Face locations are provided as the bounding box parameters \((x, y, w, h)\), where \((x, y)\) is the location of the top-left corner and \((w, h)\) are the width and height of the bounding box, respectively.
Note that the bounding box need not be fully contained within the circular patch, especially for faces that are near the perimeter, as illustrated in the right part of Fig. 2.3. From these coordinates, one can find the intersection of the bounding box and the circle if a more precise localization of the face is required.

### 2.3 Wider-360

FDDB-360 helps to increase the face detection accuracy in fisheye images. We will discuss the performance of the new face detector re-trained by FDDB-360 later in Section 2.5. However, the original images from FDDB dataset are collected from search engine Yahoo! news website. Many images are close-ups and faces take a large portion of the image. The faces collected in FDDB have limited variations in scale; the positions and scenes of where the faces appear are also limited. In a real-world application of face detection, faces appear in different scenes with a high variation scale, pose, facial expression and occlusion. For this reason, we extend our face dataset using Wider Face [59]. This is a well-annotated dataset for face detection, which consists of 32,203 images with 393,703 labelled faces. There is a significant variation in the sizes of faces; they can be grouped into different scales: small (10-50px), medium (50-300px), and large (over 300px), according to the height of faces. This provides additional variation in scale of faces for our dataset.

Wider Face is divided into 61 event categories, and multiple attributes have been annotated, for example, lighting condition, facial expression, and face bounding box. Wider Face labels the faces as ‘Easy’, ‘Medium’ and ‘Hard’ based on the level of detection difficulty, and separates the whole dataset into training, validation and testing set, with a percentage
of 40, 10 and 50, respectively. Because Wider Face does not provide the ground-truth face annotations for testing set, we only deal with the images in the training and validation sets.

For every image from the Wider Face dataset, we applied an algorithm to sample the original image into six square sub-images. We first rotated images whose width was less than height by 90° clock-wise. This step makes all the images have landscape orientation. Afterwards, each image was sampled into six patches as depicted as Fig. 2.5. The side length of each square is equal to the half-width of the image in landscape orientation. There is one particular case when the height of the original image is twice its width; we applied the same sampling procedure as described in Section 2.2.1. Those images that have been rotated were rotated back to their original position.

The sampled images were distorted using our proposed fisheye distortion model mentioned in Section 2.2.2. At the same time, the face ground-truth boxes were mapped into the new coordinates as well. In particular, we chose eight points from each bounding box of the original image and then followed the same steps described in Section 2.2.3. The annotations of training set and validation set were recorded in two separate MAT files. We decided to keep the same annotation structure as Wider Face. New ground-truth labels were updated, and other attributes of each face were kept unchanged and copied into the new annotation files.

The above procedure creates our new fisheye dataset: Wider-360. Wider-360 contains 63,897 images, of which 50,982 images are intended for training and 12,915 images for validation, as shown in Fig. 2.7(b). Following the original Wider Face dataset[59], the images are organized into 61 directories depending on the type of the scene, as shown in Fig. 2.7(c). The annotations are in the form of face bounding boxes and are stored in two MAT files, one for the training set and another for the validation set.

2.4 VOC-360

Dataset PASCAL Visual Object Classes (VOC) [22] provides a number of images for three object-related computer vision tasks: detection, classification and segmentation. The latest
version VOC2012 contains 20 classes and 17,125 images in total. Every image has one corresponding XML file to store all the necessary information, including image size, ground-truth of objects, and classes to which the objects belong. The number of images for segmentation in VOC2012 increased substantially compared to the earlier versions. The sizes of images vary, but most of the widths and heights are within the range from 375 to 500 pixels. We started from VOC2012 to build a fisheye dataset for object detection and segmentation, which we named VOC-360.

The average file size of VOC2012 images are 112KB, and the objects in the images are located in various positions. Besides, the images’ width and height are ordinarily close to each other. Based on these properties, we designed a different method to pre-process the images into square patches. Same as the steps mentioned in the previous section, all the images whose width is narrower than height were rotated by 90° at the first stage. Then, each image was equally sampled into several square patches along the width. The number of patches depended on the width-to-height ratio of the original input image and was set to be six initially. Due to the varying width-to-height ratio of each image, we set a gap limit as one-fourth of the image’s height. The gap between every two sub-images is defined as the difference between width and height divided by the total number of sub-images. The number of square patches was updated recursively to satisfy the condition that the gap cannot exceed the gap limit. After sampling them into square patches, every patch from the images that have been rotated initially was rotated back to the original orientation.

Next, the same fisheye distortion model, as discussed in Section 2.2.2, had been applied to each patch. VOC2012 contains a special sub-dataset for segmentation, where every object appearing in the image is annotated with pixel-wise segmentation. Those segmentation images were sampled and distorted utilizing the same processes as above. Subsequently, each distorted square patch had it’s own XML annotation file, and the objects that remained with over 50% in the new fisheye image were mapped to new coordinates, as well as their bounding boxes. The original VOC2012 uses 4 parameters \((x_{\text{top\_left}}, y_{\text{top\_left}}, x_{\text{bottom\_right}}, y_{\text{bottom\_right}})\) to present the bounding box for one labeled object. The conversion of the annotation box adopted the identical method as illustrated in Section 2.2.3: eight points from the ground-truth in the original image had been selected and mapped into the new coordinate. The segmented images do not need further modification in annotation since the segmentation can be represented using RGB colors as shown in Fig. 2.6.

As the final result, VOC-360 contains 39,575 images and the corresponding annotations. The images and annotations are derived from the VOC2012 dataset by post-processing. The dataset is organized into five directories: Annotations, fisheye, fisheye_class, and fisheye_object, and ImageSets, as depicted in Fig. 2.7(a), which is similar to the directory structure of the original VOC2012. Each fisheye image inside the ‘fisheye’ folder corresponds to one XML file in the ‘Annotations’ folder, with the same filename. The XML files provide all the annotations for each image, following the same structure as the original
VOC2012 dataset. Directories ‘fisheye_class’ and ‘fisheye_object’ contain the fisheye image masks with pixel-wise segmentation giving the class of the object visible at each pixel. The directory ‘ImageSets’ contains text files that specify the lists of images for training, testing and validation.

2.5 Transfer Learning for Face Detection in Fisheye Images

In this section, we show how one can re-train an existing face detection model to detect faces in fisheye-looking images. Our baseline face detector is the well-known TinyFace [30], specifically their hybrid-resolution (HR) model, which was trained on the Wider Face dataset [59]. We used their pre-trained model and tested it on FDDB-360. The precision-recall curve of the HR model is shown as the green line in Fig. 2.8a, while its ROC curve is shown as the green line in Fig. 2.8b.
Although TinyFace HR is one of the best face detectors on 2D images according to the results on the original FDDB\textsuperscript{3}, its performance on FDDB-360 is not particularly impressive. This is understandable, as the model has not been trained on images that involve fisheye-like distortions. Examination of the model’s predictions shows that indeed, near the center of the circular patches, the detection performance is quite good, but it degrades towards the perimeter where the distortions are stronger.

To show the improvement that can be obtained by transfer learning and re-training, we kept the original ten folds from the FDDB dataset. Experiments were run using 5-fold cross-validation obtained by merging pairs of the original folds, as shown in Table 2.1. In each experimental run, we started with the pre-trained model whose weights were obtained on the Wider Face dataset [59]. We then further trained the model using the software provided by [30], with a learning rate of $10^{-4}$ on one training set shown in Table 2.1 and tested on the corresponding test/validation set.

To account for various orientations and scales of faces that may be found in real fisheye images, we performed data augmentation during training. This included horizontal flipping and re-scaling already implemented in the software provided by [30], as well as newly introduced random rotation by 90°, 180° and 270°. We considered using other rotation angles as well. However, FDDB-360 contains annotations in the form of axis-aligned rectangles $(x_{\text{top\_left}}, y_{\text{top\_left}}, w, h)$ specifying the location of the face. Rotation by angles other than 90°, 180° and 270° would cause a size increase of the resulting axis-aligned enclosing rectangle, and we felt the ground-truth rectangles were already large enough (see Fig. 2.3) and should not be increased further. One possibility to increase the set of viable angles in augmentation in the future would be to switch to a polygonal representation of face locations. We refer to the resulting, re-trained model, as HR-360. Red lines in Figs. 2.8a-2.8b represent the average of five performance curves of HR-360 from the 5-fold cross-validation.

As seen in the figures, re-training with transfer learning from the original HR results in significant performance improvement. On the precision-recall results (Fig. 2.8a), HR achieves the area under the curve (AUC) of 0.864, whereas HR-360 achieves 0.954, as indicated in the figure legend. On the true positives (TP) vs. false positives (FP) test (Fig. 2.8b),

<table>
<thead>
<tr>
<th>Training</th>
<th>Test/Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folds 3-10</td>
<td>Folds 1-2</td>
</tr>
<tr>
<td>Folds 1-2,5-10</td>
<td>Folds 3-4</td>
</tr>
<tr>
<td>Folds 1-4,7-10</td>
<td>Folds 5-6</td>
</tr>
<tr>
<td>Folds 1-6,9-10</td>
<td>Folds 7-8</td>
</tr>
<tr>
<td>Folds 1-8</td>
<td>Folds 9-10</td>
</tr>
</tbody>
</table>

Table 2.1: Training-test split for 5-fold cross-validation.

\textsuperscript{3}\url{http://vis-www.cs.umass.edu/fddb/results.html}
HR-360 achieves around 0.85 TP rate with 200 false positives, while the original HR achieves around 0.72 TP rate for the same number of false positives. The difference of about 0.1 TP rate persists even for higher numbers of false positives.

While HR-360 outperforms HR on FDDB-360 overall, we did find some cases where a face was detectable by HR but not by HR-360. Apparently, transfer learning involves some “forgetting” as well, and while the model gets new capabilities during transfer learning, some of its old capabilities may disappear. One way around it could be to randomly insert the data that the model was initially trained on (in our case, the Wider Face dataset [59]) into the re-training process. However, in keeping with the FDDB-style evaluation, we did not do that, and only used the data from the FDDB-360 folds (Table 2.1) for re-training.

To visualize how accurate are HR and HR-360 depending on the location of the target face, we tested both models on the entire FDDB-360 dataset; and recorded the locations of all false negatives (FN) - the faces that were missed. This is one way to find how a model can be improved. When a face is missed, we record the location of the center of its ground-truth rectangle and normalize it in such a way that the radius of the circular image is 1 (i.e., the image becomes a unit circle). In cases where the center of the bounding rectangle is outside the unit circle, the intersection point of the circle and the line connecting the center points of the bounding rectangle and the circle is used to represent the FN location. The scatter plots of FN points for HR and HR-360 are shown in Fig. 2.9. The left graph shows FN’s of HR, and it is apparent that HR misses 2929 faces among the entire FDDB-360, including many faces near the perimeter of the image, as we expected due to geometric distortions. Meanwhile, the FN’s of HR-360 (Fig. 2.9 right) are more evenly distributed across the unit circle, indicating that the model has learned the corresponding geometric distortions. In other words, the re-trained model can correctly detect more faces near the
edge of the fisheye region. Also, from Fig. 2.9, the FN points of both HR and HR-360 are mainly distributed in the lower part of the fisheye circle. It is because the ground-truth faces are mainly located in the mid-lower part of the circle (see Fig. 2.10).

