GDAŃSK UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATICS

The author of the PhD dissertation: Kuba Łopatka
Scientific discipline: telecommunication

DOCTORAL DISSERTATION

Title of PhD dissertation: Adaptive system for recognition of sounds indicating threats to security of people and property employing parallel processing of audio data streams

Title of PhD dissertation (in Polish): Adaptacyjny system rozpoznawania dźwięków znamionujących sytuacje zagrażające bezpieczeństwu osób i mienia z zastosowaniem równoległego przetwarzania strumieni danych fonicznych

Supervisor *signature*	Second supervisor *signature*
Andrzej Czyżewski, Prof., D.Sc., Eng.	---
Auxiliary supervisor *signature*	Cosupervisor *signature*
---	---

Gdańsk, year 2015

Declaration of Authorship

I, Kuba Łopatka, declare that this thesis titled, 'Adaptive system for recognition of sounds indicating threats to security of people and property employing parallel processing of audio data streams' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

GDANSK UNIVERSITY OF TECHNOLOGY

Abstract

Faculty of Electronics, Telecommunications and Informatics
Multimedia Systems Department

Doctor of Philosophy

Adaptive system for recognition of sounds indicating threats to security of people and property employing parallel processing of audio data streams

by Kuba Łopatka

A system for recognition of threatening acoustic events employing parallel processing on a supercomputing cluster is featured. The methods for detection, parameterization and classification of acoustic events are introduced. The recognition engine is based on threshold-based detection with adaptive threshold and Support Vector Machine classification. Spectral, temporal and mel-frequency descriptors are used as signal features. The algorithms are implemented in a supercomputing environment utilizing a specialized framework for processing multimedia data streams. The recognition engine is evaluated in various conditions, both using pre-recorded signals and real-world events. First, an evaluation in laboratory conditions is performed to simulate selected acoustic environments and evaluate the recognition rates in noise. Subsequently, the recognition rates are assessed in various practical situations (related to indoor and outdoor surveillance) and compared with the results obtained in simulations. The adaptation of event detection evaluated by comparing different approaches to adapting the detection thresholds. Finally, parallel processing is introduced to improve the performance of the developed recognition engine. The experiments utilizing a supercomputing platform are introduced, which show that the employment of parallel processing leads to significant shortening of the time required to make the decision. The possible practical applications of the developed methods are outlined, including surveillance of urban space, public events or private property.

Acknowledgements

Firstly, it is all thanks to my Mom and Dad who made my education possible, sent me to school, University and to English classes, and to my Sister, who always inspired me to better myself.

I would also like to thank my loved ones, my family and friends for all the support they gave me and for always believing in me.

I would like to thank my supervisor, professor Andrzej Czyżewski, for all the advice and knowledge he has provided me with.

Coworkers from the Faculty of Electronics, Telecommunications and Informatics and from Multimedia Systems Department made this time spent on doctoral studies a bright and fascinating experience. I could not get this far without you!

I would like to thank all the institutions which provided me with scholarship and thus made this research possible. In particular: the Dean and doctoral studies of Faculty of Electronics, Telecommunications and Informatics, *InterPhd* project, and Pomorskie Voivodeship (*InnoDoktorant* project).

Special thanks to Dr. Józef Kotus for providing me with the localization algorithms and for cooperating with me on many experiments featured in this thesis and in publications.

Finally, I would like to thank the people involved in the research projects *Mayday Euro 2012*, *INDECT* and *INSIGMA* which supported parts of this work.

Kuba Łopatka

Contents

Abbreviations

AED	**A**coustic **E**vent **D**etection
ANN	**A**rtificial **N**eural **N**etwork
AVS	**A**coustic **V**ector **S**ensor
CPU	**C**entral **P**rocessing **U**nit
DET	**D**etection **E**rror **T**radeoff
DCT	**D**iscrete **C**osine **T**ransform
DFT	**D**iscrete**F**ourier **T**ransform
DoA	**D**irection of **A**rrival
DSP	**D**igital **S**ignal **P**rocessing (or Processor)
DWT	**D**iscrete **W**avelet **T**ransform
EER	**E**qual **E**rror **R**ate
FFT	**F**ast **F**ourier **T**ransform
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GMM	**G**aussian **M**ixture **M**odel
GPU	**G**raphics **P**rocessing **U**nit
HMM	**H**idden **M**arkov **M**odel
IP	**I**nternet **P**rotocol
LPC	**L**inear **P**redictive **C**oding
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
MIR	**M**usic **I**nformation **R**etrieval
MPEG	**M**otion **P**ictures **E**xperts **G**roup
PCM	**P**ulse **C**ode **M**odulation
PDF	**P**robability **D**ensity **F**unction
PSD	**P**ower **S**pectral **D**ensity
PTZ	**P**an-**T**ilt-**Z**oom

RAM	**R**andom **A**ccess **M**emory
RTSP	**R**eal-**T**ime **S**treaming **P**rotocol
SNR	**S**ignal to **N**oise **R**atio
SPL	**S**ound **P**ressure **L**evel
SVM	**S**upport **V**ector **M**achine
TDOA	**T**ime **D**ifference **O**f **A**rrival
VAD	**V**oice **A**ctivity **D**etection
XML	E**X**tended **M**arkup **L**anguage

For abbreviations denoting signal features, see Table 4.1.

Symbols

Notation:

x	scalar
$\|x\|$	absolute value
$\mathbf{x}$	vector
$\|\mathbf{x}\|$	norm of vector
$\mathbf{X}$	matrix
$\mathbf{X}^T$	matrix transposition
X	set

Sound and wave motion:

p_a	acoustic pressure [Pa]
$\mathbf{u}$	particle velocity vector [m/s]
$u_{x,y,z}$	particle velocity component in x, y, z direction [m/s]
$\mathbf{I}$	sound intensity vector [W/m^2]
ρ	density of fluid [kg/m^3]
f	frequency [Hz]
ω	angular frequency [rad/s]
Φ	phase of signal or wave [rad]
ϕ	azimuth angle [rad]
θ	elevation angle [rad]

Signals:

t	time [s]
n	sample index
$x(t)$	continuous time - continuous value - analog signal

$x[n]$	discrete time - discrete value - digital signal
$X[k]$	DFT value of signal x for k-th frequency bin
$P_x(f)$	power spectral density of signal x for frequency f
$P_x[k]$	power spectral density of signal x for k-th frequency bin
SR	sampling rate [S/s] (samples per second)
∇x	gradient of x

Random variables:

X	random variable
$\overline{X}, \mu$	mean of random variable
$< X >_t$	average of random variable over time
σ	standard deviation
$P(x) = P(X = x)$	probability of random variable yielding value x
$p(x)$	probability density function (PDF)
$erf(x)$	error function

Decision systems:

tp	number of true positives
TP	true positive rate
fp	number of false positives
FP	false positive rate (false alarm probability)
FN	false negative rate (miss probablility)
κ	Cohen's Kappa statistics

Pracę dedykuję moim Dziadkom.

Chapter 1

Introduction

In traditional human-operated surveillance human security personnel is prone to overlooking dangerous situations, for instance due to limited attention span or technical limitations (such as the insufficient number of monitors for displaying video signals). Automated computer-based surveillance (also referred to as *smart surveillance*) comes as an aid in such situations. Obviously, it is best to think of such techniques as a reinforcement, not the replacement of human supervision. Modern systems for automatic surveillance are mostly based on video-based event detection. In this thesis, the sound recognition methods which can be used in acoustic-based surveillance are examined. The analysis of sound, which is a separate modality, provides rich information which can enhance the efficiency of detection of hazardous events. The introduction of audio to surveillance enables detecting events which are not apparent in the video data. Such events include firing weapons, explosions or screams. The other advantage of incorporating the acoustic medium is that the sound is not affected by the conditions which significantly deteriorate the audio image, such as obscured, crowded or dark scenes.

In this dissertation the methods for automatic acoustic event detection for surveillance purposes are introduced and their efficiency is examined in the environment of a supercomputing cluster. In the introductory chapter, first the historical background of the problem is outlined. Next, the goals and scientific theses are posed. Subsequently, the organization of the thesis is outlined. At the end of the chapter the author's publications are listed, being the results of a five-year-long research on the subject.

1.1 Audio in surveillance

Even though audio surveillance nowadays is not as popular as the ubiquitous video-based surveillance, it is worth noting that using sound to monitor the activity of people precedes the visual systems by a couple of decades. It was only when telephony was invented, that people started to listen to each other's conversations. Later, in the late 1890's, a device called the telegraphone was invented, which comprised a telephone and a magnetic recorder. From the 1920's to 1950's video became to take over as a surveillance medium. Efforts were made to connect the camera to the CRT (cathode ray tube) monitors for live video streaming and to record the image on magnetic tape. Thus, Closed-Circuit Television (CCTV) was invented, which became the prevailing surveillance technology in the latter half of the 20^{th} century [1].

The massive deployment of CCTV systems started in the 1960's. Up to nowadays, Closed-Circuit Television is understood as a technology for transmitting the live feed from multiple cameras, installed e.g. in cities or inside buildings, to a control center, where the operator views the streams on one or multiple screens. To detect the violation of law or any other abnormal and threatening situations, human attention is necessary. However, due to the human's limited attention span, an automatic image analysis was later incorporated to reinforce the detection of abnormal situations. The development of IP CCD (Charge-Coupled Device) cameras, computer vision and video analytics enabled implementing image recognition, object detection and video event detection algorithms in surveillance systems [2].

The development of IP cameras allowed for transmitting sound and vision in the same telecommunication channel. Typically, the sound is either recorded with an external microphone connected to the camera or with a built-in microphone. Two-way audio is also frequent in IP cameras, enabling the operator to send voice back to the monitored area. Similarly to the video event detection, methods for detecting acoustic events in the audio stream are incorporated into the audio-visual surveillance system. The prior goal was to start recording vision after a sound had been detected. In such a case, the detection algorithm reacted only to the level of the signal. While the automatic

sound recognition technology matured, more sophisticated event detection algorithms were incorporated. As a result, modern systems have the capability of detecting acoustic events automatically and also localizing them, i.e. indicating the location of the sound source. A review of the current state of the art in this matter is featured in Chapter 2.

1.2 Goals and scientific theses

The primary goal of this thesis is to examine the performance and efficiency of automatic sound event recognition in real conditions operating on a supercomputing cluster. To achieve this goal, the algorithms for detection and classification of acoustic events are engineered. The developed methods are implemented in a supercomputing environment. The sound recognition engine is then evaluated in practical conditions, in the presence of noise. The influence of the conditions on the recognition accuracy is examined. The performance of the algorithms in real-life conditions is evaluated. Finally, the benefits of employing the supercomputing cluster are investigated. The parallel processing techniques are utilized to accelerate the decision making, which ensures faster and more reliable security surveillance.

The hardware platform for implementing the developed methods is a supercomputing cluster, namely Galera+ cluster located in Gdańsk University of Technology. The motivation for employing a supercomputer for automatic surveillance is the growing number of streams from surveillance cameras and microphones. Large processing powers are needed to handle multiple video and audio streams. This thesis aims to show that the employment of supercomputing leads to more efficient analysis of audio stream as far as hazardous event recognition is concerned. The majority of the work is carried out within the project *Mayday Euro 2012* conducted in Gdańsk University of Technology [3]. The author of the dissertation worked in the project from 2009 to 2012. The methods described in the thesis are implemented in the framework for analysis of multimedia data streams created in the project, named KASKADA. The KASKADA platform, described more deeply in Chapter 5, facilitates the creation and the exploitation of multimedia stream processing services.

The recognition engine developed in this work is capable of recognizing one of predefined classes of events: *explosion, broken glass, gunshot, scream* and *other*. The methods employed are suited for an online operation and, thanks to the use of a supercomputing platform, processing multiple streams simultaneously. The engine is intended to work both in indoor and outdoor conditions. It also provides the possibility of localizing the sound source in addition to recognizing the type of event, provided a specific acoustic vector sensor is used. Another key feature of the developed algorithms is *adaptation*, i.e. the feature of adapting the detection thresholds, to the changing acoustic conditions of the environment. This feature improves the performance of the sound recognition engine in practical conditions.

The author of the dissertation aims to prove the following scientific theses:

1. **The developed methods for detection, parameterization and classification of selected hazardous acoustic events enable sufficiently low loss achieved in practical conditions to be used for security surveillance purposes.**

   The aim is to assert that the methods employed by the author are a correct tool for discerning between the selected classes of acoustic events. To prove this thesis, a series of experiments is conducted, both employing the signals from the training set and real-world events. The work is considered successful if the loss generated by the developed decision system in practical conditions is sufficiently low for the methods to be usable in an automatic security surveillance system, i.e. the methods employed enable a reliable detection of sounds related to danger.

2. **The proposed way of adaptation of the detection threshold to the variance and dynamics of the level of the acoustic background reduces the detector's equal error rate compared to the adaptation to average sound level.**

   The recognition engine is designed to be flexible and adaptive. In particular, the designed algorithms for detecting acoustic events are developed in such a way that the detection threshold is automatically adjusted to the changes in the acoustic

environment. In the developed methods, not only is the detector adapted to the average sound level of the acoustic background, but also to its variance and temporal change rate. It is shown in experiments that such an approach improves the performance of the event detector by lowering the equal error rate.