Finally, we show an example of detection performance on a real fisheye image, rather than an image from FDDB-360. Fig. 2.11 shows an image obtained by a Ricoh Theta V camera with several faces along the perimeter. Image on the left shows the results of HR, where one face was detected, as indicated by the yellow rectangle. Image on the right shows the result of HR-360, which manages to find two faces. There is one more person in the scene (in the left part of the image), whose face is so much out of view that both detectors fail to find it.
Figure 2.11: Face detection results by HR (left) and HR-360 (right) on a real fisheye image with several faces along the perimeter. Detected faces are indicated by yellow rectangles.

### 2.6 Comparison with Other Fisheye Models

In this section, we discuss an alternative approach to generate fisheye-looking images from conventional 2D images. For a conventional image, the perspective projection of a pinhole camera is good enough to describe the relationship between the incoming ray and the location on the image, as in Eq. 2.3:

\[
r_{\text{perspective}} = f \cdot \tan(\theta)
\]

where \(\theta\) is the entrance angle between the principal axis and the projection ray into the image plane, \(f\) is the focal length, and \(r\) is the radial distance from the perspective center. However, for fisheye lenses, the distortion is not equally distributed as in the pinhole camera model. The fisheye model is usually defined based on one of the following projection equations [1]:

\[
r_{\text{equidistance}} = f \cdot \theta
\]

\[
r_{\text{stereographic}} = 2 \cdot f \cdot \tan\left(\frac{\theta}{2}\right)
\]

\[
r_{\text{equisolid}} = 2 \cdot f \cdot \sin\left(\frac{\theta}{2}\right)
\]

\[
r_{\text{orthogonal}} = f \cdot \sin(\theta)
\]
Figure 2.12: Difference between perspective projection for pinhole cameras and equidistance projection for fisheye cameras. $p_c = (x_c, y_c)$ is the conventional image point of object while $p_f = (x_f, y_f)$ is the fisheye image point.

The most frequently used method is equidistance projection expressed as Eq. 2.4. The difference between the projection model of pinhole cameras and fisheye cameras is illustrated in the left part of Fig. 2.12.

An alternative method to generate a fisheye-looking image from a rectangular or square image was proposed in [19], although no public data was provided with that work. The method in [19] is based on equidistant projection and requires the user to specify the focal length of the camera. Eq. 2.8 describes the mapping between a conventional image (captured by a pinhole camera) and a fisheye image proposed by [19].

$$r_f = f \cdot \tan^{-1} \left( \frac{r_c}{f} \right) \quad (2.8)$$

$$r_c = \sqrt{(x_c - x_u)^2 + (y_c - y_u)^2} \quad (2.9)$$

$$r_f = \sqrt{(x_f - x_u)^2 + (y_f - y_u)^2} \quad (2.10)$$

In this method, the two projection models share the same principal point $p_u = (x_u, y_u)$ and the focal length could be adjusted based on experiments. In Eq. 2.9, $r_c$ is the radial distance between the image point and the principal point in the conventional image. As shown in Fig. 2.12, for a certain object point, the azimuth angle $\phi$ is identical for both perspective
projection and equidistance projection. Using this property, every conventional image point \((x_c, y_c)\) has a corresponding fisheye image point \((x_f, y_f)\), as shown in Eq. 2.11 - 2.13.

\[
\phi = \tan^{-1} \left( \frac{x_c}{y_c} \right) \tag{2.11}
\]

\[
x_f = r_f \cdot \cos(\phi) \tag{2.12}
\]

\[
y_f = r_f \cdot \sin(\phi) \tag{2.13}
\]

However, the method illustrated above depends on the focal length of the fisheye camera.

In Fig. 2.13, we show three images: a real fisheye image taken by the Ricoh Theta V 360-degree camera (left), an image generated from a square original by our proposed method from Section 2.2.2 (middle) and an image generated from the same original by the method from [19] (right) using one of the suggested focal length values \((f=242)\). The image generated by our method (middle) is a closer approximation to a real fisheye image (left).

2.7 Conclusions

In this chapter, we illustrated the process of generating fisheye-looking datasets: FDDB-360, Wider-360 and VOC-360, which are derived from three well-known datasets: FDDB, Wider Face, and VOC2012, respectively. Generating new datasets based on existing datasets helps to save human time and effort of collecting information and annotating the target. The images from the original datasets were sampled into patches, and each patch was converted through a proposed fisheye-looking distortion model. Meanwhile, the face annotations were also mapped to the new coordinate system. We also showed that re-training (using transfer learning) an existing face detector on FDDB-360 significantly improves its face detection performance on this kind of image.
The fisheye distortion method we proposed is neither camera-specific nor lens-specific. Although the model does not match every single fisheye lens, it generates realistic fisheye-looking distortions. A comparison of our proposed distortion model with an existing fisheye model was made at the end of this chapter. It was shown that, compared with a real fisheye image captured by the Ricoh Theta V camera, the image generated by the proposed fisheye model is a closer and more reasonable approximation.
Chapter 3

Circular Array Beamforming

3.1 Introduction

Beamforming spatially filters out the interference and environmental noise while keeping the signals arriving from the desired direction. It has been studied in many applications, such as radar, sonar, surveillance and communications [41]. Using beamforming, a microphone array is able to estimate the direction-of-arrival (DOA) of sound sources, while a conventional single microphone cannot do that. The combination of microphone array and beamforming algorithm an important part of improving communication quality over long distances in a noisy environment.

Acoustic beamforming works with the signals received by the acoustic array, which usually consist of several components: target signal, interference and ambient noise. The beamformer generates a beam towards the target of interest in a specific radiation pattern. The pattern shows the ‘gain’ (or ‘weight’) the beam has in various directions. The higher the gain, the further the beam can reach in that direction. In this way, the beamformer achieves the goal of listening to the signals arriving from the desired direction and mitigating the noise and interference from other directions. The beam with the maximum gain is known as main lobe, and the sidelobe is defined as the local maxima with energy level lower than the main lobe. However, excessive side lobes will waste energy while the total energy of the output beamforming signal is fixed.

Beamforming algorithms could be broken down into two categories: fixed beamforming and adaptive beamforming. A fixed beamformer computes its weights without considering any information from input signals; an example is the delay-and-sum beamformer. Adaptive beamforming is always signal-dependent and needs to adaptively adjust the beamformer’s coefficients according to the input signals, in order to achieve the best performance.

There is a variety of microphone array arrangements, such as linear arrays, planar arrays and circular arrays, as depicted in Fig. 3.1. The direction of arrival (DOA) of a sound source is usually represented by two parameters: the elevation angle ($\theta$) and the azimuth angle ($\phi$). The circular array has an advantage over linear array in distinguishing whether the signal
comes from the front or the back of the array. In [33], it was shown that a circular array can achieve higher speech intelligibility in reverberant environments, compared to linear and planar arrays with the same number of microphones. In this chapter, we will focus on the circular microphone array, and assume all the microphone sensors are uniformly distributed around the circle and omnidirectional.

When considering the incoming radiation from a source located in any direction, we assume that the source is located in the array’s far field [6, 41], in which case the impinging signals arrive as plane waves from the source, as shown in Fig. 3.2. In this case, the distance between the source and the microphone array can be regarded as infinite. Thus, only the direction of the source is needed to determine the target position for the beamformer. But in reality, especially for low-frequency signals, the distance between the sound source and array does not meet the far-field requirement.

This chapter is organized as follows. First, a literature review on the principles of delay-and-sum (DAS), generalized sidelobe canceller (GSC) and minimum variance distortionless response (MVDR) beamformers is given. Then, inspired by the conventional MVDR, a new beamformer based on MVDR is proposed. Later, we illustrate 3-dimensional radiation
patterns for different beamformers, followed by a description of beamforming performance simulation and corresponding experimental results.

3.2 Literature Review

3.2.1 Delay-and-Sum (DAS) Beamformer

Delay-and-Sum (DAS) is the most common beamforming algorithm in acoustic signal processing. Signals captured by microphone arrays show a similar waveform at each microphone sensor. However, different time delays appear at different microphones. The delays are determined by the distance between sound source and microphones, as well as the sound propagation speed. DAS beamformer re-arranges the signals of all microphones according to the time delays. The re-aligned signals will be summed up and normalized by the number of microphones.

As shown in the right part of Fig. 3.1, there are \( N \) microphones located on the perimeter of the circle. The position of the \( n^{th} \) microphone in rectangular coordinates is described by

\[
(R \cos(\phi_n), R \sin(\phi_n), 0),
\]

where \( R \) is the radius of the circular array. The angular position of the \( n^{th} \) microphone is given by

\[
\phi_n = \frac{2\pi n}{N}, n = 1, ..., N.
\]

A sound source is marked as point \( P \) in the space, which can be represented by three parameters: elevation angle \( \theta \), azimuth angle \( \phi \) and radial distance \( r \). Mathematically, the array factor of the DAS beamformer in frequency domain is expressed as

\[
AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\kappa R[\sin \theta \cos(\phi - \phi_n)]},
\]

where \( w_n \) are the weights, and the exponential parts are the steering parameters between the sound source and each microphone. In the above equation, \( \kappa \) is the wave number which can be derived from

\[
\kappa = \frac{2\pi}{\lambda} = \frac{2\pi v}{c},
\]

where \( v \) is the frequency of the signal, \( \lambda \) is the wavelength and we assume the speed of sound in the air \( (c) \) is 343 m/s at 23\(^\circ\)C. A more detailed derivation can be found in [21].

3.2.2 Generalized Sidelobe Canceller (GSC)

Usually a conventional beamformer, e.g. Delay-and-Sum, not only emphasizes the signals from the desired direction, but amplifies the noise in that direction and some of the signals
in the side lobes. As a consequence, there arises a need for better beamformers, which suppress the side lobes more while keeping the total output energy unchanged.

The generalized sidelobe canceller (GSC) is an unconstrained adaptive beamformer, first presented by Griffith et al. [28]. This is an alternative but more efficient method of the linear constraint minimum variance (LCMV) beamformer [41].

The GSC beamformer usually consists of two parts, as shown in Fig. 3.3. The upper path of GSC mainly allows the signals in the desired direction to pass through using a conventional beamformer; meanwhile, the lower path filters out the desired signal and looks for the interfering signals plus noise in all other directions. The filter weights in the lower path are chosen adaptively using the least mean squares (LMS) optimization algorithm. Finally, the unwanted signals picked up by the lower path will be subtracted from the output of the upper path. The main purpose of GSC is to minimize the output power of noise and interference at the sidelobe canceller and avoid unintentional cancellation of the desired signals at the same time. In this section, we will illustrate the algorithm of GSC in detail.