3. **The implemented parallel processing schemes on a supercomputing cluster enable nearly real-time performance of the hazardous sound recognition algorithmic chain.**

   The work featured in this dissertation is pioneering as far as the recognition of hazardous acoustic events on a supercomputing cluster is concerned. The aim of this thesis is to prove that such algorithms can be successfully implemented in the cluster environment and benefit from employing parallel processing. It is shown in experiments that the employment of supercomputing improves the performance of the recognition engine, as far as the decision making time is concerned. The specialized framework for processing multimedia data streams and the parallel execution of operations related to sound recognition reduces the latency to a minimum, thus enabling nearly real-time performance. In other words, the goal is for the recognition engine to operate with a latency comparable to the state-of-the-art achievements in the field, including low-latency audio.

1.3 Outline of the dissertation

The dissertation is composed of 9 chapters. Chapters 2 and 3 constitute the theoretical part. In Chapter 2 the techniques used for automatic sound event recognition are described. A theoretical background concerning some of the algorithms used in this work is also provided. Chapter 3 introduces the topic of parallel processing of audio data, employing multi-core and multi-processor systems, as well as supercomputing clusters.

In the first of the practical chapters - Chapter 4 - the developed sound recognition engine is described. The algorithms utilized for detection, feature extraction and classification are discussed in detail. In order to evaluate the proposed methods the recognition engine is implemented on the supercomputing cluster. In Chapter 5 the hardware architecture

and software framework employed in the supercomputing environment are presented. The methods are first evaluated on a set of training signals, which is featured in Chapter 6. The set of signals used to train the sound event recognition engine is described, as well as the results of evaluation of the developed algorithms on the training data. The evaluation of the performance of the engineered methods in practical conditions is given in Chapter 7. Next, in Chapter 8 the experiments concerning parallel processing on the supercomputing cluster are described to show the benefits of employing a supercomputing platform for the task of sound event recognition. Finally, in Chapter 9 the conclusions are drawn and the results of the thesis are discussed.

1.4 Author's publications

The author of the thesis has been signed as co-author of the following publications related to the topic of the dissertation published in the years 2010-2015:

- journal articles

J. Kotus, K. Łopatka, A. Czyżewski, and G. Bogdanis, "Processing of acoustical data in a multimodal bank operating room surveillance system," *Multimedia Tools and Applications*, published online http://dx.doi.org/10.1007/s11042-014-2264-z, 17.10.2014; IF: 1.058[1]

K. Łopatka and A. Czyżewski, "Acceleration of decision making in sound event recognition employing supercomputing cluster," *Information Sciences*, vol. 285, no. 1, pp. 223–236, 2014; IF: 3.893

J. Kotus, K. Łopatka, and A. Czyżewski, "Detection and localization of selected acoustic events in acoustic field for smart surveillance applications," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 5–21, 2014; IF: 1.058

K. Łopatka, J. Kotus, and A. Czyżewski, "Application of vector sensors to acoustic surveillance of a public interior space," *Archives of Acoustics*, vol. 36, no. 4, pp. 851–860, 2011; IF: 0.656

[1]Impact Factor according to Thomson Reuters Journal Citation Reports 2013 edition

- conference papers

K. Łopatka and A. Czyżewski, "Recognition of hazardous acoustic events employing parallel processing on a supercomputing cluster," in *138th Convention of the AES, in print*, (Warsaw), 2015

K. Łopatka, J. Kotus, and A. Czyżewski, "Evaluation of sound event detection, classification and localization in the presence of background noise for acoustic surveillance of hazardous situations," in *Multimedia Communications, Services and Security*, vol. 429 of *Communications in Computer and Information Science*, pp. 96–110, Springer International Publishing, 2014

J. Kotus, K. Łopatka, A. Czyżewski, and G. Bogdanis, "Audio-visual surveillance system for application in bank operating room," in *6th Int. Conf. on Multimedia, Communications, Services and Security*, pp. 107–120, 2013

K. Łopatka and A. Czyżewski, "Automatic regular voice, scream and raised voice recognition employing fuzzy logic," in *132nd Convention of the AES, preprint no. 8636*, (Budapest), 2012

J. Kotus, K. Łopatka, A. Czyżewski, and H. Krawczyk, "Multimedia system assisting lecturers and public speakers," in *INFOBAZY 2011*, pp. 80–86, 2011

K. Łopatka, J. Kotus, M. Szczodrak, P. Marcinkowski, A. Korzeniewski, and A. Czyżewski, "Multimodal audio-visual recognition of traffic events," in *Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on*, pp. 376–380, 2011

K. Łopatka, J. Kotus, and A. Czyżewski, "Monitoring of public events audience employing acoustic vector sensors," in *14th International Symposium on Sound Engineering and Tonmeistering*, 2011

K. Łopatka, A. Czyżewski, and H. Krawczyk, "Automatic recognition of events in audio data using supercomputer cluster," in *130th Convention of the AES, preprint no. 8337*, (London), 2011

K. Łopatka, J. Kotus, and A. Czyżewski, "Improving automatic surveillance by sound analysis," in *5th Future Security Conference*, pp. 51–51, 2010

J. Kotus, K. Łopatka, K. Kopaczewski, and A. Czyżewski, "Automatic audio-visual threat detection," in *IEEE Int. Conf. on Multimedia, Communications, Services and Security*, pp. 140–144, 2010

K. Łopatka, P. Żwan, and A. Czyżewski, "Dangerous sound event recognition using support vector machine classifiers," *Advances in Multimedia and Network Information System Technologies*, vol. 80, pp. 49–57, 2010

K. Łopatka, P. Żwan, and A. Czyżewski, "Parameterization of sounds for recognizing hazardous events," *Zeszyty Naukowe Wydzialu Elektroniki, Telekomunikacji i Informatyki Politechniki Gdanskiej: Technologie Informacyjne*, vol. 19, pp. 225–230, 2010

- book chapter

A. Ciarkowski, J. Cichowski, D. Ellwart, P. Guzik, K. Kopaczewski, J. Kotus, K. Lisowski, K. Łopatka, A. Matiolański, M. Papaj, M. Szczodrak, and G. Szwoch, *KASKADA platform and multimedia applications*, vol. 2, ch. Applications for recognition of persons and events. Gdansk University of Technology, 2013

Chapter 2

State of the art in sound event recognition

This chapter contains a review of the techniques and algorithms employed in automatic recognition of acoustic events. It also serves as a theoretical background providing the Reader with helpful information to understand the thesis. In the first section we introduce the problem of sound event recognition. Next, we mention the operations which have to be carried out to recognize the type of the event, namely feature extraction, detection of foreground events and pattern recognition by means of statistical classification. In the following section the metrics used to assess the accuracy of acoustic event detection and classification are introduced. The chapter is concluded by a review of the most representative approaches featured in the literature and some commercial applications.

2.1 Basic concepts

Two terms are often distinguished: acoustic event detection (AED) and acoustic event classification (AEC). The term AED is also often used to indicate the recognition process as a whole. Lyon also dubbed this field of research "Machine hearing" [21].

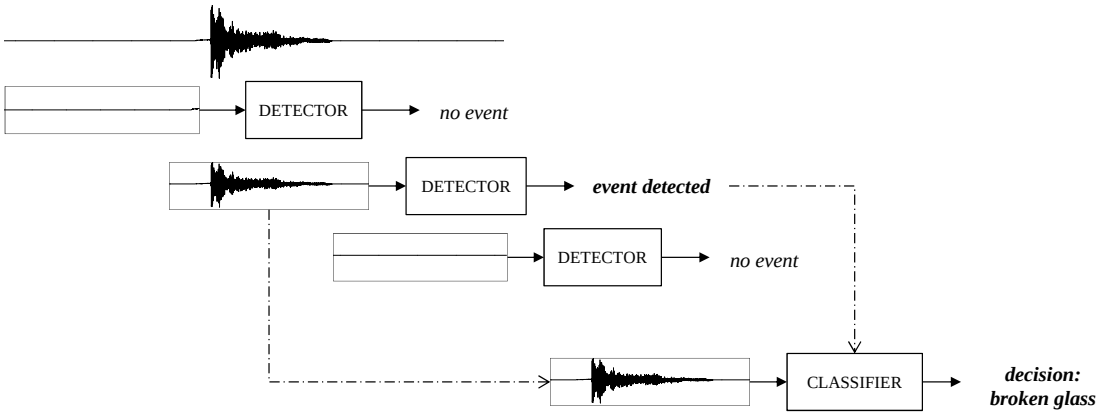


FIGURE 2.1: Illustration of the *detection-and-classification* approach

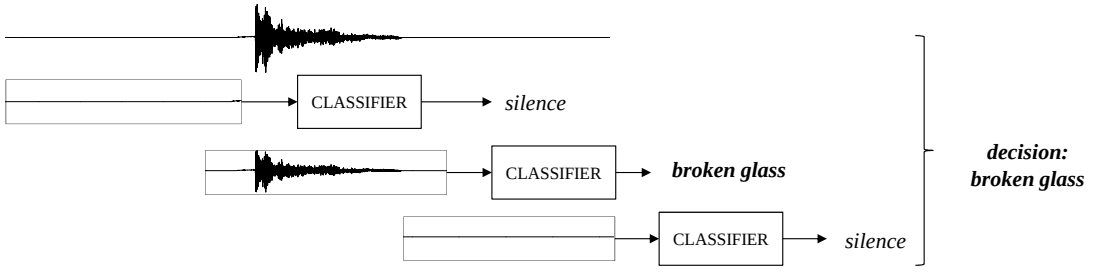


FIGURE 2.2: Illustration of the *detection-by-classification* approach

As assumed in this dissertation. the *detection* term refers to the process of separating the event from the acoustic background, i.e. indicating that an acoustic event took place at a specified time. The *classification* operation determines the *type* of event which was observed. Two approaches to acoustic event detection are present in the literature: *detection-and-classification* and *detection-by-classification* [22]. In the first approach, depicted in Figure 2.1, the recognition process consists of two steps - first the event is sought in the acoustic background by the *detector* and subsequently, after it has been detected, the type of the event is recognized by the *classifier*. In the *detection-by-classification* approach, shown in Figure 2.2, the sound is processed online by the classification algorithm and when the classifier recognizes an event of specified type - the decision is made that such event took place.

A vast majority of the state-of-the-art approaches employ statistical pattern recognition to recognize the type of event. The principle of statistical classification is to identify the type of the object on the basis of a feature vector comprising a limited number of parameters [23, 24]. The parameters are calculated from the signal during the *feature*

extraction operation. The signal features constitute a *feature vector*, which contains parameters whose purpose is to underline the distinct properties which allow for discerning between the different classes of events. The features are fed into the pattern recognition algorithm (classifier) which produces the decision.

In the statistical approach the classification process requires establishing a model which is generated from a set of exemplary observations, i.e. the *training set*. The operation of creating a model basing on the features extracted from the objects belonging to the training set is referred to as *training* of the classifier. Both the feature extraction operation, the choice of the training vectors and the structure of the classifier have a large influence on the performance of the sound event recognition engine.

2.2 Related fields and applications

From the historical point of view the field of sound event recognition originates partially from speech recognition and partially from musical instrument recognition [25]. In fact, some techniques have been adopted directly from these two domains, which will be indicated throughout the chapter. Initially, recognition of acoustic events was in the shadow of speech recognition. Since the late 1990s, the task of acoustic event recognition has grown into a separate field of research and the researchers have developed original methods and approaches.

The relationship with **Automatic Speech Recognition** (ASR) is a very obvious one. The similarity lies in the signal processing and pattern recognition algorithms employed. A vast majority of ASR engines relies on Mel-Frequency Cepstral Coefficients as signal features and Hidden Markov Models for classification [26]. Such an approach is often encountered in sound event recognition as well. One can also conceive that the task of recognizing words is similar to recognizing sound events. However, it is probably more appropriate to compare sound event recognition to **speaker recognition**, in which we focus on the type of sound source instead of the content of the signal. Jonathan Dennis provides a remarkable discussion of similarities and differences between ASR and AED in his PhD dissertation [27]. He claims that the major differences are in

the recording environment, the approach to feature extraction and pattern recognition techniques employed. In the case of ASR the signals are often close field with little noise, whereas in AED the distance from the microphone is much larger, thus deteriorating the Signal-to-Noise Ratio. The granularity of feature extraction is also different. In ASR the features are extracted on very short-time frame basis (typically 25 ms), whereas in AED the frames are slightly longer (100 - 500 ms), thus covering a whole *segment* of sound. In fact, the ASR engines are typically organized based on the phonetic structure of speech. This paradigm is not possible to adopt in AED, since sound events lack the repeatable systematic structure, which is present in speech. As far as pattern recognition is concerned, apart from Hidden Markov Models, structures such as Support Vector Machine and Artificial Neural Networks are prominent in the field of acoustic event recognition. It is worth noting that such structures, however almost non-existent in ASR, are often utilized in *speaker recognition*.

A controversy lies in the fact that ASR is often regarded as a *solved* problem. Such conviction stems from the high performance of commercial ASR engines. Thus, the detection of acoustic events, (which may deceptively seem less complicated) appears trivial to those loosely familiar with the subject. One should note that even though very efficient continuous speech recognition engines exist, automatic speech recognition is still a great challenge in difficult conditions, such as reverberation, far field or noisy environments [28]. It has to be noted that such conditions are almost always the case in real-life acoustic event recognition. Moreover, the intra-class variance of sound events is much greater than that of speech units. A sound of a certain type (e.g. gunshot) may have totally different character, depending on the type of weapons used, type of ammunition, firing conditions etc. Yet, it is still expected to be recognized as the same class of sound. Due to its difficulty, the problem can be compared to recognizing speech units, independent of the speaker, for all possible accents and pronunciation variants.

Another field which is closely related to sound event recognition is **Music Information Retrieval** (MIR). In MIR the signal features (MPEG-7 descriptors are predominant) and classifiers are employed to recognize different aspects of music, such as: genre, artist, instrument type, mood etc. [29]. It is typical in MIR systems that a very large database

of musical recordings is parsed [30–32]. Both in MIR and in AED, similar features (Mel-Frequency Cepstral Coefficients, MPEG-7 low-level descriptors) and classifiers (Support Vector Machine, Artificial Neural Network) are often used. The difference is that whereas musical tracks typically last for a few minutes, the sound events only last for up to a few seconds. Therefore, Music Information Retrieval is less time-critical. Moreover, in MIR it is a common paradigm to directly compare the patterns of known musical recordings, e.g. with a kNN (k-Nearest Neighbors) classifier. Such approach is regarded too expensive for AED, as far as computational time is concerned. Finally, some features which are particularly useful in MIR, are hardly usable in AED, e.g. parameters related to harmonics or musical key.

Acoustic event recognition benefits from the field of **Machine Learning** and **Pattern Recognition** [24]. The art of learning classification algorithms is employed to build classifiers which discern between the various types of sounds. The specific structures employed in this process are outlined further in this chapter, in Section 2.5. It has to be underlined that the recognition of sound events is a task which much differs from other problems encountered in pattern recognition, namely text categorization, classification of medical data, etc. The feature representations of sound events tend to be very different, even if the sounds sound much alike to human ear. This is precisely what makes recognizing the type of acoustic event a difficult problem to be solved.

The first application of sound event recognition was in underwater acoustics. From the 1980s the sound recognition methods have been employed to analyze the sonar signals and identify targets, especially maritime vessels [33]. Nowadays, the applications are widely spread from home automation [34] through meeting room assistance [35], infant cry detection [36] and aids for the hearing-impaired [37] to acoustic-based surveillance [38–40]. In the field of audio surveillance the events related to danger are most often recognized, e.g. gunshots [38, 41, 42] or screams [42, 43]. Nevertheless, except from expert systems (such as those presented by Maher regarding gunshots [44]), the state-of-the-art techniques encountered in the literature can be used regardless of the type of the recognized event and the application. In this dissertation the acoustic surveillance

application is emphasized. The review of existing approaches to the task of sound event recognition, both academic and commercial, is provided at the end of this chapter.

2.3 Signal features

First, let us consider that all acoustic events are stochastic processes, which we measure as time-variable acoustic pressure $p_a(t)$. After digitalization the acoustic pressure signal is stored in the computer memory as a series of samples $x[n]$, i.e. a digital signal. The signal itself and its values can be considered a random variable. The values of signal x for a limited range of sample indices $n_1 \leq n \leq n_2$ form a vector:

$$\mathbf{x} = \begin{bmatrix} x[n_1] & x[n_1+1] & ... & x[n_2-1] & x[n_2] \end{bmatrix}^T \tag{2.1}$$

The feature extraction operation transforms the vector $\mathbf{x}$ of length $N = n_2 - n_1 + 1$ to another vector v which has different dimensionality K:

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & ... & v_{K-1} & v_K \end{bmatrix}^T = FE(\mathbf{x}) \tag{2.2}$$

where $FE(\mathbf{x})$ denotes the feature extraction operation. The elements of the feature vector are called *signal features* or *parameters*. The feature extraction function FE is in fact a number of K functions, each of which extracts a separate feature value $v_k = FE_k(\mathbf{x})$ where $k \in \{1; 2; ..; K\}$. The feature values and the feature vector are also considered random variables. However, the feature extraction operation should be precisely defined mathematically to ensure repeatable non-deterministic calculation of parameters. It means that if the exact same set of samples is offered for feature extraction twice, the results should be identical in both cases.

The signal features, which are used for detection or classification, are essential in the recognition process. The purpose of the features is to mathematically express the qualities which humans attribute to known classes of sound, thus enabling us to discern

between them. A whistle, e.g., is highly tonal, high-pitched, and has a very narrow frequency spectrum (close to a simple tone). Expressing these qualities in terms of signal features would lead to formulation of the following parameters: periodicity, fundamental frequency and spectral spread. By feeding these features into the classifier, it would be possible to distinguish the whistle sound from, e.g. a motor noise, which is atonal, bassy and broad-banded. In practice, however, the choice of features is seldom that straightforward. This simple example also does not take into consideration the temporal features of sound, such as signal envelope or transients.

The features are typically calculated employing short-time (possibly overlapping) sample frames, which is referred to as the *bag-of-frames* approach (a term derived from the *bag-of-words* methods known in the domain of natural language processing [45]). In such approach the input samples are divided into a number of frames of fixed length (typically ca. 100 ms, which is longer than the frame used in speech recognition). The features are calculated from each frame and fed into the pattern recognition algorithm. In case of the detection-by-classification approach, the samples are constantly fed into the classifier (e.g. Gaussian Mixture Model), which performs both the detection and classification of acoustic events [42]. When the *detection-and-classification* technique is employed, only the fragment of the input signal, which contains the foreground event, is subject to feature extraction. It is also possible to treat the acoustic event holistically and extract the features from the entire acoustic event, instead of the short-time frames. However, due to the often non-stationary character of the signal representing the acoustic events and long durations of the events (even up to a few seconds), such an approach often yields unsatisfactory performance.

2.3.1 Features taxonomy

The features used in sound event recognition are often adopted from other domains such as speech and speaker recognition or music information retrieval. The features can be divided into the following categories:

- temporal features - are extracted from the time domain representation of the signal. They often reflect the shape of the waveform or the simple qualities of the sound wave. The most widely used features include: Zero Crossing Density (ZCD), Temporal Centroid (TC), High Zero Crossing Rate (HZCRR), Low Short-Term Energy Ratio (LSTER), Log Attack Time (LAT) [40, 42, 46, 47].

- spectral shape descriptors - whose purpose is to reflect the shape of the power spectral density (PSD) function. The features in this group provide rich information about the spectral content of the analyzed signal. The most commonly employed features of this group are: spectral kurtosis, spectral slope, spectral roll-off, Spectral Flatness Measure (SFM) and spectral moments, including Audio Spectrum Centroid (ASC) and Audio Spectrum Spread (ASS) [46–48].

- spectro-temporal features - it is a rather new trend to utilize features which simultaneously capture the spectral and temporal properties of sounds. Such features can be derived e.g. from the spectrogram [27, 37] or cochleogram [49]. Another example is the time-frequency distribution utilized by Ghoraani and Krishnan [50].

- Mel-frequency cepstral coefficients (MFCC) - adopted from speech recognition, the MFCC descriptors provide the information about the spectral envelope of the signal in the transformed logarithmic mel-scale frequency domain [46, 51]. The MFCC features are successfully used in various sound event recognition applications [25, 34, 38, 40, 42, 52–55].

- Linear Predictive Coding Coefficients (LPCC) are the coefficients of the autoregressive filter whose characteristics matches the spectral envelope of the signal spectrum. The LPC analysis, originating from speech processing, is used by some researchers for discerning between acoustic events [43, 53]. The LPCCs are obtained by performing linear prediction analysis of the signal in short time frames.

- other types of features can also be derived from various representations of the signal, Discrete Wavelet Transform (DWT) [54], log-filterbank representation [35, 55] or autocorrelation function [42].

A great number of audio features are defined in the MPEG-7 standard [48, 56]. It covers both temporal and spectral descriptors, e.g. temporal centroid, spectral flatness, spectral envelope etc. The researchers often use the MPEG-7 descriptors as state-of-the art parameterization method [34, 40, 42]. To take into account the time variation of the signal features, in the *bag-of-frames* approach, the *delta* and *delta-delta* features are often used [35, 38, 53, 54]. The delta features are obtained by computing the difference of the parameter value in the current and previous frame (first order derivative). The delta-delta (or *acceleration*) features are calculated as the difference between delta features in the current and previous frame (second order derivative).

It is a subject for discussion which types of features are more appropriate for the task of acoustic event detection. On one hand, it is known that the human auditory system is sensitive to the spectral characteristics of sound. The human ear has been proved to react to formants or critical band stimulation. Thus, the popularity of MFCC features. On the other hand, the differences in time-domain representations of the hazardous acoustic events are very apparent. Consider a gunshot and a scream sound. The different temporal qualities of such events can be easily reflected by temporal features. As far as typical spectral descriptors are concerned, it may seem that they are more suitable for recognizing music sounds. However, studies show that they also provide important information in the process of discerning between different classes of acoustic events.

2.3.2 Feature selection

In a typical machine learning task the initial feature vector contains a large number of coefficients. It is because often there is no *a priori* knowledge of which parameters are best for the recognition task. Therefore, at some point in the process of creating a recognition system, one has to deal with a set of observations, which are scattered in a highly dimensional space (each containing tens or even hundreds of features). This problem, commonly referred to as the *dimensionality curse* leads to increased computational cost and data storage requirements, but most importantly - to overfitting of the classifier [57]. Hence, it is beneficial to reduce the dimensionality of the feature space by employing the so-called *feature selection* methods. The aim of all feature selection

algorithms is to chose a subset of the initial feature set, which maximizes the result of classification. Two approaches are present in the literature: *filter* methods and *wrapper* methods [58, 59].

In the **filter** approach the statistical intrinsic qualities of the parameters are examined, regardless of the classification algorithm employed. A universal metrics is employed to assess the discriminative power of each feature individually. The χ^2 statistics, information gain or Fisher measure are often used to that end [58–60]. Subsequently, ranking methods or space search methods are employed to choose the subset of the feature space [60]. The advantage of filter methods is low computational complexity and that they can be used with any classification algorithm. The drawback is that the individual discriminative power of each feature is only assessed. In practice, it is possible that even though some features are not statistically important, their combination contributes to the learning or classifying process.

In the **wrapper** methods the specific classifier is used. The features are ranked by assessment of their direct performance in the classification task [61]. Typically, in an iterative manner, several subsets of the feature set are used to train the classifier, and the evaluation on the training set is used to compare these subsets. Such approach is highly time-consuming, but it is well-suited to the particular recognition task. To facilitate the computations, heuristic methods are employed, such as forward/backward search, simulated annealing etc. [61].

The alternative approach to the dimensionality reduction is to transform the feature space into another space, in which the features are decorrelated and their number reduced. Such algorithms as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) are often employed [57]. The approach is especially profitable when the input is *raw* data, such as the coefficients of DCT (Discrete Cosine Transform) or DWT (Discrete Wavelet Transform) of the training signals. It is also worth noting that some classifies, such as SVM, have a form of feature selection embedded in the learning process. Therefore, advanced dimensionality reduction can be considered superfluous in such cases. Guyon et al. also proposed a method for feature selection based on the

weights assigned to each feature in the SVM optimization procedure [62]. A number of dimensionality reduction techniques were reviewed in a study by van der Maaten [63].

2.4 Detection methods

In case of online acoustic event recognition most of the audio data do not concern any acoustic events. It is referred to as an acoustic background and comprises typical sounds which can be encountered in a given environment (e.g. vehicle sounds in urban area, cocktail-party noise in crowded public spaces, natural sounds in rural surroundings). The task of discerning between the foreground events and the acoustic background is referred to as detection. The detection problem can be approached in two ways:

- threshold methods;
- detection-by-classification.

In the next subsections, these approaches are explained. Moreover, the issue of adaptive detection is addressed.

2.4.1 Threshold-based detection

The other approach is to compare a selected signal feature (or a group of features) with a threshold. The approach is less computationally demanding than detection-by-classification. The threshold methods originate from the Voice Activity Detection (VAD) algorithms utilized in telephony [64]. In the simplest approach the detection parameter is the signal level [53, 65, 66]. Such a straightforward approach, however, relies strongly on SNR. It is virtually impossible to detect the events whose level is lower than the noise floor. A more sophisticated choice of features can aid to abate this difficulty. Example features used for threshold-based detection of acoustic events are voicedness [67], variance of signal power [65] or DWT features [54].

2.4.2 Detection-by-classification

Numerous state-of-the-art solutions do not employ a detection algorithm per se, but rely on a selected pattern recognition algorithm to discern between the background and the foreground event. The statistical classifier analyzes the online audio data frames and produces decision: *event*, *noise* or *silence*. The most popular pattern recognition algorithm for this task is Gaussian Mixture Model (GMM) [38–40, 42, 43]. However, other structures are also used, e.g. SVM [35] or hybrid ANN-HMM structure [68]. The algorithm typically uses a vector of features whose number can vary from a dozen or so [42] to over one hundred [35]. The large number of features theoretically yields high accuracy, however the employed pattern recognition algorithms can be prone to false alerts. The other disadvantage of such approach is relatively high computational complexity, since every frame of audio data has to be fed into a pattern recognition algorithm. The solution is also not flexible, i.e. a separate classifier has to be trained to discern between a given class of events and the background noise.