The multi-channel input signal $X$ ($N \times K$ matrix) from the microphone array will be passed into both upper and lower paths. The microphone array consists of $N$ microphone sensors, and $X$ can be represented as $[x_1, x_2, ..., x_N]^T$, while $K$ is the total number of samples in the signal. In the upper path, $X$ is filtered by a conventional beamformer with a set of fixed weights, $w_q$. In our work, the simplest Delay-and-Sum is implemented as the fixed beamformer, and the output is denoted as $y_q$.

$$y_q = w_q^H X,$$

where $H$ is the Hermitian transpose. $X$ is also fed into a fixed matrix $B$, which filters out the signal arriving from the wanted direction and yields the signal containing interference and noise. This is why we refer to $B$ as the ‘blocking matrix’. We will discuss the design of
the blocking matrix \( B \) later. \( Y_b \) is the output of the blocking matrix.

\[
Y_b = BX
\]  

(3.6)

In our case, \( B \) is a \((N - 1) \times N\) matrix and the dimension of matrix \( Y_b \) is \((N - 1) \times K\). This signal \( Y_b \) is again filtered by an adaptive noise canceller with weights \( w_a \) to obtain \( y_a \). Then the system can obtain the desired output \( y \) by subtracting the interference signal \( y_a \) from \( y_q \).

\[
y_a = w^H_a Y_b
\]  

(3.7)

\[
y = y_q - y_a = w^H_q X - w^H_a B X = w^H X,
\]  

where the overall weight vector for the whole system is given by

\[
w = w_q - B^H w_a
\]  

(3.9)

The weight vector \( w \) can be computed by minimizing the total output power and not subject to any constraints. \( C_{xx} \) is the covariance matrix of the input signals.

\[
\begin{align*}
\text{minimize } w_a & \quad E \left[ \| y \|^2 \right] \\
\text{minimize } w_a & \quad E \left[ \| w^H X \|^2 \right] \\
\text{minimize } w_a & \quad \left( w_q - B^H w_a \right)^H C_{xx} \left( w_q - B^H w_a \right)
\end{align*}
\]  

(3.10)

Designing the filter \( w_a \) is an iterative procedure since \( w_a \) of the lower part of the beamformer should be updated adaptively. After applying the LMS algorithm, the weights \( w_a^n(k+1) \) at index \( k + 1 \) for the \( n^{th} \) microphone in the adaptive beamformer are updated according to the current filter coefficients \( w_a^n(k) \), blocking matrix output and current beamforming signal output, which is represented in Eq. 3.11, with a customized step size \( \mu \).

\[
w_a^n(k+1) = w_a^n(k) + \mu y(k) Y_b(n,k), k \in [1, 2, ..., K - 1], n \in [1, 2, ..., N - 1],
\]  

(3.11)

The relationship between \( w_a \) and \( w_q \) can be derived from the Eq. 3.10 as well.

\[
\frac{\partial}{\partial w_a} \left( w_q - B^H w_a \right)^H C_{xx} \left( w_q - B^H w_a \right) = 0
\]

\[
B C_{xx} \left( w_q - B^H w_a \right) = 0
\]

(3.12)

\[
w_a = \left( B C_{xx} B^H \right)^{-1} B C_{xx} w_q
\]

Therefore, based on the chosen \( B \) and \( w_q \), an optimal set of \( w_a \) can be determined.
As for the blocking matrix, the cascaded columns of the differencing (CCD) [34] is widely used. This matrix calculates the difference between adjacent columns. Specifically, $B$ is an $(N - 1) \times N$ matrix.

$$B = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & -1 \end{bmatrix}$$

(3.13)

### 3.2.3 Minimum Variance Distortionless Response (MVDR) Beamformer

MVDR beamforming, proposed by Capton in 1969 [12], is an iteratively adjusted method to estimate the beamformer weights, by minimizing the total output energy in an array system while guaranteeing the desired signal has unit gain. Here we present the derivation of this method.

For an $N$-element microphone array environment with $M$ speakers, we consider the received signals comprise one desired signal, $M$-1 interference signals and ambient noise,

$$X = AS + n$$

$$= A \begin{bmatrix} s_0^T \\ s_1^T \\ \vdots \\ s_{M-1}^T \end{bmatrix} + n$$

(3.14)

$A$ is the $N$-by-$M$ steering matrix and $S$ is the associated $M$-by-$K$ input signal, where $K$ is the total number of samples of each signal; and $n$ represents the environmental noise. $S$ can be further decomposed into desired signal $s_0$ plus the unwanted signal $[s_1^T, s_2^T, \ldots, s_{M-1}^T]^T$. Mathematically, the beamformer output can be written as the multiplication of the chosen weights and input signal,

$$y = w^H X$$

(3.15)

In this case $w = [w_1, w_2, \ldots, w_N]^T$ is the weight vector corresponding to $N$ microphones. The optimal weights can be achieved by maximizing the signal-to-interference-plus-noise ratio (SINR) of the beamforming output, which is equivalent to minimizing the total weighted power output:

$$\text{SINR} = \frac{P_{\text{signal}}}{P_{\text{noise+interference}}} = \frac{w^H C_{ss} w}{w^H C_{nn} w},$$

(3.16)

where $C_{ss}$ and $C_{nn}$ are the covariance matrices of the desired signal and noise-plus-interference, respectively. In theory, if we assume the desired signal and noise-plus-interference are uncorrelated, then we have $C_{xx} = C_{ss} + C_{nn}$. Since MVDR sets a constraint to prevent the gain in the direction of target signal from being suppressed, the numerator of Eq. 3.16 will be a
fixed number. Mathematically, minimizing the denominator \( w^H C_{nn} w \) is equivalent to maximizing SINR. However, noise is hard to separate from the signal in practice, so we replace \( C_{nn} \) by \( C_{xx} \). In other words, minimizing the total array output power with a constraint that the gain towards target direction be 1 can also give an optimal weight vector.

\[
\begin{align*}
\mathbf{w}_{opt} &= \min_{\mathbf{w}} E \left[ \| \mathbf{y} \|^2 \right] \\
&= \min_{\mathbf{w}} E \left[ \| \mathbf{w}^H \mathbf{X} \|^2 \right] \\
&= \min_{\mathbf{w}} \mathbf{w}^H E \left[ \mathbf{X} \mathbf{X}^H \right] \mathbf{w} \\
&= \min_{\mathbf{w}} \mathbf{w}^H \mathbf{C}_{xx} \mathbf{w} \tag{3.17}
\end{align*}
\]

subject to \( \mathbf{w}^H \mathbf{h}_0 = c_0 \),

where \( \mathbf{h}_0 \) is called the steering vector of the microphone array towards the specific direction, for example, \( \mathbf{h}_0 \) has \( N \) elements, and each one is the steering number corresponds to one microphone. \( c_0 \) is usually set to 1.

Lagrange multiplier strategy is applied here to solve the given constrained optimization problem. In the following equation, \( \mathbf{C}_{xx} \) refers to the same covariance matrix of the input signal as in Eq. 3.17.

\[
\begin{align*}
\mathcal{L}(\mathbf{w}, \lambda) &= \mathbf{w}^H \mathbf{C}_{xx} \mathbf{w} + \lambda_0 (\mathbf{w}^H \mathbf{h}_0 - c_0) \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^H} &= \mathbf{C}_{xx} \mathbf{w} + \lambda_0 \mathbf{h}_0 = 0 \\
\frac{\partial \mathcal{L}}{\partial \lambda_0} &= \mathbf{w}^H \mathbf{h}_0 - c_0 = 0 \tag{3.19}
\end{align*}
\]

The optimized weights are then found from Eq. 3.19:

\[
\mathbf{w} = c_0 \frac{\mathbf{C}_{xx}^{-1} \mathbf{h}_0}{\mathbf{h}_0^H \mathbf{C}_{xx}^{-1} \mathbf{h}_0} \tag{3.20}
\]

A regularization method called diagonal loading has been performed in order to enhance the invertibility of the covariance matrix [13, 42, 55]. Particularly, increase the diagonal element of the covariance matrix by the parameter \( \mu \),

\[
\tilde{\mathbf{C}} = \mathbf{C}_{xx} + \mu \mathbf{I}, \tag{3.21}
\]

where \( \mathbf{I} \) is the identity matrix and \( \mu \) is a small loading constant. Thus, in practice \( \tilde{\mathbf{C}} \) is substituted in Eq. 3.20 instead of \( \mathbf{C}_{xx} \).

We implemented this method in Matlab, and the major part of MVDR was processed in the frequency domain [14, 31]. We will illustrate the details about our implementation.
method here, and the pseudocode is shown below. The elevation angle $\theta$ and the azimuth angle $\phi$ is required at the input to define the target direction. The steering vector $h_0$ could also be pre-calculated by

$$h_0 = \begin{bmatrix}
e^{-j\kappa R \sin \theta \cos(\phi - \phi_1)} \\
e^{-j\kappa R \sin \theta \cos(\phi - \phi_2)} \\
\vdots \\
e^{-j\kappa R \sin \theta \cos(\phi - \phi_N)}
\end{bmatrix}, \quad (3.22)$$

similar to Eq. 3.3. The beamformer input matrix $X$ consists of the signals captured from a circular microphone array of $N$ sensors.

A short-time Fourier transform (STFT) is applied to convert input signals $X$ from time domain into frequency domain $X_f = [X_1^f, X_2^f, ..., X_N^f]^T$. Specifically, STFT divides a single-channel signal into short frames of equal length, and then applies Fourier transform to each frame. We adopt a fast Fourier transform (FFT) for a faster implementation. The covariance matrix of the noisy input signal is then calculated, the details are shown in Algorithm 1, lines 6 to 8. The input signals $X_f$ in the frequency domain are organized as $L$ frames, and divided into $I$ frequency bins in each frame using FFT. We extract all these frequency bins from every frame separately, named as $U_i$. $C_{i,l}$ represents the local covariance matrix of the $i^{th}$ frequency bin at $l^{th}$ frame. After normalizing $C_{i,l}$ by its trace, the estimated global covariance matrix is computed by averaging the local covariance matrix from each frame. The final estimate of the covariance matrix $\hat{C}$ is obtained using diagonal loading. Each frequency bin represents one frequency in the spectrum, which is related to $\kappa_i$ in the line 10, and the steering vector $h_0$ at the corresponding frequency is calculated. Next, the calculated weights for each microphone at each frequency bin, according to Eq. 3.20, are multiplied with input signals in the frequency domain. The inverse STFT helps to obtain the beamformed signals $\tilde{x}$ in time domain.