2.4.3 Adaptive detection

The adaptation of the acoustic event detection is a feature which is not always considered in published works. It was reported in the literature that only two of the eight state-of-the-art sound recognition engines addressed the issue of adaptation [40]. The research closely related to the scope of this article was performed in the field of Voice Activity Detectors (VAD) that often employ a detection algorithm based on adaptive thresholding. A comprehensive review of VAD adaptation techniques was presented in the literature [64]. The presented algorithms were in majority based on exponential averaging.

The adaptive detection aims at following the changes of the acoustic environment for more robust separation of foreground events from the acoustic background. As far as the *detection-by-classification* approach is concerned, the adaptation can be achieved employing the capabilities of machine learning algorithms. If, during training, diverse sound examples are used (e.g. recorded in different conditions) the detection algorithm

will have the ability to work in changing conditions robustly. Another approach is to introduce an adaptation loop, in which the parameters of the employed model are adjusted [40]. In the threshold-based method, the adaptation can be achieved e.g. by adapting the detection threshold to the new background characteristics using exponential averaging:

$$T_{new} = (1 - \alpha) \cdot T_{old} + \alpha \cdot T_{curr} \tag{2.3}$$

where T_{new} is the new threshold value, T_{old} is the last threshold value, T_{curr} is the threshold calculated basing upon the current background characteristics and α is the adaptation constant, which determines the inertia of the threshold adaptation process [64]. If we define the time step of the adaptation T_s as the difference in time between *current* and previous step, then T_s/α can be understood as the averaging time constant. The adaptation formula in Equation 2.3 can also be used in the detection-by-classification approach to adapt the parameters of the Gaussian distributions in the GMM [39].

Since adaptation is a key feature of the algorithms designed in this work, close attention needs to be paid to the choice of the adaptation strategy. The adaptive threshold-based detection is less computationally demanding than adaptive *detection-by-classification*. We find its capability to follow the subtle changes of the background characteristics more suitable to the task of online real-world acoustic event detection. Another advantage of the adaptive threshold approach is the ability to start the analysis without the prior need for establishing a model of background sounds and foreground events.

2.5 Classification algorithms

In general, classification (or pattern recognition) is a problem of assigning to an observation, of unknown nature, a discrete quantity which identifies this observation as belonging to one of the known classes [69]. The observation is a vector of signal features. The feature vector, extracted according to the definition in Equation 2.2, is from

now in the chapter denoted $\mathbf{x}$. The mapping of the feature vector to a class is represented by a classification function $g(\mathbf{x})$:

$$g : \Re^k \mapsto \{1, ..., I\} \tag{2.4}$$

where K is the dimension of the so-called feature space and I is the number of recognized classes [69]. The function g is an equivalent to the mathematical structure of the *classifier*. Moreover, the classification function can be non-deterministic. The optimum classification function is sought in the training phase. If X is a set of exemplary observation vectors $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$ and Y is the set of desired responses of the classifier $\{y_1, y_2, ...y_N\}$, the training can be understood as a minimization task:

$$g^* = \underset{g:\Re^K \mapsto \{1,...,I\}}{\arg\min} P\{g(\mathrm{X}) \neq \mathrm{Y}\} \tag{2.5}$$

where $P\{g(\mathrm{X}) \neq \mathrm{Y}\}$ is the probability of error of the classifier [69] and g^* is the optimum classification function. Practically, the minimization of the error function is performed (e.g. mean squared error).

In the field of acoustic event recognition the observation comprises a vector of features of the audio signal. In the literature three structures are most widely used as classifiers:

- Gaussian Mixture Models [40, 42, 54, 70, 71],
- Hidden Markov Models [65, 68, 72],
- Support Vector Machines [70, 73–75].

Other classification algorithms are also encountered, such as: Dynamic Time Warping [25], Artificial Neural Networks [65], Nearest Neighbour [47], decision trees [76, 77], rule-based methods [78] or Learning Vector Quantization (LVQ) [25].

2.5.1 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) employs a probabilistic model (probability density function - PDF) of the signal features in each class. Provided that $\mathbf{x} = [x_1..x_k]$ is the vector value of random variable, representing the vector of k features of the acoustic event, the PDF for the considered class is expressed as a superposition of Gaussian distributions:

$$p(\mathbf{x}) = \sum_{i=1}^{M} w_i p_i(\mathbf{x}) \tag{2.6}$$

where M is the number of Gaussians used in the model, w_i is the weight assigned to the Gaussian component and $p_i(\mathbf{x})$ is the i-th Gaussian distribution [24]. This PDF is expressed by a so-called multivariate Gaussian distribution:

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma_i}|^{1/2}} e^{-(1/2)(\mathbf{x}-\boldsymbol{\mu_i})^T {\Sigma_i}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{2.7}$$

where $\boldsymbol{\mu}_i$ is the mean vector and $\boldsymbol{\Sigma_i}$ is the covariance matrix of the distribution.

During training the $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ parameters of the distributions are estimated using the Expectation Maximization (EM) algorithm and the features extracted from the example audio data [70]. The result of the EM algorithm is a PDF which best fits the given training data, according to the so-called *maximum likelihood* criterion [79]. A number of N Gaussian Mixture Models are created, where N equals the number of recognized classes. To discern between the considered class and noise (e.g. gunshot from noise, scream from noise [42]) a threshold is applied to the PDF of the considered class. To discern between multiple classes of acoustic events the *maximum a posteriori* (MAP) probability criterion is used to determine the type of the acoustic event.

The Gaussian Mixture Models are widely used in sound event recognition applications, both in the detection-by-classification approach to separate foreground events from noise [39, 40, 42] and to discriminate between different classes of acoustic events [54, 70]. They are reported to yield high recognition rate - comparable to other algorithms, such as SVM [70] or ANN [25]. Compared to HMM, they can yield similar or slightly worse

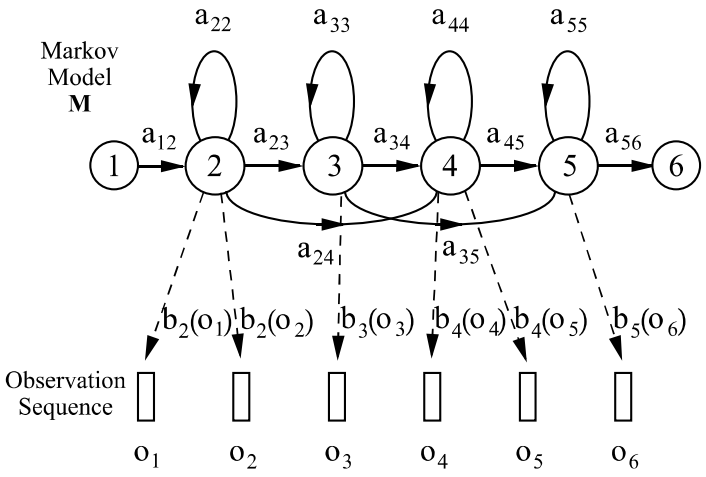


FIGURE 2.3: Example structure of a HMM [81]

accuracy, depending on the usage scenario [80]. The GMMs are strongly dependent on noise, since the addition of noise changes the values of the attributes. It is reported that the recognition rate in noisy conditions can be improved by using noisy training samples [42].

2.5.2 Hidden Markov Models

Gaussian Mixture models, described in the previous subsection, are an efficient tool to describe the statistics, i.e. probability density functions, of the parameters of acoustic events. However, they do not store any information about how these parameters behave over time [65]. Therefore, another structure is often used, which incorporates the knowledge of the dynamics of the process into the classification model, namely - Hidden Markov Models (HMM).

Markov models are a well-known mathematical tool for modelling time series. The example structure of a Hidden Markov Model is presented in Figure 2.3. The model produces a series of observations o_t, where t is a monotonous index corresponding to a point in time. In a discrete system, the observations are elements of a finite set. In sound event recognition continuous models are used, in which the calculated feature vectors are understood as observation. The model incorporates the following parameters [81, 82]:

- N states, showed as nodes in the graph. In sound event recognition the states are connected to the distinct phases in the event (e.g. attack, sustain release [65]). The initial and final states are often non-emitting, i.e. it is assumed that they do not produce observations. Typically, 3-5-state HMMs are used for sound event recognition;

- transition probabilities a_{ij} from state i to j - the transition probabilities are derived from the knowledge of the dynamics of the process. To illustrate, let us assume that a HMM models the event of breaking glass. The states of the model can be: silence, attack (knock on the glass), decay (shattering glass). The silence state will most probably remain in itself or transition into the attack state. The transition directly into the decay state will be far less likely. Moreover, the attack state will quickly transition into the decay state, which on the other hand, lasts for a longer time period and can remain in itself for some time. If the attack state transitioned back into silence, without the decay state, it would mean that the event is most likely some other impulsive event, e.g. gunshot;

- emission probabilities $b_i(o_t)$ of an observation o_t from state i. In a continuous model, which are used in sound recognition, the emission probabilities are represented as PDFs of the values of signal features which are observed in each state. It is most often expressed as a multivariate Gaussian, as defined in Equation 2.7. Therefore, it is often said that a HMM-GMM engine is employed, in which HMMs model the evolution of acoustic events and GMMs model the probability distribution of features in each HMM state.

Provided the parameters of the model are known and the sequence of observations $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T]$ is observed, the probability of this sequence modelled by the model M is given by the formula:

$$P(\mathbf{O}|M) = \sum_{\mathbf{S}} \left[a_{s(0)s(1)} \prod_{t=1}^{T} b_{s(t)}(\mathbf{o}_t) a_{s(t)s(t+1)} \right] \tag{2.8}$$

where $S(t)$ is the state of the model in time $t = 1, ..., T$ and $\mathbf{S} = [s(1) \quad s(2) \quad ... \quad s(T)]$ is the state sequence [81]. The sum in Equation 2.8 indicates summing over all possible sequences of states. This is necessary, because in the recognition task the state sequence is not known (hence - *Hidden* Markov Model). Only the observation is known and there is more than one possible state sequence which can produce this observation.

To employ the HMMs for sound event recognition, a model has to be trained to represent each known class of acoustic events: $M_1, M_2, ..., M_I$. The transition probabilities a_{ij} and emission PDFs $b_i(\mathbf{o})$ are adjusted during the training procedure. During training, the maximum likelihood criterion is used to maximize the probability of producing the training observation sequences given the established parameter models. The Baum-Welch algorithm is most often used to that end[82].

Once the HMMs $M_1, M_2, ..., M_I$, where I is the number of recognized classes, are established, the output class is simply that which maximizes the *a posteriori* probability of the observed sequence:

$$i = \underset{i \in \{1,...,I\}}{\arg\max} P\{\mathbf{O}|M_i\} \tag{2.9}$$

However, the estimation of the *a posteriori* probability, according to Equation 2.8, requires iterating over all possible state sequences, which yields a vast number of mathematical operations. Hence, several algorithms have been developed to enable efficient estimation of the most probable HMM to model the given time series (commonly referred to as HMM decoding). The most widely used is the Viterbi algorithm [81]. It is worth noting, that solving the problem in Equation 2.9 does not provide the optimum state sequence, which best models the given observation. This information, however, is not needed in the classification process.

Hidden Markov models have been used in speech recognition since the 1960s and still yield unbeatable efficiency in this domain. This mathematical structure has also been successfully adopted to sound event recognition. Many researchers report high accuracy and strong noise robustness with HMMs. However, negative opinions are also present in the literature. Cowling and Sitte claim that HMMs are not suitable for recognition of

sound events, since it is impossible to establish an *alphabet* of sound events, similar to speech alphabet [25, 27]. Dennis also points out that a narrow time window employed in HMM-based classification is a problem. The features of overlapping events (or events mixed in noise) are often treated together, which deteriorates the performance of the HMM classifier in less than ideal conditions [27]. In the opinion of the author of this dissertation, the HMM is an efficient tool for modeling the time-varying structure of the signal, but may fail to capture the general properties of sound. It is believed that the general spectral qualities of sound are more important than the temporal structure. Consider for example a breaking glass event. The temporal properties, such as the power of impact, length of the decay etc. depend on the structure of the material, the type of tool used to break the glass etc. However, beyond these temporal aspects, there is some general spectral quality that enables us to discern between a breaking glass event and any other impulsive sound. Another example is the scream event. If an HMM is used to recognize screams, there is a possibility that the Markov chain will model the content of the scream, yet the goal is to recognize a person screaming regardless of which word they are shouting.