### 3.3 MVDR with Two Constraints (MVDR-2C)

Based on the conventional MVDR, we proposed an improved version of MVDR with an additional constraint. This proposed method is aimed at maximizing the power ratio between one desired signal and one specific interference. Instead of only listening to the desired signal and suppressing all the surrounding sound, the two constraints MVDR focuses on two given directions. Specifically, MVDR-2C works to suppress the beam response at the selected interference direction, while keeping the desired signal with unit gain. Consider the same condition as described in Section 3.2.3: we have $N$ microphone signals, $M$ sound
**Algorithm 1: Conventional MVDR**

```
Input: θ, φ
Input: N signals: X = [x_1, x_2, ..., x_N]^T
1 c_0 = 1 // Set constraint to 1
/* Calculate steering vector for each microphone at different frequency */
2 X_f = STFT{X} // Has II frames in STFT
3 for i == 1:I do
  4 U_i : i^{th} frequency components from all frames // I represents the number of frequency bin
  5 for l == 1:L do
    /* Calculate covariance matrix across every single frame */
    6 C_{i,l} = U_i(l)U_i(l)^H
    7 C_{i,l} = C_{i,l} / tr(C_{i,l})
    8 globalC_i = 1/L sum_{l=1}^{L} C_{i,l} // Average across every FT frame
    9 \hat{C}_i = globalC_i + \mu I // Diagonal loading
    10 h_0^i = e^{j\kappa R[sinθ cos(φ - φ_n)]}
    11 w_i = c_0 \hat{C}_i^{-1} h_0^i (\hat{C}_i^{-1} h_0^i)^H
    12 w = [w_1, w_2, ..., w_I]^H
    13 \hat{x} = ISTFT{w^H X_f}
```

Sources, the only difference we take an extra interference into account.

\[
X = \begin{bmatrix} 
  s_0^T \\
  s_1^T \\
  \vdots \\
  s_{M-1}^T 
\end{bmatrix} + n, \quad (3.23)
\]

where A is the N x M steering matrix; s_0 is the desired signal, s_1 is the considered interference, and [s_2^T, s_3^T, ..., s_{M-2}^T]^T consists of M - 2 interfering signals; n represents ambient noise. Eq. 3.17 need to be modified to satisfy the second constraints:

\[
w_{opt} = \min_w w^H C_{xx} w \\
\text{subject to } w^H h_0 = c_0, \quad (3.24) \\
w^H h_1 = c_1,
\]

where h_0 and h_1 are the steering vectors for the desired signal and targeted interference, respectively. c_0 and c_1 are the associated constraint values. According to the conventional MVDR, c_0, the gain for the desired signal, it usually set to 1. c_1 is the gain for the interference in this problem. In other words, the ratio of \( c_0 / c_1 \) decides the difference between the target
and the considered interference. We define the target-to-interference constraint ratio $\frac{0}{c_1}$ to be $C_r$.

Lagrange multiplier method was again applied here to solve this optimization problem.

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^H \mathbf{C}_{xx} \mathbf{w} + \lambda_0 (\mathbf{w}^H \mathbf{h}_0 - c_0) + \lambda_1 (\mathbf{w}^H \mathbf{h}_1 - c_1)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{C}_{xx} \mathbf{w} + \lambda_0 \mathbf{h}_0 + \lambda_1 \mathbf{h}_1 = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_0} = \mathbf{w}^H \mathbf{h}_0 - c_0 = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = \mathbf{w}^H \mathbf{h}_1 - c_1 = 0$$

The above equations can be solved as below,

$$\lambda_0 = \frac{c_0 a_{11} - c_1 a_{10}}{a_{10} a_{01} - a_{11} a_{00}}, \quad \lambda_1 = \frac{c_1 a_{00} - c_0 a_{01}}{a_{10} a_{01} - a_{11} a_{00}},$$

$$\mathbf{w} = -\lambda_0 \mathbf{C}_{xx}^{-1} \mathbf{h}_0 - \lambda_1 \mathbf{C}_{xx}^{-1} \mathbf{h}_1,$$  

(3.26)

where $a_{00} = h_0^H \mathbf{C}_{xx}^{-1} \mathbf{h}_0, a_{01} = h_0^H \mathbf{C}_{xx}^{-1} \mathbf{h}_1, a_{10} = h_1^H \mathbf{C}_{xx}^{-1} \mathbf{h}_0, a_{11} = h_1^H \mathbf{C}_{xx}^{-1} \mathbf{h}_1$.

### 3.4 Simulation of a Circular Array Beam Pattern

We implemented the 3D simulation code for DAS in Matlab. In order to create 3D radiation patterns, we made some modifications to the array factor from Eq. 3.3.

$$\text{DAS}_\text{AF}(\theta, \phi) = \sum_{n=1}^{N} w_n e^{jR[\sin \phi \cos(\phi_n) - \sin \theta_0 \cos(\phi_n)]},$$  

(3.27)

where $(\theta, \phi)$ is the random angle scattered in the entire 3D space, and $(\theta_0, \phi_0)$ is the direction of the target with maximum radiation. $R$ is the radius of the microphone array and $\kappa$ is the wave number as defined in Eq. 3.4. For 3D simulation, we assume all the microphones are omnidirectional and the weights $w_n$ are the same among all the microphones, and are set to be 1.

The 3D patterns are obtained based on the following setup: the radius of the circular microphone array is 5.5 cm, and the target direction is set to $(90^\circ, 135^\circ)$, which is equivalent to $(-\frac{\sqrt{2}}{2} x, \frac{\sqrt{2}}{2} y, 0)$ in the Cartesian coordinates. The sampling rate of input speech signals
is assumed as 16k samples/second. Thus, we simulated the 3D beam patterns at different frequencies, starting from 400Hz to 8kHz, according to the Nyquist theorem. The power pattern is then converted into the logarithmic domain by the following equation:

$$\text{Radiation pattern (in dB)} = 10 \log_{10} |AF|^2,$$

(3.28)

where $AF$ could be $DAS\_AF$ from Eq. 3.27 or $MVDR\_AF$ from Eq. 3.29. Fig. 3.5 below provides the DAS radiation responses at selected frequencies 400Hz, 1400Hz, 2000Hz, 4400Hz, 5600Hz and 8000Hz. Fig. 3.6 illustrates the comparison of different beamforming patterns at the corresponding frequencies in units of dB. We scaled the pattern to the range from 0 dB down to -40 dB, and the colormap was scaled to $[-40, 0]$ dB as well. In both figures, we use the jet colormap (see Fig. 3.4) to define the strength of the radiation in a specific direction. The rectangular coordinate system $(x,y,z)$ is adopted, and every axis is normalized to $[-1, 1]$. The intersection of the dotted lines represents the origin of the coordinate system, and the blue dot at $(-\sqrt{2}/2, \sqrt{2}/2, 0)$ on each sub-image indicates one point in the target direction, where the maximum radiation should occur. The red dot that appeared in the opposite direction represents the interference direction that will be explicitly suppressed in MVDR-2C.

The power plot of the DAS beamforming pattern in linear scale (Fig. 3.5) shows the array pattern at different frequencies. Each point in the graph has a reference in the colormap, which helps to distinguish the level of pattern intensity. The radial distance from the origin also corresponds to the strength of the pattern in a specific direction. As shown in the figure, the radiation patterns at all the frequencies have the largest lobe towards the target $(-\sqrt{2}/2, \sqrt{2}/2, 0)$ and reach the maximum value at that point, whose color is maroon in the colormap. The lobe oriented in the desired direction is usually called the ‘main lobe’. From the plot, the radiation pattern behaves less directional at low frequencies, e.g. 400Hz. In Fig. 3.5b, the color at the origin shown as navy blue indicates that it has relatively weaker pattern at the origin compared to the target direction. As frequency increases, side lobes arise in some unwanted directions and are separated by several nulls. At 2000Hz, a ‘back lobe’ occurs away from the target direction. This plot supports the theoretical predictions that the low-frequency pattern tends to be omnidirectional, whereas the array directivity gets stronger as frequency increases.

We also implemented the 3-dimensional radiation pattern for the conventional MVDR beamformer, whose weights can be obtained through Algorithm 1. At any specific frequency, 8 weights are returned to steer the array beam. MVDR weights are not generic but input-dependent, since the weights are computed from the steering vector of the target direction and the covariance matrix of the input signals. In other words, different inputs will end up with a different set of weights. Below is the equation for the conventional MVDR 3D
Figure 3.5: 3D radiation patterns generated by Delay-and-Sum beamformer at frequency 400Hz, 1400Hz, 2000Hz, 4400Hz, 5600Hz and 8000Hz. The blue dot at \((-\sqrt{2}/2, \sqrt{2}/2, 0)\) represents the desired sound source with distance to the origin equals to 1, and the line connected the origin and the red dot at \((\sqrt{2}/2, \sqrt{2}/2, 0)\) shows the direction of interference.
pattern:

\[ MVDR\_AF(\theta, \phi) = \sum_{n=1}^{N} w_n e^{j\kappa R \left[ \sin \theta \cos(\phi - \phi_n) \right]} , \]  

(3.29)

where \((\theta, \phi)\) is the angle distributed over the entire space, \(R\) is the radius of microphone array and \(\kappa\) is the wave number. Other than Eq. 3.27, where \(w_n\) are the amplitude weights without phase shift, the steering vector of MVDR is already included in weights \(w_n\) in conventional MVDR, as shown in Eq. 3.20. Accordingly, Eq. 3.29 does not include a redundant steering vector in the exponential part. For the same reason, the radiation response for MVDR-2C can be implemented using Eq. 3.29 with different weights \(w_n\) calculated by Eq. 3.26.

Fig. 3.6 demonstrates the simulated array patterns of different beamformers in the units of decibel. We compared the behavior of DAS, MVDR and the proposed two-constraints MVDR, at frequencies of 400Hz, 1400Hz, 2000Hz, 4400Hz and 5600Hz. From the figure, the array patterns of all three beamformers are almost equivalent in all directions when frequency is 400Hz. At the same time, the side lobes arise naturally with the growth of frequency. Starting from 1400Hz, MVDR-2C shows a sharp ‘null’ response towards the direction of interference. On the other hand, DAS and MVDR patterns do not exhibit such strong attenuation in the direction of interference.

To further examine whether MVDR-2C maintains the desired signal with maximum gain, while providing better suppression towards interference, the energy differences between target and interference were computed, based on Fig. 3.6. The signal-to-interference ratio (SIR) in Fig. 3.7 is calculated by subtracting the radiation gain (in dB) at interference direction from the gain towards target. This plot illustrates SIR for DAS, MVDR and MVDR-2C within the frequency range of \([400, 8000]\) Hz. From the figure, MVDR-2C has the highest SIR value at most frequencies, and has an overwhelming advantage over DAS and MVDR, especially when the frequency is between 3000Hz and 6000Hz. A particular case occurs at 1200Hz where DAS outperforms both MVDR and MVDR-2C. The area under the curve computed by trapezoidal numerical integration indicates that MVDR-2C performs better interference attenuation subject to the same target signal than DAS and MVDR.

3.5 Beamformer Evaluation

The beamforming performance is influenced by multiple factors, for instance, the geometry of the microphone array, room reverberation and the beamforming algorithm itself. There are numerous methods to assess the correctness and robustness of different beamforming systems. Here we apply four most commonly used objective measurements of speech quality to compare different beamformers.
Figure 3.6: Comparison of 3D power patterns between DAS, MVDR and MVDR-2C at frequency 400Hz, 1400Hz, 2000Hz, 4400Hz and 5600Hz are shown in dB.
3.5.1 Experiment Setup

Controllable volume and signal power of the input speech helps to reduce uncertainty in the experiment. Therefore, we decided to use pre-recorded speech instead of real-time speech signals. We selected signals from the WSJ0 corpus [25], which consists of read speech from Wall Street Journal news articles. The selected signals were treated as raw input audio samples and passed into an acoustics room simulator called pyroomacoustics [50]. Pyroomacoustics is an open-source package aimed at testing microphone array processing algorithms with easily generated simulation scenarios. It takes various parameters into account, i.e., room size, sound source location, microphone array’s location and arrangement, and room reverberation coefficient. Based on the image source model (ISM) [2] and the specified room configuration, the room impulse response (RIR) [2] is simulated and convolved with the raw audio input. In our case, as shown in Fig. 3.8, we simulated raw input audio in both free space environment and reverberant environment within a 5 × 5 × 5 meters 3D cubic room. The simulation with reverberation considers the absorption factor of 0.2 for the walls. An 8-microphone circular array with a radius of 0.055 m is placed in the center of this virtual room, and 1 meter above the floor. The sound source is always 1 meter away from the array center and 0.6 meters higher than the array plane. This setup attempts to mimic a speaker sitting at a desk in front of the microphone array.

We only considered one sound source per simulation. For the scenario of multiple speakers, i.e., one target and one interference, we simulated them separately and added them together to create the mixture signal.
Figure 3.8: Left is the simulation setup of virtual acoustic environment. Eight black dots represent microphones which uniformly distributed among a circular array with radius of 0.055 meters. The coordinate of the array center is $[2.5, 2.5, 1]$, with the unit in meter. Right figure gives the preview for the locations of target/interference relative towards the array center.

We also tested the relationship between beamformer performance and several variables with controlled parameters. In order to assess whether the angle ($\alpha$ shown in Fig. 3.8) between the target and interference sound source will influence the beamformer performance, we rotate the interference sound source position clockwisely relative to the center of microphone array; meanwhile, the position of target source is never changed, as shown in Fig. 3.8. In total, we take 23 different angle positions into account, from $15^\circ$ to $360^\circ$ in steps of $15^\circ$.

3.5.2 Evaluation Metrics

Various metrics are used to evaluate the performance of four beamformers: DAS, GSC, MVDR and MVDR-2C. Signal-to-distortion ratio (SDR) and Signal-to-interference-plus-noise ratio (SINR) are common methods to show the level difference between the desired signal and the other unwanted signal gained from beamformers. Speech quality and intelligibility are usually measured by PESQ [47] and STOI [53] metrics. Before evaluating, the simulated audio signals are beamformers’ input, and the beamformed output signals are time-aligned with the input signals to compute the metrics.

From [48, 53], the signal-to-distortion ratio (SDR) in logarithmic domain is defined by Eq. 3.30-3.31:

$$\gamma = \sqrt{\frac{\|s_0\|^2}{\|y\|^2}},$$  \hspace{1cm} (3.30)

$$SDR = 10 \log_{10} \left( \frac{\|s_0\|^2}{\|\gamma y - s_0\|^2} \right),$$ \hspace{1cm} (3.31)
where $s_0$ is the clean desired signal, and $y$ is the processed beamformer output. $y$ is scaled by the scaling factor $\gamma$ when calculating SDR, in order to equalize the energy level of clean and beamformed signals. This is known as scale-invariant SDR.

The beamformed signal $y$ is a mixture of wanted signal $s_0$, interference $i$, and background noise $n$. $i$ is a column vector derived from $\sum_{j=1}^{M-1} s_j$, where $s_j$ is the $j^{th}$ interfering component shown in Eq. 3.14. The signal-to-interference-plus-noise ratio (SINR) can be treated as

$$SINR = 10 \log_{10} \left( \frac{\|w \odot s_0\|^2}{\|w \odot i + w \odot n\|^2} \right),$$

(3.32)

$w$ is the weighted vector obtained from beamformers, which is input-dependent. As mentioned in Section 3.5.1, the target and interference speech is simulated separately by py-roomacoustics, it can be assumed that target, interference and noise are uncorrelated with each other.

For testing speech quality and intelligibility, we adopt Perceptual Evaluation of Speech Quality (PESQ) [47] and short-time objective intelligibility (STOI) [53] assessment model. Both metrics are well-developed and available online\(^1\).\(^2\). The PESQ subjective scores range from -0.5 (bad) to 4.5 (no distortion), and STOI ranges from 0 to 1, the higher the better.

### 3.5.3 Results and Analysis

**Free Space**

We evaluated the different beamformers with multiple variables controlled in a free space, where the acoustic signals are propagated from the source to infinity without reflection and energy absorption. First of all, in order to assess overall beamforming performance, an extreme case is taken into account: female and male speech sources are involved and located in the opposite directions; both of them are 1 meter away from the microphone array’s center. The female audio source is closest to microphone 3 (Mic3) while the male audio is closest to microphone 7 (Mic7), as shown in Fig. 3.9.

For result consistency, we chose 0.1 to be the diagonal loading parameter ($\mu$) for both conventional MVDR and proposed two-constraint MVDR, and the target-to-interference constraint ratio ($C_r$) for the proposed MVDR-2C is set to be infinite. Table 3.1 represents the performance difference among four beamformers. From the table, MVDR and MVDR-2C generally outperform both DAS and GSC methods, whereas the result of two MVDR versions does not differ too much under these conditions.

Another set of experiments was conducted to evaluate the influence of diagonal loading ($\mu$) and target-to-interference constraint ratio ($C_r$) on two MVDR-based methods. As

\(^1\)https://github.com/dennisguse/ITU-T_pesq

\(^2\)https://www.ceestaal.nl/code/
Figure 3.9: Experiment setup. Number 1 - 8 represents the location of eight microphones in our system coordinate. Female and male signals are one meter away from the coordinate origin. Two speeches are places in opposite direction (180° difference).

<table>
<thead>
<tr>
<th></th>
<th>SDR (dB)</th>
<th>SINR (dB)</th>
<th>STOI</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAS</td>
<td>1.487</td>
<td>0.162</td>
<td>0.746</td>
<td>2.052</td>
</tr>
<tr>
<td>GSC</td>
<td>1.163</td>
<td>0.364</td>
<td>0.732</td>
<td>1.955</td>
</tr>
<tr>
<td>MVDR</td>
<td>4.990</td>
<td>4.971</td>
<td>0.841</td>
<td>2.390</td>
</tr>
<tr>
<td>MVDR-2C</td>
<td>9.744</td>
<td>13.291</td>
<td>0.913</td>
<td>2.900</td>
</tr>
</tbody>
</table>

Table 3.1: Result of DAS, GSC, MVDR with 0.1 diagonal element value and MVDR-2C with 0.1 diagonal element and infinite target-to-interference constraint ratio, the angle $\alpha$ between target and interference is 180°.

summarized in Table 3.2, conventional MVDR gives the highest scores on all four metrics with $\mu = 0.001$. The two-constraint MVDR has the best performance with $\mu = 0.1$, even better than MVDR. When $\mu = 0.01$, although MVDR-2C does not deliver its best result, the performance is still acceptable. These conclusions hold for all values of the angle ($\alpha$) between target and interference.

The performance results with respect to the constraint number $C_r$ of MVDR-2C are reported in Table 3.3. The experiment still follows the same setting as described above: female speech source located close to Mic3 is chosen as the target, while male sound source near Mic7 is interference. The ‘constraint ($C_r$)’ in the table describes the target-to-interference constraint ratio. The target and interference are exchanged when $C_r < 0$, which implies that the target in this experiment is changed to male. Constraint equal to 1 means we treat both target and interference the same; in other words, there is no extra suppression towards interference. From the top half of the table (MVDR-2C with $\mu = 0.1$), we can conclude that the higher the constraint ratio, the better the suppression towards interference. Surprisingly, for $\mu = 0.01$, the highest scores are generated when $C_r$ is 100 instead of infinite, although the SDR and SINR between those two cases only differ in approximately 0.1 dB. Again, the result of MVDR-2C with infinite constraint and $\mu = 0.1$ only has 0.1 dB gain over 100 constraint ratio in both SDR and SINR compared with the case when the constraint is 1,
<table>
<thead>
<tr>
<th>$MVDR$</th>
<th>Diagonal loading $\mu$</th>
<th>SDR (dB)</th>
<th>SINR (dB)</th>
<th>STOI</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0001</td>
<td>6.242</td>
<td>8.723</td>
<td>0.853</td>
<td>2.643</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td><strong>6.568</strong></td>
<td><strong>8.948</strong></td>
<td><strong>0.869</strong></td>
<td><strong>2.680</strong></td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>6.124</td>
<td>7.085</td>
<td>0.852</td>
<td>2.531</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>3.634</td>
<td>2.813</td>
<td>0.795</td>
<td>2.211</td>
</tr>
<tr>
<td>$MVDR-2C$</td>
<td>Diagonal loading $\mu$</td>
<td>SDR (dB)</td>
<td>SINR (dB)</td>
<td>STOI</td>
<td>PESQ</td>
</tr>
<tr>
<td></td>
<td>0.0001</td>
<td>1.686</td>
<td>1.505</td>
<td>0.714</td>
<td>2.063</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>4.330</td>
<td>4.979</td>
<td>0.784</td>
<td>2.354</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>7.455</td>
<td>9.452</td>
<td>0.868</td>
<td>2.690</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td><strong>8.920</strong></td>
<td><strong>12.341</strong></td>
<td><strong>0.910</strong></td>
<td><strong>2.889</strong></td>
</tr>
</tbody>
</table>

Table 3.2: Vary diagonal loading value for conventional MVDR and MVDR-2C with infinite $C_r$ constraint.