2.5.3 Support Vector Machines

Support Vector Machine (SVM) is a binary classifier, which discriminates between two classes by creating a hyperplane in the k-dimensional feature space [83]. The separation of an example 2-dimensional space by a hyperplane (here: a line) is depicted in Figure 2.4. In the case in which the data are linearly separable, the data vectors $\mathbf{x}_i$ and their corresponding decisions y_i are assumed to satisfy the following condition:

$$y_i = \begin{cases} 1 & \text{for } \mathbf{w} \cdot \mathbf{x}_i + b \geqslant 1 \\ -1 & \text{for } \mathbf{w} \cdot \mathbf{x}_i + b \leqslant -1 \end{cases} \tag{2.10}$$

where $\mathbf{w}$ is the weight vector normal to the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ [84]; the index i corresponds to the number of observation vector; and the coefficient b is the constant term of the hyperplane equation $\mathbf{w} \cdot \mathbf{x} + b = 0$. Thus, it is assumed that the positive data lie on the one side of the hyperplane and the negative data - on the other. The

data vectors which lie on the hyperplanes $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$ are referred to as *support vectors*.

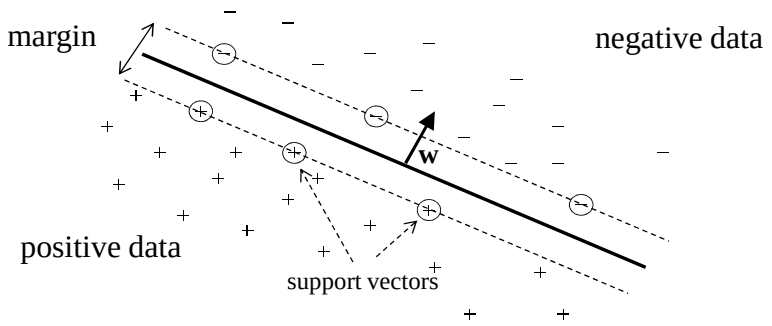


FIGURE 2.4: Separation of negative and positive data by a hyperplane in SVM method

In the training phase the margin between the support vectors and the hyperplane is maximized. It can be shown that this margin equals $\frac{2}{\|\mathbf{w}\|}$ [84]. Hence, the optimum hyperplane is defined by the weight vector:

$$\mathbf{w}^* = \arg\min \|\mathbf{w}\|^2 \tag{2.11}$$

which satisfies the constraint 2.10. This optimization is achieved by solving the Lagrangian formulation of the problem:

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b\right) + \sum_{i=1}^{l} \alpha_i \tag{2.12}$$

where l is the number of constraints and αi is the i-th Langrangian multiplier. The problem in (2.12) can be transformed into a dual problem, which is easier to solve:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.13}$$

in which $i, j \in \{1, 2, ..., l\}$ where l is the length of the problem. The details of this solution have been extensively studied in the literature [73, 84, 85].

In the case where the data are not linearly separable in the current feature space, a kernel function is applied to map the data into a new space with higher dimensionality:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{2.14}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two data vectors and Φ is the transformation of the data vectors into a different feature space [84]. In the Lagrangian dual problem, the data vectors are only present as dot products between two vectors (see Equation 2.13). Thus, if we replace these products with the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, the optimization is done on the mapped data vectors $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$. Applying the kernel trick to all the data vectors leads to solving the SVM problem in another feature space. If the data are inseparable in the linear space, transforming them into a nonlinear space usually enables separation of the data with an optimum hyperplane found in the nonlinear space. The most frequently used kernel functions mapping the data vectors to non-linear spaces are:

- Polynomial function with degree d

$$K(\mathbf{x}, \mathbf{y}) = (\gamma \cdot \mathbf{x} \cdot \mathbf{y} + \delta)^d \tag{2.15}$$

- Radial-Basis Function (RBF)

$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|^2} \tag{2.16}$$

- Sigmoidal function

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \cdot \mathbf{x} \cdot \mathbf{y} + \delta) \tag{2.17}$$

The symbols γ and δ in Equation 2.15-2.16 correspond to the parameters which alter the shape of the kernel functions and influence the classification process. More details concerning kernel functions can be found in the literature [83].

When dealing with real data, it is hardly possible to linearly separate 100% of the data vectors, in any space. The solution is to allow some points to be left on the wrong side of the hyperplane and penalize them by introducing the *cost* parameter (C) into the optimization procedure. The higher the C parameter, the less smooth is the separating

hyperplane. The γ parameter influences how much the classifier fits to the single sample. In general, high values of C and γ lead to overfitting of the classifier.

Since SVM is by definition a binary classifier, a modification has to be introduced to recognize more than one class. In the case of recognizing $M \geq 2$ classes the following methods can be used [86]:

- one-vs-all - in which M classifiers are constructed. For each classifier, the class $m \in \{1..M\}$ is treated as positive and the remaining class is treated as negative data. The winner is the class, which yields the highest output of the classification function;
- one-vs-one - in which $M \cdot (M-1)/2$ classifiers are trained - one for each pair of classes. The winner is the class which was most frequently the winner of one-vs-one comparison.
- other methods, e.g. pairwise coupling of SVM outputs [86] or rooted graphs of SVM classifiers [85].

The main advantage of is little computational cost, especially compared to HMMs. Another strength is that, unlike many other pattern recognition algorithms, SVM training always leads an optimum solution, provided the model is convergent. As far as accuracy of acoustic event classification is concerned, SVMs are reported to yield equal or better performance than HMMs [87].

2.6 Localization of acoustic events

The term *localization* concerns the calculation of the position of the sound source, based upon the received signals. It is directly related to determining the acoustic *Direction of Arrival* (*DoA*). The localization is often treated as a complimentary part of the event recognition engine [42]. Apart from surveillance, localization is also exploited in such fields as military, robotics, speech recognition or gaming systems. There is a multitude

of methods for sound source localization described in the literature. They can be divided into the following groups, based on the configuration of transducers employed:

- sensor networks,
- transducer arrays,
- sound intensity measurement.

2.6.1 Sensor networks

The sensor network approach utilizes a net of spaced transducers nodes, connected by a transmission medium (recently: wireless). Due to wave propagation, the acoustic signals are received at spaced sensors and different time and with different energy. Thus, the signals received at each sensor are compared by the means of time of arrival (TOA), angle of arrival (AOA) or received signal strength (RSS) [88]. Next, the position of sound sources is obtained using the maximum likelihood criterion [89].

The main drawbacks of the sensor network approach are low noise robustness, complicated and costly infrastructure and dependence on terrain. However, the technique has been successfully used both in research and in commercial applications (e.g. in the ShotSpotter system [90]).

2.6.2 Transducer arrays

It is a common practice to use transducer arrays (microphone arrays) for the localization of sound sources. One of the most popular methods is called the *Time Difference of Arrival* (TDOA) [91]. It exploits the time differences in the arrival of the sound wave at the spaced microphones. Generalized cross-correlation can be used for estimating the TDOA, as shown by Stachurski et al. [92] The methods based on interaural level difference (ILD) and Head-Related Transfer Functions (HRTF) are also present [93].

The physical limitation of the microphone array methods is related to the wavelength of the analyzed signal. If the distance between the microphones is comparable to the

wavelength, the localization is no longer precise. The next approach is free of such drawback.

2.6.3 Sound intensity measurement

The spherical wave field in open space propagates according to the following solution of the wave equation [94] - for acoustic pressure p_a:

$$p_a(r,t) = (A/r)exp\,[i(\omega t - kr)] \tag{2.18}$$

where A is the complex amplitude, r is the distance from the point source, ω is the angular frequency and k is the wavenumber; and for particle velocity u [94]:

$$u(r,t) = (A/\omega\rho_0 r)(k - i/r)exp\,[i(\omega t - kr)] \tag{2.19}$$

where ρ_0 is the density of fluid particles measured at equilibrium. The sound intensity vector is defined as:

$$\mathbf{I}(t) = p_a(t) \cdot \mathbf{u}(t) \tag{2.20}$$

It can be shown, that the *active* (i.e. real) component of the sound intensity vector is proportional to the gradient of the phase [95]:

$$\mathbf{I} = -\frac{|p_a|^2}{2\rho c}\frac{\nabla\Phi}{k} \tag{2.21}$$

where c is the propagation speed. It means that $\mathbf{I}$ is perpendicular to the surfaces of equal phase, i.e. the wavefronts [95]. Thus, the direction of the sound intensity vector is identical with the acoustic direction of arrival. In practice, the time averaged intensity vector is used rather than its time-varying form [95]:

$$< \mathbf{I} >_t = \int_{t_1}^{t_2} p_a(t)\mathbf{u}(t)dt \tag{2.22}$$

where $<>_t$ denotes average value over time.

The monograph of Fahy [94] and the book chapter by Jacobsen [95] cover a multitude of approaches to measurement of sound intensity. The intensity measurement is substantially different from measuring acoustic pressure, which can be done with a single microphone. In addition to acoustic pressure, particle velocity has to be measured precisely at the same point. Three methods for such measurement can be distinguished [95]. In the $p-p$ approach the intensity is determined by analyzing the pressure gradient from two microphones. In the $p-u$ strategy a specific sensor is employed to measure particle velocity, apart from a typical pressure sensor. Finally, the $u-u$ method relies on estimating the acoustic pressure from the divergence of the particle velocity.

An apparatus for sound intensity measurement is commonly referred to as a *sound intensity probe* or *acoustic vector sensor*. Currently, the leading sensor is manufactured by Microflown [96, 97]. It follows the $p-u$ approach, comprising a pressure sensor and three orthogonally placed particle velocity sensors. It has been shown in related research that the Microflown sensor can be successfully used to measure noise [98] or to localize the sounds on the battlefield [99]. The acoustic vector sensor is also applicable to detection of threatening sounds for surveillance purposes [6, 17]. It is shown in experiments by Kotus and Czyzewski that the accuracy of DoA estimation employing the intensity probe is very good - error in moderate noise conditions counts in single degrees [100]. The advantages of the vector sensor, compared to the traditional microphone array or sensor network approach is the incomparably smaller size of the apparatus, greater accuracy, and lack of wavelength limit. Hence, the intensity probe is a favorable instrument for the sound source localization.

2.7 Measures of detection and classification accuracy

In the process of evaluating the acoustic event recognition engine, the metrics which reflect the system's ability to correctly detect and classify the events need to be defined. In this chapter we define the metrics which are suitable for the evaluation of the designed algorithms, which is provided later in the dissertation. These metrics are also encountered in the state-of-the-art work, regardless of the classification algorithm employed and event type detected.

2.7.1 Detection metrics

The following metrics are the most frequently used in evaluation of the task of detection of acoustic events, similarly as in the evaluation of detection of visual events:

- True Positive Rate - TP - represents the ratio of correctly detected events; an event is counted as a true positive detection if it was present in the signal and it was detected in the correct time;

$$TP = \frac{number\,of\,correctly\,detected\,events}{number\,of\,all\,present\,events} \tag{2.23}$$

- False Positive Rate - FP - also referred to as *false alert rate* (FAR) - represents the ratio of all events which were wrongfully detected, i.e. those which were not present in the signal, yet the detection was indicated;

$$FP = \frac{number\,of\,incorrectly\,detected\,events}{number\,of\,all\,detected\,events} \tag{2.24}$$

- False Negative Rate - FN - also denoted *miss probability* or *false rejection rate* (FRR) - represents the ratio of events which were not detected, although they were present in the signal

$$FN = 1 - TP = \frac{number\,of\,missed\,events}{number\,of\,all\,present\,events} \tag{2.25}$$

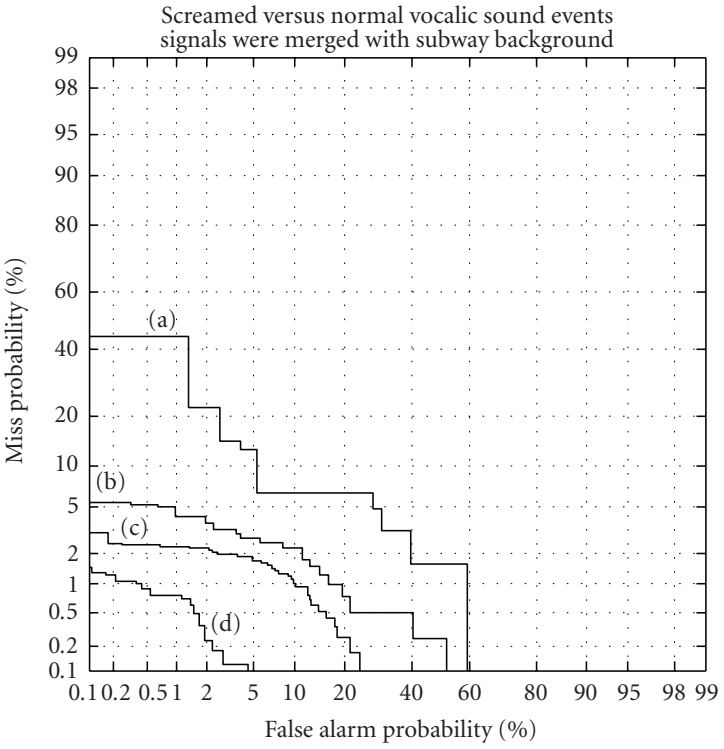
As in all detection tasks, the TP and FP rates influence one another in a manner referred to as the *detection error tradeoff* (DET) problem [101]. If the TP rate is increased, e.g. by lowering the threshold of the detection algorithm, FP rate is bound to rise. On the other hand, if the threshold is elevated, the FP rate will drop, but the rate of TP detections will also be lowered. The tradeoff is often depicted by the so-called DET curve [101]. The DET curve is a plot of the relation between the false positive detection rate (or, more often, false rejection probability) and the false positive detection rate, in which the sensitivity of the detector is a variable parameter. An example DET curve is presented in Figure 2.5. It shows the results of detecting screams in a noisy environment. The miss probability and false alarm probability are expressed in logarithmic scales, which improves the clarity of the plot. Several lines are drawn, each for a different experiment condition. Such a presentation can be used to compare the performance of the algorithm in varying conditions (e.g. SNR, type of noise) or with different parameters or methods. The line which is closest to the center of the coordinate system is considered the best, since it provides the highest TP rate and the lowest FP rate at the same time. The DET curve is a variation of the Receiver Operating Characteristics (ROC) curve known from the domain of decision systems and telecommunication [102].