<table>
<thead>
<tr>
<th>$MVDR-2C$, $\mu = 0.1$</th>
<th>Constraint $C_r$</th>
<th>SDR (dB)</th>
<th>SINR (dB)</th>
<th>STOI</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
<td>-2.580</td>
<td>-17.171</td>
<td>0.328</td>
<td>0.924</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.237</td>
<td>-1.315</td>
<td>0.645</td>
<td>1.753</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8.494</td>
<td>11.156</td>
<td>0.883</td>
<td>2.754</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>8.910</td>
<td>12.297</td>
<td>0.909</td>
<td>2.883</td>
</tr>
<tr>
<td></td>
<td>Inf</td>
<td><strong>8.920</strong></td>
<td><strong>12.341</strong></td>
<td><strong>0.910</strong></td>
<td><strong>2.889</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$MVDR-2C$, $\mu = 0.01$</th>
<th>Constraint $C_r$</th>
<th>SDR (dB)</th>
<th>SINR (dB)</th>
<th>STOI</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
<td>-2.594</td>
<td>-14.527</td>
<td>0.372</td>
<td>0.992</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.863</td>
<td>-1.637</td>
<td>0.635</td>
<td>1.737</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>7.427</td>
<td>9.358</td>
<td>0.855</td>
<td>2.647</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td><strong>7.479</strong></td>
<td><strong>9.499</strong></td>
<td><strong>0.868</strong></td>
<td><strong>2.690</strong></td>
</tr>
<tr>
<td></td>
<td>Inf</td>
<td>7.455</td>
<td>9.452</td>
<td>0.868</td>
<td>2.690</td>
</tr>
</tbody>
</table>

Table 3.3: Result of different constraint numbers of MVDR-2C, while diagonal loading values are $0.1$ (top table) and $0.01$ (bottom table), and female is treated as the primary target.
the output from infinite constraint version has a 7.7 dB improvement in SDR and 13.6 dB gain in SINR. Furthermore, \( C_r = 0.1 \) forces the beamformer to steer its beam towards the male, but the four assessment methods still consider the clean female speech as the desired target. This is the reason that SINR is -17.2 dB and -14.5 dB when \( C_r \) is 0.1 in both \( \mu = 0.1 \) and \( \mu = 0.01 \) cases. The above comparison shows that our proposed two-constraint MVDR can improve the beamforming output performance with a specific diagonal loading value and large \( C_r \) constraint.

The location and distance between the target and interfering sources is another factor that affects beamforming performance. As mentioned in Section 3.5.1, the target position is fixed, and the angle difference \( \alpha \) between target and interference is increased from 15° up to 360° in steps of 15°. Fig. 3.10 provides all the results with different speech-source settings. As depicted in the figure, two MVDR-based methods give better overall performance than DAS and GSC beamformers. The yellow and green lines represent the behavior of conventional MVDR with 0.1 and 0.01 diagonal loading values, respectively. These two curves are symmetric with respect to the vertical line at 180° angle difference, and the best performance is achieved when the target and interference are located in opposite directions. In the meantime, the curves in purple and azure illustrate the results for two-constraint MVDR when diagonal loading parameter is 0.1 and 0.01. From the figure, the MVDR-2C always reaches a local maximaum at 45°, 150°, 210° and 315°, and has a performance drop around 90°, 180° and 270°. This is due to the nature of MVDR-2C algorithm: it generates different beam patterns with different \( \alpha \). For example, Fig. 3.11 gives two patterns of the MVDR-2C when \( \alpha = 45° \) and \( \alpha = 90° \). The blue dot represents the direction of the target while the red dot gives the direction of the interfering signal. From the figure, it can be seen that the beam pattern for \( \alpha = 45° \) provides a sharper null in the direction of the interference than \( \alpha = 90° \). Generally speaking, in terms of SDR and SINR, the proposed two-constraint MVDR with 0.1 diagonal loading value provides a significant improvement over all the other compared beamformers.

**Reverberant Environment**

Beamforming in a reverberant environment is more challenging than in a free space environment. The objective measurement results of a set of signals simulated in a reverberant room are shown in Fig. 3.12. Compared with the objective evaluation in free space, the result with reverberation presents more fluctuations as the angle between the target and interfering signal changes. All the evaluation values with reverberation drop compared to the result obtained from a free space environment. For example, MVDR-2C with \( \alpha \) equals to 180° decreases 7 dB in SDR and decreases 10 dB in SINR. The overall trend for beamforming performance shows that MVDR-2C delivers the best behavior, the conventional MVDR has the second high score, and the behaviors of DAS and GSC are relatively weaker, in terms of SDR, intelligibility and speech quality, respectively. This can be mitigated by fur-
Figure 3.10: (a)-(d) show SDR, SINR, PESQ and STOI evaluated in a free space among different beamformers, respectively. The angle difference along x-axis represents the angle between target and interference relative to the center of microphone array. Results shown for MVDR and MVDR-2C here are associated with two diagonal loading values: 0.1 and 0.01. Infinite $C_r$ constraint of MVDR-2C is considered.
ther noise cancellation. Besides, the figure reports that the beamforming performance has a drop every 45°. For instance, all four beamformers will reach a local minimum for SDR when α is 135° and 180°.

The proposed two-constraint MVDR outperforms the other three beamformers in all aspects when α is in a range from 45° to 315°, whereas the MVDR-2C performance drops dramatically when α is out of this range. The signals simulated with reverberation consist of sound from the direct path and reflected paths, and the dominant reflected path might influence the beamforming performance. MVDR-2C is highly sensitive to the accuracy of the target and interfering signal direction, which will be discussed in Section 4.4.2.

3.6 Conclusions

In this chapter, algorithms for three different beamformers (DAS, GSC and MVDR) are discussed in detail. Inspired by the conventional MVDR, we propose a new MVDR-based beamformer with an extra constraint, which attempts to maximize the power ratio between the target and interfering signals.

The beam patterns of DAS, conventional MVDR and the proposed MVDR-2C for a circular array of 8 microphones are simulated. The presented beam responses indicate that MVDR and MVDR-2C generate higher energy beams towards the target compared to DAS, and MVDR-2C has sharper ‘nulls’ in the direction of a specific interference.

Four speech metrics are adopted to assess the performance of different beamformers in both free space and reverberant environments. The computer-simulated result shows both conventional MVDR and the proposed MVDR-2C outperform the DAS and GSC in terms of all four evaluation metrics. Besides, MVDR-2C also provide better performance result than the conventional MVDR with an appropriate choice of diagonal loading values. Although
Figure 3.12: Results for objective evaluation in a reverberant room. $\alpha$ represents the angle between the target and interference. The MVDR and MVDR-2C are tested when the diagonal loading parameters are set to 0.1, and $C_r$ for MVDR-2C is infinite.

the beamforming performance degrades with reverberation, the proposed MVDR-2C still achieves the best result among the four compared beamformers.
Chapter 4

Visually-Guided Beamforming

4.1 Introduction

Microphone arrays can be useful in many audio-related applications, such as sound source localization (SSL), beamforming, noise/echo reduction, cocktail party effect mitigation, and speech separation [6]. Using a microphone array gives more flexibility than working with a single-microphone system. With a single-microphone system mounted at a fixed place or hand-held, talkers may be restricted to a given physical region or orientation. Compared to the traditional single microphone, which is only able to offer a fixed/limited directionality, a microphone array provides the possibility to capture the sound from a desired direction. Besides, microphone arrays can also help to estimate the sound source location automatically, and emphasize the desired sound from the direction of interest while rejecting other interfering signals.

As mentioned in the previous chapter, beamforming is achieved by spatial filtering the signals obtained from a set of microphone sensors. Most of the recent studies in automatic speech recognition (ASR) use these steering methods to attenuate ambient noise and unwanted interference. The direction-of-arrival (DOA) of the desired signal is essential to beamformers. Speech source localization (SSL) [8] is widely used to determine the target angle for the beamforming algorithm. Microphone array has been embedded into many products, especially for teleconferencing. For example, Rally from Logitech estimates the location of the active speaker by SSL and voice activity detector, and then makes use of the knowledge of that direction to steer the camera and suppress background noise. In this case, the microphone array is used to zoom in the video portion where the active speaker is located while it does not need to provide a full HD video of the entire conferencing room. However, if microphones are too close to each other in a small sensor array, the accuracy of SSL is decreased [8][45].

In computer vision, various algorithms for face detection and tracking are already implemented in many real applications. In this chapter, we explore the possibility of using face detection to provide DOA estimates for beamforming.
Figure 4.1: MATRIX Creator front view (left) and back view (right) with 8 microphones: M1 to M8.

A conventional camera has a limited field-of-view. Choosing a 360° camera (usually two wide-angle fisheye sensors placed back-to-back) instead of multiple conventional cameras helps to capture images covering the whole 360° scene with the least number of vision sensors and avoids synchronization and multi-image alignment problems.

Faces detected by cameras (including the conventional cameras and 360° cameras) can be tracked without the help of SSL. The goal here is to exploit a 360° camera to guide the microphone array, so that the beamformer can choose its target direction towards the speaker’s face. Two pieces of hardware are used here: MATRIX Creator with a circular microphone array and Ricoh Theta V 360° camera.

In this chapter, we will discuss the equipment specification in detail. The system structure is also well demonstrated and explained here. Last but not least, the impact of the face localization error on beamformers is simulated and analyzed.

4.2 Equipment Specification

Our visually guided beamforming system is a combination of MATRIX Creator with a circular microphone array and Ricoh Theta V with dual-fisheye lenses. The following sections introduce these two devices and their specifications.

4.2.1 MATRIX Creator

MATRIX Creator, shown in Fig. 4.1, is a fully-developed hardware board equipped with an FPGA and multiple sensors, including gyroscope, light, humidity, temperature and microphones. The eight MEMS microphone sensors are equally spaced and located at the back of the MATRIX Creator, as depicted in the right part of Fig. 4.1. The microphones M1 to M8 form a circular array with a radius of 5.5 centimeters.
MATRIX Creator is designed to be compatible with Raspberry Pi and should be connected with Raspberry Pi through GPIO pins. There is a built-in driver for the microphone array to record audio with all eight microphones simultaneously.

Performance Measurements of Microphones on the MATRIX Creator

Our circular array is composed of 8 microphone sensors, and each performs an independent measurement of the surrounding acoustic environment. When implementing a beamforming algorithm on a real microphone array, the relative amplitude gain and relative phase difference between the microphones will influence the performance of the beamformer. Therefore, selecting microphones with similar parameters or performing a pre-calibration process helps to achieve stable performance of beamforming. A study presented in [54] shows that the microphone array is more sensitive to variations in amplitude gain than variations in phase.

With the intention of obtaining the amplitude gain for each microphone on MATRIX Creator, experiments were conducted to collect data. We placed the sensor array at the bottom center and the loudspeaker at the top center of the custom-built rig, as shown in Fig. 4.2a. Sets of 5-second-length sinusoidal signals with different frequencies received by sensor array were used to analyze microphone gains. The sinusoidal source was generated by an iPhone application called ‘Tone Generator’ and was played by a Bluetooth loudspeaker. Each received signal had its DC component removed, interpolated (upsampled) by cubic spline, and passed through a narrow-band band-pass filter before analyzing. The rig helps to ensure the stable distance between audio source and microphone array; the foam located at bottom and top reduces acoustic reflection and absorption.

A sliding window, which is set to be one period of the signal, is used to traverse every 5 seconds of the interpolated signal, and the maximum amplitude of each microphone within the sliding window is recorded. As depicted in Fig. 4.2b, microphone 1 has the least amplitude gain among all the microphones in this specific MATRIX Creator we have. This conclusion is consistent across the frequency range from 440Hz to 5000Hz. We also see that the gain is frequency-dependent for all microphones.

4.2.2 Ricoh Theta V

The full-view fisheye camera we used in this project is called Ricoh Theta V, which aims at providing an immersive 360° video in one shot. Instead of using multiple camera sensors to cover the entire space, Ricoh camera delivers the convenience of shooting the whole surrounding in all directions using 2 ultra-wide fisheye lenses. The lens on the side with a button on the camera has been defined as the ‘front’ lens, as shown in the left part in Fig. 1.1, the other lens on the non-button side is the ‘rear’ lens. Each lens on the camera has a field-of-view (FOV) that exceeds 180 degrees. Since the FOV of two lenses is over 180°, the area covered in the edge portion of the two fisheye images overlaps to some extent.
Figure 4.2: **Left:** Rig for data collection. **Right:** Received max amplitude vs. frequency of eight Microphones on MATRIX Creator.

Through embedded software inside the Ricoh Theta V, the spherical image is automatically stitched from two fisheye images. We do not have access to the raw images in the current version of this camera. Nevertheless, we are able to get the raw video generated from Ricoh Theta V, which is in a format called dual-fisheye. It captures the entire scene in $360 \times 180$ degrees. With some post-processing, we can further convert the dual-fisheye format video into a set of panoramic frames. Fig. 1.3 is one converted example mentioned before. Fortunately, we have created a dataset that facilitates direct face detection or tracking of images/videos in the fisheye format. This can avoid unnecessary fisheye stitching steps and parallax differences from stitching.

### 4.3 System Overview

Fig. 4.3 illustrates an overview of our proposed visually-guided beamforming system, which we will discuss in more details in this section. The whole system can be separated into two major parts: audio and video processing. We start recording audio and $360^\circ$ video at the same time with MATRIX Creator and Ricoh Theta V camera, respectively. As mentioned in Section 4.2.1, MATRIX Creator is a fully-functioning daughter board with 8 microphones for Raspberry Pi. It connects with Raspberry Pi through GPIO pins. At the beginning of recording, multiple people are talking in front of the camera. Eight separate audio files are output from the MATRIX Creator, while one comprehensive $360^\circ$ video is recorded by the Ricoh camera. The experiment setup is shown in Fig. 4.4. The camera is fixed on the tripod and MATRIX Creator is placed 20 centimeters below the camera. There is a piece of
foam between MATRIX Creator and the wood table, which aims at minimizing the acoustic reflection.

In the current stage, we assumed the speakers would stay in the same position during the whole experiment. We applied the re-trained 360° face detector (HR-360 as mentioned in Section 2.5) on the recorded dual-fisheye video frame. As mentioned before, the face detector re-trained with our proposed 360° dataset has better performance on finding faces in fisheye images. Our application interface is user-friendly, and the user can select one specific target speaker to listen to, from all the detected speakers that appear in the video.

We used \((x, y)\) to represent the location of the target face with respect to the image center in the pixel coordinates. The pixel coordinates of the selected speaker are passed to the program, which undertakes 3D reconstruction on the image to find the target location in the real-world coordinate system relative to the centre of the camera. This reconstructed 3D location is converted to the polar angles (azimuth \(\phi\) and elevation \(\theta\)) relative to the camera origin. Now we have the polar angles to the selected human speaker relative to the hardware orientation. In terms of the 3D reconstruction, we constructed a lookup table to realize the mapping between image and real-world coordinate systems. This is because our application only needs two polar angle (azimuth and elevation) to drive the beamformer. It should be noted that we only considered the 2D points \((x, y)\) instead of \((x, y, z)\) in the image plane, where \(z\) is the depth information. \(z\) could be found through a depth camera, or deep learning methods can be exploited to learn the face size at different distances between a person and
the camera. The Table 4.1 is used to map 2D image points to the real-world azimuth angles \( \phi \). We use several ellipses to cover different space regions, as shown in Fig. 4.7a and Fig. 4.7b. 

\( a \) and \( b \) in the table are the length of the semi-major axis and semi-minor axis, which help to define the ellipse. Each fisheye image from Ricoh Theta V uses 936 pixels in height to represent 180° vertically. To simplify the calculation, we assume every 5.2 pixels represent 1 degree in height. The elevation angle \( \theta \) can be directly computed using this relationship.

As mentioned earlier, our 360-degree camera consists of two fisheye lenses, and we designate the button side as the front and the non-button side as the back. The raw video recorded by this camera is composed of frame-by-frame dual-fisheye images, one example is shown in Fig. 4.5. We can think of the entire space (360°) as consisting of two fisheye images/videos, as shown in Fig. 4.6, which illustrates a view from the top. We defined the front fisheye lens covering the region from 0° to 180°, and the rear lens covering the area from 180° to 360°. Since the FOV of each lens exceeds 180 degrees, the edge regions captured by the two lenses will overlap each other, as the shaded area presented in the right part of Fig. 4.6.

We divided the whole 360° space into 12 equal regions, as depicted in Fig. 4.6. We marked a circle with a radius of one meter and placed the tripod in the center of this circle. Twelve points were marked on the circumference of the circle, and every two points were spaced at 30 degrees. The camera started recording when a 160 cm tall person stood on those points. The recorded video was saved as a dual-fisheye video, we separated every fisheye frame from the front and rear lenses into 7 regions, as shown in Fig. 4.7a. Due to the nature of distortion, each area that appears in a fisheye image has a different size. The two regions
<table>
<thead>
<tr>
<th>$a$ (pixel)</th>
<th>$b$ (pixel)</th>
<th>$\pm x$</th>
<th>azimuth angle $\phi$ (L/R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>468</td>
<td>480</td>
<td>+</td>
<td>180° / 0°</td>
</tr>
<tr>
<td>468</td>
<td>353</td>
<td>+</td>
<td>210° / 30°</td>
</tr>
<tr>
<td>468</td>
<td>201</td>
<td>+</td>
<td>240° / 60°</td>
</tr>
<tr>
<td>468</td>
<td>69</td>
<td>±</td>
<td>270° / 90°</td>
</tr>
<tr>
<td>468</td>
<td>201</td>
<td>−</td>
<td>300° / 120°</td>
</tr>
<tr>
<td>468</td>
<td>353</td>
<td>−</td>
<td>330° / 150°</td>
</tr>
<tr>
<td>468</td>
<td>480</td>
<td>−</td>
<td>360° / 180°</td>
</tr>
</tbody>
</table>

Table 4.1: The lookup table to map pixel points in the image domain to the real-world coordinate system. The origin of the image coordinate system is the center of the image, and x-axis is towards the left. It is assumed that the origin of Cartesian coordinates is the center of the ellipse; the x-axis is the minor axis for ellipses. $a$ and $b$ are the height and width of ellipses, respectively. $\pm x$ shows whether the image point is located on the positive x-axis or negative x-axis. L/R represents the image captured from the left/right hemispherical lens.

Figure 4.5: One frame of dual-fisheye videos from Ricoh Theta V.
Figure 4.6: Cameras with two fisheye lenses can capture the entire surroundings; each lens covers the half-space. The whole 360° area is split into 12 sub-regions. The shaded portion shown on the right represents the overlapped area due to the ultra-wide FOV.

Figure 4.7: (a) Fisheye FOV is split into 7 regions. (b) Ellipse model used for the lookup table.
at the edge of the fisheye image represent 0°/360° and 180°, respectively. People/objects located in these regions will be captured by both front and rear lenses because of the overlap in wide FOV, as shown in the right part of Fig. 4.6. In other words, they will appear in two fisheye images. A lookup table was constructed between the pixel location on the image and their corresponding angles. In this way, \((x, y)\) can be easily mapped into the polar angles \((\theta, \phi)\). Besides, linear interpolation had been used during 3D reconstruction, which helped to improve the accuracy of remapping. The following gives an example of how linear interpolation works for the region of ‘90°’. According to Table 4.1, the ellipse for the ‘90°’ region was defined as \(a = 468\) pixels and \(b = 69\) pixels, and herein we assumed every \(69/15 = 4.6\) pixels represent one azimuth degree in this region. The bisection method with different \(b\) value was applied until the image point \((x, y)\) intersects with the ellipse.

The polar angles \((\theta, \phi)\) are now used for beamforming the circular microphone array, in order to maximize the signal gain in that direction. So the beamforming suppresses the signals from other speakers and attenuates the unwanted noise coming from other directions. All the 8 audio channels and the angles \((\theta, \phi)\) are the inputs of the beamforming system.

The beamformed signal should be synchronized with the 360° video. To simplify the synchronization process, we decided to pat three clear claps at the beginning of the experimental recording, and no speaker spoke during the first 3 seconds. Thanks to the beamformer, the target speech is emphasized in the output signal. However, the beamformer may enhance the noise as well. This feature may increase the difficulty of finding three handclaps in time domain. Our solution is to align one recorded audio with the 360° video using three handclaps, insert timestamps into all recorded audios, and then perform beamforming. Timestamps help perform synchronization between beamformed signal and 360° video. The final result of this pipeline is a synchronized 360° video with the enhanced/beamformed target speaker’s speech.

It should be clarified that the system for real-time processing has not been built yet, but a prototype is already available to work offline. The experiments in Section 4.4 do not relate to the whole proposed system from this section, but to the sensitivity of beamforming to the accuracy of target/interference direction.

### 4.4 Sensitivity to the Accuracy of Face Detection

The visually guided beamforming system might be sensitive to the imprecise knowledge of the DOA of desired signal and interference estimated by face detector. Here we study the variation of SDR and intelligibility when the estimated face position is different from the true position. Both free space and reverberant environment are considered.
4.4.1 Experimental Setup

Similar processing steps for simulation as described in Section 3.5.1 were conducted: female speech and male speech were placed 1 meter away from the center of the circular array, and female speech source was always placed closest to microphone 3. Female speech is treated as the desired signal, while the male is the interference. To model the effect of face localization errors in face detection, let $\alpha$ be the correct angle between the target and interference, and let $\beta$ be the angle error caused by imperfect face detection. In Fig. 4.8, ‘Female’ is the actual position of the target speaker, while ‘Female’ is the estimated location $\beta$ degrees away from the actual position. In order to simplify the simulation, the interference DOA is considered as a perfect estimation without error while assessing how the inaccurately detected target position will affect the beamforming performance. On the other hand, when evaluating the impact of imprecise interference DOA on MVDR-2C, it is assumed that the target source located by the face detector is perfect.

The simulation considered three target-and-interference settings: $\alpha$ is $45^\circ$, $90^\circ$ and $180^\circ$. For every $\alpha$, 10 distributions of $\beta$ are considered as the DOA error, and for each distribution of $\beta$, 100 simulation were performed. Mathematically, $\beta \sim N(0, \sigma^2)$, where $\sigma$ varies from $1^\circ$ to $10^\circ$ in steps of $1^\circ$, to get the 10 distributions.