Another important measure is *equal error rate* (EER). The EER metrics is related to the point in the DET plot, in which the false rejection rate and false alert rate achieve the same value. The smaller the EER, the better the algorithm's ability to correctly detect acoustic events.

2.7.2 Classification metrics

In decision systems theory loss is often considered as a measure of how many errors the classifier produces. In Bayesian theory for a two-class problem the *0-1 loss* is defined as: [103]

$$L_{0/1}(\mathbf{x}, y) = sgn(-y \cdot g(\mathbf{x})) \tag{2.26}$$



(a) SNR = −5 dB, EER = 0.0646
(b) SNR = 0 dB, EER = 0.0329
(c) SNR = 5 dB, EER = 0.0209
(d) SNR = 10 dB, EER = 0.0093
(e) SNR = 15 dB, EER = 0.0005

FIGURE 2.5: Example detection error tradeoff curve [40]

where $\mathbf{x}$ is the observation vector, $y \in \{0; 1\}$ is the desired label and $g(\mathbf{x})$ is the classifier's decision function. Another approach to loss estimation is to assign the costs to particular errors. For example, it can be defined that false positive gunshot detection costs 100, a false negative scream costs 50, etc. Hence, the average loss can be computed as the average cost of classification errors [104].

Most often, loss can be used interchangeably with other metrics. For instance, it can be noted that the overall 0-1 loss of a classifier equals 1 minus its accuracy. Therefore, throughout the thesis, the term *loss* will be used to describe the classifier's performance, yet other measures, i.e. accuracy F1 score or Kappa will be used to evaluate the classification. Hence, the following classification metrics are defined which will serve as a measure to estimate the *loss* achieved in the examined decision system.

- confusion matrix - provides the most complete information about classification results. An example of a confusion matrix, with 3 classes, is shown in Figure 2.6.

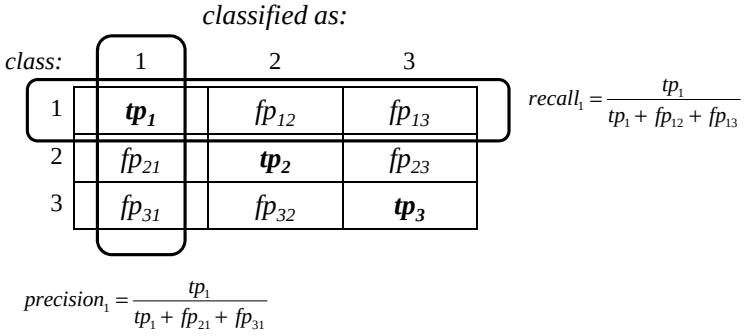


FIGURE 2.6: Example confusion matrix and formulae for recall and precision

All confusion matrices featured further on in the dissertation are formatted accordingly. Rows correspond to events which actually belong to a given class and columns pertain to the assigned classes. The elements on the main diagonal are the numbers of correctly classified instances in each class - denoted tp_i where i is the class index. The elements outside the main diagonal are the numbers of incorrect classifications. They are denoted fp_{ij} and indicate the numbers of elements which belong to class i, but are erroneously assigned to class j;

- recall - denotes the ratio of events that belong to a given class and were correctly classified; it is also referred to as the *sensitivity* of the classifier;

$$recall_i = \frac{number\,of\,correct\,classifications\,in\,class\,\boldsymbol{i}}{number\,of\,all\,events\,belonging\,to\,class\,\boldsymbol{i}} = \frac{tp_i}{tp_i + \sum_j fp_{ij}} \tag{2.27}$$

- precision - expresses the ratio of correct classifications in all events assigned to a given class; it is also called the *specificity* of the classifier. The higher the precision rate, the more certain the decision of the classifier;

$$precision_i = \frac{number\,of\,correct\,classifications\,in\,class\,\boldsymbol{i}}{number\,of\,all\,events\,assigned\,to\,class\,\boldsymbol{i}} = \frac{tp_i}{tp_i + \sum_j fp_{ji}} \tag{2.28}$$

- accuracy - is a metrics indicating the ratio of correctly classified elements from all classes. It is typically expressed in %;

$$accuracy = \frac{\sum_i tp_i}{\sum_i tp_i + \sum_i \sum_j fp_{ij}} \cdot 100\% \tag{2.29}$$

- F-score (or F-measure) is computed from precision and recall according to the formula:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision + recall)} \tag{2.30}$$

Where β is a non-negative real number and the higher it is, the more emphasis is put on recall. For $\beta = 1$ the F1 score is obtained, which equals the harmonic mean of precision and recall and is frequently used in this dissertation.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.31}$$

- Cohen's κ statistics can be used to validate the confusion matrix as a whole, instead of evaluating each class separately. The coefficient can be used to assess the agreement of two raters [105]. In the case of classification the two raters considered are actual and predicted class. The formula for κ is given by the equation:

$$\kappa = \frac{n_a - n_e}{N - n_e} \tag{2.32}$$

where n_a is the number of agreements (elements on the main diagonal of the confusion matrix), n_e is the number of agreements by chance and N is the total number of classified objects. The number of agreements by chance for class i is calculated as:

$$n_e(i) = \frac{1}{N} \sum_{j=1}^{I} c_{ij} \cdot \sum_{j=1}^{I} c_{ji} \tag{2.33}$$

where c_{ij} is the element of the confusion matrix located in row i, column j.

2.7.3 Classifier validation methods

To validate a classification algorithm, two sets of observation vectors have to be established:

- training set - which is used in the training procedure, when the classes of observations are known to the classifier;
- test set - which is used to validate if the classifier can correctly predict new data.

Provided that a set of N observations is available, the following protocols can be used to validate the classifier [65]:

1. Testing on the training set - in this case the training set and the test set are identical and both contain N vectors. This procedure does not assess the classifier's ability to recognize unknown data, but it provides the useful information about the correctness of the training procedure or about the usefulness of the employed signal features;
2. Division into the training and testing set - the training set is composed of $x \cdot N$ randomly chosen vectors and the test set contains the remaining $(1-x) \cdot N$ vectors, where $0 < x < 1$. This method validates the algorithm's ability to recognize new data. However, the disadvantage is that not all vectors are used for testing. In a variation of this method the division is performed multiple times to limit the influence of randomness on the evaluation results and to include more vectors in the testing set.
3. Cross-validation - in this method the operation of dividing the set into training set and test set is repeated k times. In each iteration N/k observations are used for testing in such a way, that each vector is exactly once present in the test set.

2.8 Real-world sound event recognition

It is worth noting that most of the published work in the field of acoustic event recognition is based on experiments on a database of recorded signals, often containing isolated events. One approach is to utilize an open database of benchmark signals, such as CLEAR (Classification of Events, Activities, and Relationships evaluation campaign) [35]. Some of the published results are obtained from the analysis of signals recorded by

the authors themselves [43, 49, 53]. Another approach is to use the signals from films, radio or sound effects libraries [38, 42]. The task of recognizing the recorded events is much simpler than online detection. In case of online detection most of the input audio data constitute the acoustic background and is easily confused with threatening evetns, thus leading to *false alarms*. The task of online detection is sometimes referred to as *real-world* acoustic event recognition [52, 68, 80]. In real-world recognition the noise added to the signal is a crucial factor, which influences the recognition accuracy. Hence, many researchers evaluate the dependence of the classifier's performance on Signal-to-Noise Ratio (SNR). In case of outdoor propagation other phenomena are also present which influence the characteristics of sound. Works of Embleton [106], Attenborough [107] and Berengier [108] provide exceptionally good reviews of the matter. In general, the following phenomena can be identified:

- **loss of energy with distance** - according to the inverse square law, sound intensity, and thus also sound pressure level decreases by 6 dB with doubling of distance [106]. The consequence of this rule is quite intuitive - the events which are farther away, yield lower SNR.

- **sound absorption in air** - some of the energy of the acoustic wave in scattered on the molecules of gas [106]. The effect is more prominent for higher frequencies. As a result, sounds recorded from a distance have different spectral features than the close ones, which can significantly influence the recognition accuracy.

- **reflections from the ground** - it is shown in the literature than the sound characteristics is subject to significant change depending on the distance between the microphone and the ground. In some cases, interference can occur, which leads to colorization of sound, again influencing the spectral features [108].

- **change of propagation speed with temperature** - the gradient of propagation speed in the horizontal or vertical direction leads to refraction of the wave [108]. This phenomenon can have an impact on the results of localization of the acoustic events.

- **dispersion of sound in air** - differences of propagation speed due to frequency are also apparent. The impulsive sound become more and more smeared as distance increases. This is one of the reasons that a gunshot or explosion heard from afar sound differently than those heard nearby. Changes in the temporal features of sounds make it difficult to robustly recognize an impulsive event from a distance.

Obviously, the above list is not complete. The outdoor sound field is very complex and difficult to predict, due to a multitude of factors to consider. Numerical approaches have been made to take the propagation factors into consideration, e.g. while creating noise maps [109, 110]. In the opinion of the author of this dissertation, at the current state of knowledge it is impossible to substantially improve sound recognition accuracy by taking these phenomena into account analytically. It is due to the fact that the propagation phenomena and noise addition occur simultaneously. However, some effort should be made to allow for the distortion of sounds due to propagation. An examination of the robustness of the features against the variable distance between the source and the microphone could be considered. Also, a separate classifier model trained on close and on far events could be employed to improve the recognition rates.

2.9 Audiovisual event recognition

The methods for detecting events in audio and video data streams are well-known and described in the literature. There is also a research trend devoted to joining the two modalities for more robust analysis. The concept of joining the audio and video data stretches beyond the domain of event detection. Audio-visual fusion has been used in such fields as e.g. video indexing [111] or emotion recognition [112]. There are not many published works on audiovisual event detection, particularly in the surveillance domain. In this section most representative researches illustrating the two prevailing approaches are listed.

One of the approaches is to join the acoustic and visual modalities at the feature level, which is referred to as *early fusion*. An example of such work is described in a paper by Cristani et al. [113]. The histogram features from the image are concatenated with

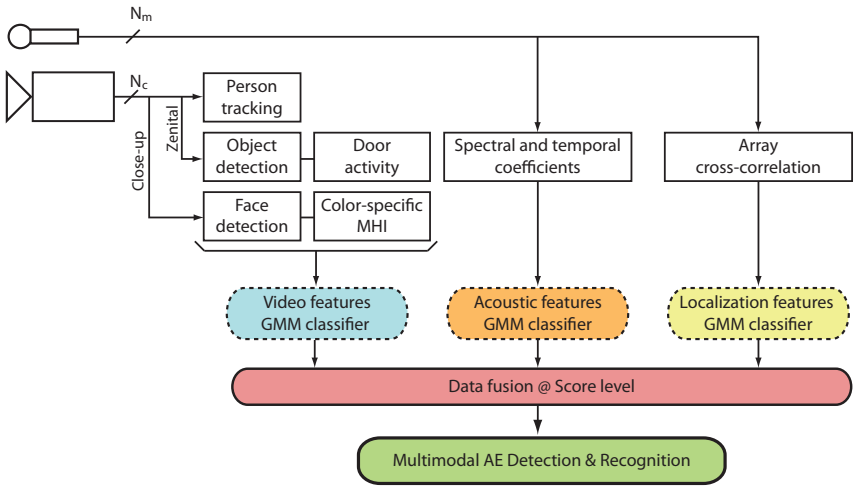


FIGURE 2.7: Diagram of the multimodal event detection system proposed by Canton-Ferer et al. [115]

the acoustic spectral parameters and form an *Audio-Video Concurrence* matrix, which is fed into a k-NN classifier. Everyday events were considered such as making/receiving a phone call, entering/exiting the room etc. The authors report a substantial increase in classification accuracy (ca. 20 pp.). Jhuo et al. proposed integration at an even earlier level [114]. A bi-modal codebook was constructed and the resulting audio-visual words were fed into a MKL (Multiple Kernel Learning) classifier.

Another approach is *late fusion*, i.e. to classify the data originating from each modality separately and join the information at the decision level. Such paradigm is followed by Canton-Ferrer et al. [115]. They utilized three GMM classifiers for video, audio and localization data respectively. The flowchart of the system developed by Canton-Ferrer is shown in Figure 2.7. The authors focused on meeting room events such as applause, chair moving, footsteps etc. The modalities were fused using two methods: weighted arithmetical mean (WAM) and fuzzy integral (FI). The authors observed an increase in classification performance (by means of F-score) in comparison with the baseline AED system, but not for all events. For example, the F-score for door slam is improved from 0.92 to 0.95 but for cough event no improvement is observed. The reason is probably that cough has no distinctive visual cues. The importance of each modality for event detection was also assessed. The results show that the acoustic modality is the most important one.