4.4.2 Results Analysis

Free Space

The SDR and STOI are calculated when the target direction is considered as non-perfect, due to errors from face detection, and then compared to the ideal case stated in Section 3.5.3. The following reported results are simulated in free space. Fig. 4.9 demonstrates the deviation in terms of SDR and STOI. The x-axis represents the DOA error $\beta$ with standard deviations, which varied from $1^\circ$ to $10^\circ$. The negative values in ‘SDR difference’ and ‘STOI
Figure 4.9: SDR (left) and STOI (right) changed with different error angle $\beta$ among four beamformers. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDDB-360 described in Section 2.5.

difference' represent the performance drop, compared to the case where the camera module provides the perfect speaker location without error. From the left figure, a conclusion can be drawn that DAS and GSC beamformers have high DOA error tolerance in terms of SDR, while both MVDR and the proposed MVDR-2C are more sensitive to an error in target DOA. In particular, MVDR and MVDR-2C reach 0.5 dB SDR drop when the standard deviation for $\beta$ is approximately 6$^\circ$. Our new proposed beamformer seems to be more sensitive towards DOA error than conventional MVDR, in terms of SDR. For the intelligibility of the output, all four methods show the stability with respect to varying $\beta$, the difference between STOI under free space situation and the reverberant room is only around the order of $10^{-3}$.

Although in terms of SDR, the performance of MVDR and MVDR-2C declines faster, this does not indicate that these two MVDR-based beamformers are worse than DAS and GSC in absolute terms. The previous chapter already showed that both MVDR and MVDR-2C offer higher performance by at least 2 dB SDR, compared with the other two beamformers (Fig. 3.10a and Fig. 3.12a). The absolute performance in terms of SDR and intelligibility as functions of DOA error for the four beamformers are shown in Fig. 4.10. The solid line shows the mean value at different values of error angle $\beta$, while the semi-transparent bands around the mean value correspond to one standard deviation of SDR/STOI from the mean. It can be seen from the figure that the SDR values of MVDR and MVDR-2C tend to decrease as estimated DOA error grows, while the performance of DAS and GSC stays stable. The standard deviation of SDR for MVDR-2C increases more significantly than the other three beamformers, but MVDR-2C still provides the best performance overall, while MVDR is the second-best. Surprisingly, GSC and MVDR have higher variation than DSB and MVDR-2C in terms of intelligibility.
Figure 4.10: SDR (left) and STOI (right) with different error angle $\beta$ among four beamformers. Solid lines represent the mean, and the semi-transparent region shows one standard deviation around the mean. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDDB-360 described in Section 2.5. The experiments were conducted under free space.

The real error rate of the re-trained face detector in fisheye images, as described in Section 2.5, was calculated. The standard deviation of detection error is 4.56°, that means the detected face by our re-trained FDDB-360 will have a $\pm 4.56^\circ$ error. We marked this value as the green vertical lines shown in Fig. 4.9 and Fig. 4.10. When the standard deviation of $\beta$ is 4.56°, MVDR-2C has an at least 5 dB SDR improvement over the conventional MVDR, and has an almost perfect result in terms of STOI.

In addition, Fig. 4.11a illustrates how the DOA estimation error of interference will affect the performance of MVDR-2C. Note that other three beamforming methods do not depend on the interference direction, so their performance is not affected by interference DOA estimation errors. In this case, the target position is assumed to be estimated without error, while DOA estimation of interference has an error of $\beta$ with zero mean and varying standard deviations. Besides, we only report the result for $0^\circ \leq \alpha \leq 180^\circ$ because of the symmetric performance illustrated in Fig. 3.10. According to Fig. 4.11a, when the target and interference are placed in the opposite directions, where $\alpha = 180^\circ$, the proposed MVDR-2C has stable SDR performance, for the range of DOA error standard deviations up to $10^\circ$. A very noticeable trend is the steady decrease in SDR when $\alpha$ is $30^\circ$ and $45^\circ$. The graph leads to the conclusion that the smaller the $\alpha$ between the target and interference, the more sensitive is the performance of MVDR-2C. Three angles were selected to further observe the variations of SDR, as shown in Fig. 4.11b. The semi-transparent band around the solid lines (averaged SDR) represents one standard deviation for each distribution of $\beta$. When $\alpha$ is $45^\circ$ and the distribution of $\beta$ is $\mathcal{N}(0, 10^2)$, the standard deviation of SDR reaches 1.5 dB, whereas for the same distribution of $\beta$, SDR is impressively stable when $\alpha$ is $180^\circ$. This
Figure 4.11: (a) SDR changes for MVDR-2C with various DOA estimation error for interference under a free space condition, corresponding to different target-and-interference position settings. (b) SDR for MVDR-2C with various DOA estimation error for interference in a free space environment.

Figure can also be linked to Fig. 3.10, where SDR reaches a local maximum when $\alpha$ is 45° and SDRs are 3.5 dB and 1 dB less when $\alpha$ is 90° and 180°, respectively.

Reverberant Environment

The impact of imprecise face detection of the target speaker was also examined in the case of reverberation. The SDRs and STOIs of output signals from different beamformers are compared against each other. In a reverberant environment (Fig. 4.12), all four algorithms have a considerable performance drop in both SDR and intelligibility compared with the experiments in free space. MVDR-2C outperforms DAS and GSC by an average of 2 dB in terms of SDR. MVDR-2C also provides 2.5 dB SDR enhancement over the conventional MVDR when $\alpha$ is 45°, whereas the SDRs of MVDR and MVDR-2C do not have a considerable difference when $\alpha$ is 90° and 180°. The intelligibility in terms of STOI of beamformed signals is shown in Fig. 4.12b, where MVDR-2C always delivers the best performance, regardless of the target-and-interference settings and face detection errors.

Another set of experiments was conducted to assess how inaccurate face detection of the interfering speaker influences the proposed MVDR-2C performance. With different interfering signals’ DOA error $\beta$, the experimental SDR results of MVDR-2C are shown in Fig. 4.13. The SDRs with reverberation are surprisingly more stable than the performance in free space. For instance, MVDR-2C has a maximum of 0.3 dB SDR drop in the reverberant room when $\alpha$ is 30° relative to the case where the knowledge of interference is perfect, as depicted in Fig. 4.13. Meanwhile, MVDR-2C simulated in free space has a maximum of 2.4 dB SDR drop when $\alpha$ is 30°. It can also be seen that the beamforming performance is more stable as the angle between the target and interfering signals increases.
Figure 4.12: SDR (left) and STOI (right) with different error angle $\beta$ among four beamformers. Solid lines represent the mean, and the semi-transparent region shows one standard deviation around the mean. The green vertical line demonstrates the standard deviation of the detection error estimated by re-trained FDDB-360 described in Section 2.5. The experiments were simulated in a reverberant room.

Figure 4.13: SDR changes for MVDR-2C with various DOA estimation error for interference in a reverberant environment, corresponding to different target-and-interference position settings.
4.5 Conclusions

In this chapter, the proposed ‘visually guided beamforming system’ pipeline has been explained, along with the specification of two involved hardware components, MATRIX Creator and Ricoh Theta V. Based on the performance analysis of the given circular microphone array, the experimental setup has been adjusted in order to mitigate the effect of unbalanced amplitude gains among microphones. Finally, the effect of imprecise knowledge of both target and interference DOA is simulated and discussed in detail. We can draw a conclusion that DAS and GSC are less sensitive to DOA error than MVDR and the proposed MVDR-2C. However, MVDR and MVDR-2C still have better performance in absolute terms for a range of DOA estimation error distributions. These conclusion are valid in both free space and reverberant conditions.
Chapter 5

Conclusion and Future Work

5.1 Conclusion

The primary goal of this work is to improve beamforming performance by obtaining target location hints from a vision device. Therefore, a vision-guided beamforming system was proposed. In this thesis, vision and audio systems were discussed separately first, then the combination of a 360° camera and a circular microphone array was simulated.

Chapter 2 presented three new fisheye-looking datasets named FDDB-360, Wider-360 and VOC-360, derived from existing datasets of conventional images FDDB, Wider Face and VOC accordingly. A re-training experiment with FDDB-360 showed that the generated 360° datasets could be exploited to re-train an existing face detection model, and the re-trained model has an improved detection performance on fisheye images. At the end of this chapter, our proposed fisheye model, which is used to convert regular images into fisheye images, had been compared with another fisheye model and shown to deliver a close and reasonable approximation to a real fisheye image.

In Chapter 3, algorithms of three well-known beamformers (DAS, GSC and MVDR) had been discussed; an improved two-constraint MVDR (MVDR-2C) based on the conventional MVDR was proposed. The new MVDR-2C takes interference into account and aims at maximizing the power ratio between the target and that specified interference. The beam patterns of DAS, MVDR and MVDR-2C corresponding to different frequencies were demonstrated and verified to be coincident with theory. The radiation pattern of MVDR-2C provided a sharper ‘null’ for interference, compared with both DAS and MVDR. Several computer-simulated experiments were conducted, and different evaluation metrics were adopted to assess the performance of the four beamformers. Both MVDR and MVDR-2C outperform DAS and GSC in all cases as predicted, and the proposed MVDR-2C delivers better performance than MVDR, with appropriate diagonal loading.

In Chapter 4, a full pipeline of our visually guided beamforming system had been presented with details. MATRIX Creator with a circular array of 8 microphones and Ricoh Theta V had been integrated into this system. Rather than using beamforming techniques
to estimate the sound source location as used by many existing works, this work employed the 360° camera to provide an estimate of the direction of speakers relative to the microphone array. The test for error tolerance of target direction-of-arrival (DOA) among different beamformers reveals that DAS and GSC are less sensitive to DOA error. As for MVDR and MVDR-2C, the performance according to SDR value is influenced by the error, but they are still able to deliver the best performance overall. The effect of interference DOA error on MVDR-2C indicated that the further the target and interference were, the less sensitive the beamformer is to interference DOA estimation error.

5.2 Future Work

Based on this research, there are a few potential extensions that could be done in the future:

1. Section 2.5 mentioned that the re-trained face detector could not remember all the existing features during transfer learning. In order to alleviate this ‘forgetting’ and reinforce primary trained features while learning new capabilities, data from the original training set can be randomly inserted into the re-training process. How much of this original data is needed is an interesting research question.

2. The spherical microphone array like VisiSonics with 64 microphones and 5 cameras can be involved to perform more accurate speaker localization and beamforming. One possible benefit from this device is that the camera and microphones are already synchronized and time aligned.

3. Deep learning-based audio-visual models are a new trend to enhance the speech in the signal processing field, as the visual information is beneficial to understand the context better. Specifically, a more precise lip-reading model can be incorporated to supplement beamforming and extract target speech from a noisy environment.
Bibliography