This brief review of the known approaches to multimodal event detection shows that significant improvement is attainable thanks to fusion of information from audio and video data streams. Such gain should also be apparent for the threatening events considered in this dissertation. The KASKADA framework, exploited in this work, provides efficient tools for multimodal audiovisual event detection (see Section 5.1). However, the topic of multimodal event detection is not in the scope of the thesis. Nevertheless, some works carried out with the participation of the thesis author concern the multimodal approach. In one of the author's publications methods for multimodal detection of traffic events were proposed [13]. The acoustic event detection methods featured in this thesis were used to detect car horn sound and video analysis was performed to detect sudden stopping of a vehicle, thus enabling automatic detection of collision. In some other works the localization data was employed to point the moveable PTZ (Pan-Tilt Zoom) camera in the direction of the detected event [7, 10, 17]. This concept is mentioned again in the dissertation in Section 7.2.4.

2.10 Review of existing approaches to sound event recognition

The section provides the review of the most representative works described in literature concerning sound event recognition. The survey is focused on the approaches to detection and classification of events, as well as features and algorithms employed. According to the author's knowledge, none of the known solutions utilizes a supercomputing platform for audio stream processing. The summary of the discussed approaches is presented in Table 2.1 at the end of the section.

Temko and Nadeau proposed a SVM-based technique for acoustic event detection in meeting room environemnts [35]. They follow a *detection-by-classification* approach by employing two SVM classifiers with the *bag-of-frames* scheme, as illustrated in Figure 2.8. The first classifier discerns between *silence* and *non-silence* classes. The second classifier recognizes 14 types of events, including speech, steps, door knocking, chair

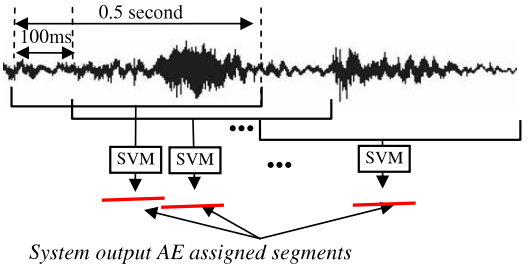


FIGURE 2.8: Detection-by-classification approach proposed by Temko and Nadeau [35]

moving, phone ring, key jingle etc. The feature vector comprises 60 features, including MFCC, log-filterbank, temporal and spectral shape features. The actual size of the feature vector equals 120, since means and standard deviations of the chosen 60 features are calculated. The research utilizes a database of meeting room recordings from the CLEAR (Classification of Events, Activities, and Relationships evaluation campaign workshop) evaluation sets and on real-life recordings from seminars. The decision system proposed by the authors is compared to a HMM-GMM-based classifier, yielding significantly better results, especially in low SNRs. The approach is similar to the one proposed by the author of this dissertation, as far as *bag-of-frames* technique and SVM classifier is concerned. However, our work differs by means of other type of acoustic events detected and different operational environment.

Zhuang et al. proposed a different approach to the problem of meeting room events recognition [68]. Their work exploits a combination of HMMs and Artificial Neural Networks. The ANN processes the signal features (derived from the spectrogram) in a *bag-of-frames* approach and outputs the probabilities for respective event classes. These probabilities are subsequently fed into the HMM engine which analyzes the events in a wider context. The authors claim that such approach, adopted from speech recognition, can boost the efficiency of event detection.

Valenzise et al. published the results of their work on a scream and gunshot detection and localization system [42]. It is an example of a straightforward *detection-by-classification* approach. Two GMM classifiers are used to recognize screams and gunshots respectively. A small feature vector is employed, comprising 13 elements for screams and 14 elements for gunshots. Feature selection is performed, starting with the initial set of 49 features, which include MFCCs, spectral shape descriptors, temporal parameters

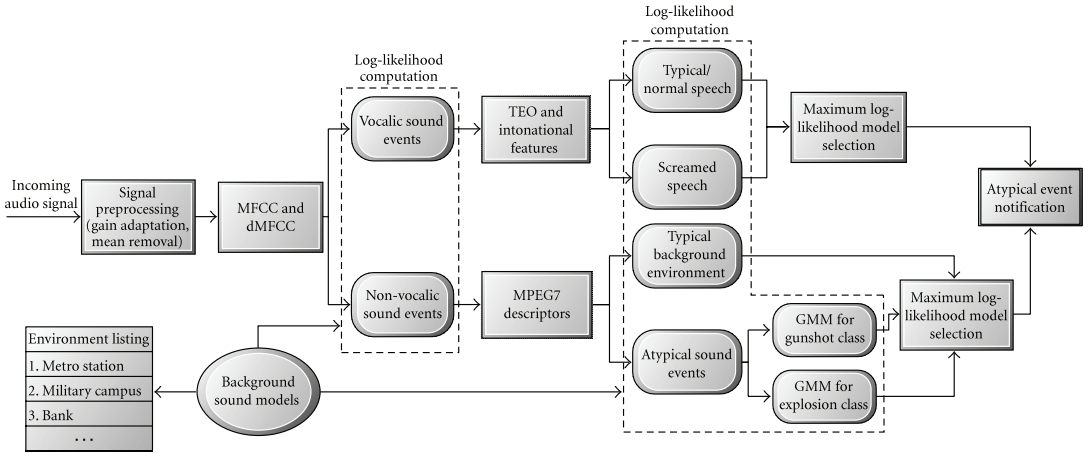


FIGURE 2.9: Architecture of the sound recognition system introduced by Ntalampiras et al. [40]

and correlation features. We find that the GMM approach to sound event detection is less flexible than the threshold-based detection proposed in this thesis. Also, the *detection-by-classification* approach is more prone to false alerts and generates higher computational load than the techniques adopted in this dissertation.

An interesting solution was proposed by Ntalampiras et al. [40]. They employ a cascade structure of Gaussian Mixture Models to discern between threatening sound events such as scream, gunshot and explosion. Separate classifiers are used to separate vocalic from non-vocalic sounds, screams from other vocalic sounds, gunshots from non-vocalic sounds etc. As far as features are concerned, MFCCs and MPEG-7 descriptors, as well as intonational features are utilized. Ntalampiras et al. try to simulate the online operation of the event recognition system. Hence, they prepare a test signal comprising a number of events, which are mixed with military, urban and metro noise. To reduce the rate of false alerts, they introduce the adaptation loop in which the probability distributions of the GMMs are modified to adapt to the background conditions. The experiments show that the adaptation contributes to a significant improvement of EER (ca. 60-70%). The adaptation of the Gaussian model inspired the choice of one of the adaptation strategies described in Section 4.1.3.

The work of Cristani et al. [39] introduces a novel approach to the detection of acoustic events. Cristani et al. propose an algorithm for detection of foreground events, without determining the type of event. An adaptive mixture of Gaussians is employed. A vector

of 8 PSD (Power Spectrum Density) features, representing the energy in logarithmically spaced frequency bands, is considered. Each feature is modeled by an adaptive GMM. The parameters of the model are updated to allow for changes in the environment. The probability obtained from the GMM and the features of the currently analyzed frame are used to determine whether the sound constitutes the acoustic background or a foreground event.

In the PhD thesis presented by Dufaux an elaborate system for recognition of impulsive sound events is introduced [65]. The recognized classes of events include door slams, explosions, glass breaking, gunshots, phone rings, screams etc. Original detection algorithms are introduced, based on threshold methods. The detection of impulsive events is performed with the use of normalized power sequence, variance or median filtering. Spectrogram features, LPC, cepstral and perceptual features (including MFCCs) are used as signal parameters. A feature selection technique based on PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) is employed. The size of the feature vectors varies from 10 (after reduction with PCA) to 16384 (for spectrogram features without reduction). Several classification algorithms are investigated: Bayesian, GMM, HMM and Multi-Layer Perceptron neural network. It is shown that a 3-state HMM classifier achieves nearly 100% efficiency for clean signals from a limited number of classes (i.e. 3, 6 or 10). The efficiency drops to 80% when noise is added to the signal (at 10 dB SNR), provided the noise is added to the training signals as well. The work of Dufaux, however wide and thorough, fails to face the problem of real-world detection, unfortunately. The experiments with a database of sounds were only presented, whereas in this dissertation it is also an aim to evaluate the performance of the sound recognition engine in practical conditions.

Rabaoui et al. [75] focus on finding robust features to discriminate between 9 classes of acoustic events, both threatening and everyday sounds (i.e. screams, gunshots, glass breaks, explosions, door slams, phone rings etc.) Perceptual Linear Prediction (PLP) features with RASTA compression are used. Also, wavelet-based features are investigated. The precise size of the feature vector is not given. A multiclass SVM is used as a classifier with 1-vs-all and 1-vs-1 approaches. The classifier is compared to a Hidden

Markov Model. It is shown that the SVM classifier outperforms the HMM. Moreover, RASTA features yield better robustness against noise than wavelet features. The detection of events is not considered, since the experiments are conducted on a database of isolated events.

In their work on audio classification and segmentation Lu et al. [47] propose a recognition algorithm discriminating between the following classes of sounds: speech, music, environmental sounds and other. The purpose of this solution is to enhance indexing of video material. A vector of 9 features is used, including temporal parameters (e.g. high zero-crossing rate HZCRR), spectral flux, band periodicity or Linear Spectral Pairs distance (LSP). The speech/nonspeech discrimination is performed with a k-Nearest Neighbour (kNN) classifier, whereas the environmental/music classification is achieved with threshold methods. In addition, speaker segmentation algorithm based on GMM is implemented. The authors report that the overall recall rate is up to 89.89% wheras the precision is up to 83.66%. Such choice of classification techniques and a rather short feature vector, raises doubts it their system would be able to robustly detect real-life events in practical conditions. We believe that more advanced machine learning algorithms (such as SVM or ANN) and larger feature vectors enable much more accurate recognition.

Tran and Li propose a novel approach to recognition of acoustic events [87]. Instead of the common feature extraction/pattern recognition scheme, they introduce probabilistic distance SVMs which serve as a measure of distance between subband temporal envelope of events. Tran and Li point out that MFCC-HMM-based engine, which is the prevailing approach in speech recognition, may not be appropriate for sound event recognition. The recognized events include: speech, music, breaking glass, explosion, cry, laugh, scream and knock. A thorough analysis of the general characteristics and subband temporal envelope of the events is given. The experiments on a database of recordings with added noise are reported. The probabilistic distance SVM approach outperforms the MFCC-SVM and MFCC-GMM approach by 1-3%, reaching a maximum efficiency in cross validation of 96.7%. It is also reported that the probabilistic distance SVM approach is

computationally effective, which favors this method in the context of online sound event recognition.

The detection of gunshots has been a popular research topic. The works of Simon, Maroti et al. are worth mentioning [116, 117]. They utilized a network of spaced sensors (in the number of tens) to detect gunshots in an urban area. Their system is claimed to be robust against reverberation and able do determine both the location of the shooter and the direction of the shot. Time difference of arrival between sensors is used to indicate the position of gunfire. The reported accuracy equals 1 meter. The sensor networks, however able of covering large areas, are costly and not as accurate as acoustic vector sensors, as far as determination of acoustic direction of arrival is concerned [97]. In the previous research conducted in Multimedia Systems Departmentan acoustic vector sensor was used to localize the events in an indoor space [6]. The utilized techniques can also be successfully used in an outdoor space. The experiments utilizing an acoustic vector sensor are mentioned in Section 7.2.4. The latency of the system introduced by Maroti et al. is said to be less than 2 seconds [116]. Our experiments described in Section 8.2 lead to a much shorter latency, thanks to employing a supercomputing cluster.

As far as the analysis of gunshot is concerned, credit is due to Maher, who published some comprehensive papers on the acoustics of gunfire [41, 44]. Although this work does not cover the recognition engine, it provides information which can be used to recognize gunshots and to improve forensic analysis of gunshot recordings. In Figure 2.10 an example analysis of a two-channel recording of a gunshot is presented. The following intervals are marked: 1) time between shock wave and its reflection (left channel) 2) offset between the shock wave in left and right channel, 3) time between shock and blast wave, 4,5) offset between blast and blast reflection in left and right channel respectively. Maher points out that the main difficulty in analyzing gunshots in realistic conditions is the influence of noise. The forensic analysis of gunshot recordings allows for assessing the distance between the shooter and the microphone and the shot angle, if two or more microphones are employed.

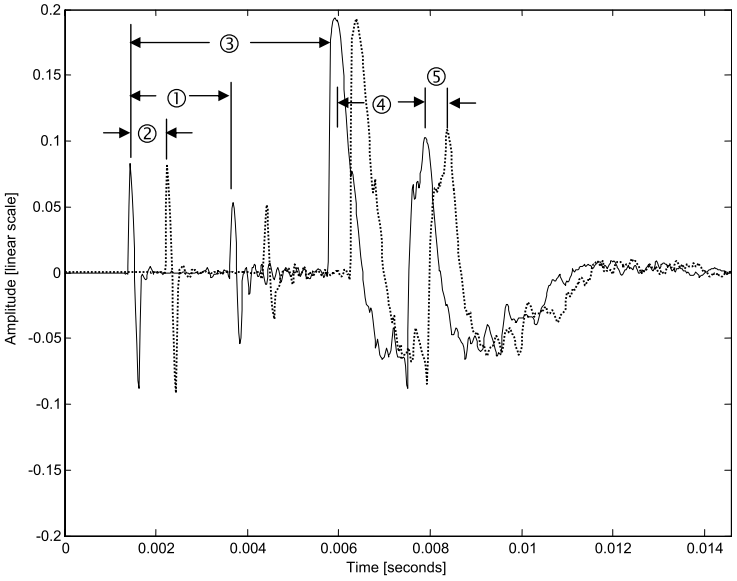


FIGURE 2.10: Two-channel acoustic recording of a gunshot as analysed by Maher [41] (solid line - left, dashed line - right)

The proof that the sound event recognition constantly expands to new platforms is the work of Nirjon et al. which features *Auditeur* platform for acoustic event detection in mobile devices [76]. Auditeur is a framework for developers, researchers and users which allows for configuring and creating sound recognition applications for smartphones. A variety of features is implemented, including temporal (ZCR, RMS, low energy features), spectral (Energy, Rolloff, Centroid, Flux) and MFCC. Moreover, classifiers such as Bayes, decision tree, GMM, ANN, SVM and HMM are available as building blocks. The audio data processing is handled by the mobile phone processor, but the system also utilizes cloud storage for exchanging data (e.g. recorded events and XML processing schemas) between users.

2.11 Existing commercial acoustic surveillance solutions

The development of the mentioned methods for sound event recognition enabled the creation of efficient practical applications. In this section the known commercial solutions which feature automatic detection and classification of acoustic events are briefly mentioned. The applications stem from the theoretical foundations described throughout the chapter. It is visible that gunshot detection systems are most popular on the

TABLE 2.1: Review of approaches to sound event recognition found in literature

Author	Reference	Recognized events	Features	Feature vector size	real-world / recordings	Detection approach	Classifier	Classification approach
Temko, Nadeau	[35]	meeting room events	log-filterbank, temporal, spectral	120	recording, real-world	det-by-class	SVM	bag-of-frames
Zhuang et al.	[68]	meeting room events	spectrogram	78	recording	det-by-class	ANN+HMM	bag-of-frames
Valenzise et al.	[42]	scream, gunshot	MFCC, spectral, temporal	13-14	recording	det-by-class	GMM	bag-of-frames
Cristani et al.	[39]	foreground events	Power Spectrum Density	8	recording	adaptive GMM	N/A	N/A
Dufaux	[65]	impulsive events, threatening, daily	spectrogram, LPC, MFCC, cepstral	10-16384	recording	threshold	Bayes, GMM, HMM, ANN	static / bag-of-frames
Ntalampiras et al.	[40]	scream, gunshot, explosion	MFCC, MPEG-7, intonational, Teager	37	recording, real-world	det-by-class	GMM	bag-of-frames
Rabaoui et al.	[75]	threatening, everyday	RASTA-PLP, J-RASTA, wavelet	N/A	recording	isolated events	SVM	static
Lu et al.	[47]	speech, music, environment	temporal, spectral	9	recording	det-by-class	kNN, GMM	bag-of-frames
Tran and Li	[87]	vocalic, music, threatening	subband temporal envelope	26-36	recording	N/A	SVM	static

market, since they have the best ability to attract a potential customer such as military or law authorities.

The ShotSpotter Gunshot Location System [90] is considered the leader on the market of acoustic surveillance. The system is based on a network of dedicated sensors, which cover a large urban area. In case a gun is fired within such area, the signals from the sensors are sent to the incident review center, where triangulation is performed to pinpoint the location of the detected gunshot. The information sent to the respective authorities includes the location of the shot, number and times of rounds fired and the direction of movement, provided the shooter is moving. The system is popular in the USA, being deployed in many cities, including Washington D.C. and Los Angeles. A report from 1998 published by the United States Department of Justice states that the ShotSpotter system is able to detect 80 % of the shots and localize 75 % of the shots with a 25-foot (7.5 meters) margin [118].

The company NetLogix introduced a Video-Integrated Gunshot Detection System (viGDS) [119]. The product is a continuation of the system priorly developed by the company SafetyDynamics (named SENTRI). It has the capability of detecting gunshots, localizing the shooter and pointing the camera in the direction of the shot. The hardware utilized in the viGDS system is presented in Figure 2.11. The manufacturer reports that the solution can recognize almost any sound, e.g. gunshot, breaking glass or human voice. The Dynamic Synapse Neural Network is used for pattern recognition. Apart from recognizing the type of event, the invention utilizes an array of four microphones to calculate the acoustic direction of arrival. It is said that the location is correctly estimated for distances from two inches to two football fields. The system is also linked with GPS and transmits the exact location of the shooter to the police.

One of the most widely known military systems for gunshot detection is called Boomerang [120]. The setup of the device is presented in Figure 2.12. According to the manufacturer, it can also be installed on a manned or unmanned vehicle. The Boomerang system detects and localizes shooters from maximum weapon ranges. It also has the capability of detecting the passing bullet and assessing its trajectory. The device comprises an

FIGURE 2.11: The hardware utilized in the NetLogix viGDS system [119]

FIGURE 2.12: Fixed site setup of the Boomerang system [120]

array of microphones and a signal processing unit, contained in a black box, which analyzes the signals from the microphones and determines the acoustic direction of arrival of the detected gunshot.

Chapter 3

Audio supercomputing

In the following sections the state of the art in parallel processing of audio data will be introduced. First, a brief theoretical introduction to the subject of parallel processing is performed. Next, in Section 3.2 the solutions employing local processors (such as GPU or DSP) are introduced. Subsequently, in Section 3.3 the existing trends in utilizing supercomputing platforms for audio analysis are outlined. The chapter is concluded with some critical remarks.

3.1 Introduction to parallel processing

The term *parallel processing* refers to executing a computer program on more than one *computing node*. The nodes may be part of one machine (e.g. multiple cores of single processors), or can be distributed on different, homogeneous or heterogeneous computers. Proficz names the following typical distributed computing architectures [121]:

- cluster computing - a cluster comprises a group of homogeneous nodes connected with a fast computer network;
- grid computing - a grid is composed of heterogeneous nodes, geographically dispersed and connected with a wide area network (typically Internet);

- cloud computing - cloud computing is oriented on delivering the computing infrastructure as a service, the computers are also heterogenous and geographically dispersed, the service provider delivers the infrastructure, software and license;
- sky computing - comprising multiple clouds and thus different software and standards providers.

The machines employed for parallel computing comply with one of the following architectures (the so-called Flynn's taxonomy [122]):

- Single Instruction Single Data - SISD,
- Single Instruction Multiple Data - SIMD,
- Many Instructions Single Data - MISD,
- Many Instructions Multiple Data - MIMD.

The above terms are also commonly used to express the manner in which a piece of computer code can be executed in parallel. According to Flynn, the MIMD approach has the highest capability of accelerating the computations, however it also has the largest demands, since both the multiple processing units and data must be available at low level [122]. In fact, the MIMD architecture is the most popular among supercomputing clusters.

The ability of a distributed system to increase the efficiency of computations by expanding the resources is referred to as *scalability* [123]. The resources can be expanded either *horizontally* by connecting additional nodes or *vertically* by improving the structure of a single node [121]. The increase in computational efficiency can be understood as *speedup* which is defined as follows [124]:

$$S(n) = \frac{T_1}{T_n} \tag{3.1}$$

where T_1 is the execution time on a single node and T_n is the time elapsed for program execution on n nodes. In case of processing multimedia data stream, it is better to

employ another metrics for scalability, such as the one proposed by Jogalekar [125]. The scalability of the cluster should be evaluated as the ratio of productivity $F(n)$, defined as follows:

$$F(n) = \frac{\lambda(n) \cdot f(n)}{C(n)} \tag{3.2}$$

where λ is the throughput (e.g. in frames per second), f is the measure of quality of service and C is the running cost per second. The scalability at scale n can then be calculated according to the formula:

$$\psi(n) = \frac{F(n)}{F(1)} \tag{3.3}$$

Such criterion is employed by Proficz in his doctoral thesis to evaluate the scalability of the KASKADA platform [121].

3.2 Centralized parallel processing of audio data

In this section, the methods for parallel audio processing which exploit the local computing nodes of a given machine will be discussed. The mentioned solutions do not employ grid computing or supercomputing, which will be discussed further on.

3.2.1 Vectorization of audio algorithms

One of the basic ways to speed up the computation time in audio algorithms is to perform the operations which take vectors as input in parallel mode. This scheme is of SIMD architecture. It is worth noting that most operations executed in audio signal processing algorithms (in particular, vector multiplication, addition, dot product etc.) are highly vectorizable. It is possible to benefit from vectorized computing by using the processor's vector instructions. In case of CPUs, the instruction sets like SSE (Stream SIMD Extension) or AVX (Advanced Vector Extensions) are found in products manufactured by Intel and AMD [126]. These instructions support, among

other, addition, summing and scalar product of vectors. In case of DSPs, it is a standard to support SIMD operations, hence appropriate instructions are found in almost any digital signal processor.

3.2.2 Audio applications for GPU

Many works have been published in the recent years concerning the employment of GPU (Graphics Processing Unit) to support the parallel processing of audio data. The CUDA (Compute Unified Device Architecture) toolkit released by the leading graphics card manufacturer NVIDIA contributed greatly to the popularity of such applications [127]. The operations which are offloaded to the graphics processor are recommended to be SIMD type. Tsingos et al. provide a remarkable review of existing and possible audio applications for GPU [128]. The following can be named:

- signal processing and filtering - some of the existing audio workstations also utilize GPU apart from vector instructions. The trick to use graphics API for audio processing is to substitute pixels with time samples and RGBA (Red, Green, Blue and Alpha) color components with frequency bands. The GPU is also appropriate for performing multiplications and additions needed in FIR (Finite Impulse Response) filtration [129].

- sound synthesis - an interesting work was published by Savioja et al. concerning additive synthesis of sound [130]. A GPU was utilized to compute the sinusoidal components and thanks to the employment of parallel processing a superposition of one million sinusoids was achieved, which contributed to the good quality of the synthesized signal. Spatial synthesis is also considered by Tsingos et al. in the context of processing the sound with multiple HRTFs (Head-Related Transfer Functions) and synthesizing a complex acoustic scene [128].

- room acoustics - the Finite-Difference Time-Domain technique has been successfully employed on a GPU in a number of works [131, 132]. The acoustic ray tracing method has also been employed in a manner imitating the ray tracing performed in graphics [133].

- computational auditory scene analysis (CASA) - in another application parallel processing with the aid of a GPU was applied to source signal separation, accelerating the computationally demanding Independent Component Analysis [134].
- audio feature extraction - in a work by Schmadecke et al. [135] it was shown that employing a GPU for audio feature extraction leads to a substantial speedup. The MFCCs were considered and a speedup of up to 30 times was achieved with a 448-node GPU.

The rather new trend of employing GPU for processing audio data is clearly opening up to new applications and we can surely expect more interesting works in the near future. Tsingos et al. believe that one of such applications will be wave field synthesis or microphone array techniques [128]. It is also reported that GPU is employed for decoding speech using Hidden Markov Models [136]. This is very close to the acoustic event recognition addressed in this dissertation.

3.2.3 Other approaches

A solution closely related to the algorithms described in this thesis was proposed by Chen et al. [137]. Parallel processing on a multiprocessor CPU was employed to speed up searching for a known audio clip in a large data stream. The method is illustrated in Figure 3.1. The audio data stream was divided into chunks and each chunk was processed by a different processor. The acoustic fingerprint technique was employed to recognize the known audio data. Acoustic features (here MFCCs) were calculated from the stream and compared to the fingerprint of the sought clip with the use of a Common Component Gaussian Mixture Model (CCGMM). Chen et al. reported that employing multiple processors allows for achieving linear speedup on 2 or 4 processors and up to 11.3 times shorter computation time when 16 processors are employed.

It is shown in the literature that numerous operations concerning audio data processing can be run in parallel. Schimmel finds that in a typical PC sound processing application, i.e. a digital mixing console (included in every Digital Audio Workstation), mostly consists of time-consuming multiplications and additions of samples from numerous logical

audio channels [138]. Thanks to the proposed parallel implementation of the mixing functions the almost constant speedup is achieved for a multicore desktop CPU. The multiprocessor support for audio effects exploited in audio synthesis and editing is incorporated into the software designed by such companies as Native Intruments, Avid, Steinberg and other.

Another platform which is successfully used for parallel execution of audio data processing is a Field Programmable Gate Array (FPGA). The works of Maka and Dziurzanski are particularly relevant to the topic of this dissertation. They introduced a System-on-Chip (SoC) for parallel execution of several audio and video processing algorithms, including speech recognition [139, 140]. The methods for feature extraction and classification were implemented on an FPGA. A separate work was devoted to parallel audio feature extraction [141]. Among others, spectral descriptors, MFCCs and LPC features were considered. It was shown that the computation time was reduced by a factor of 60% thanks to parallel execution. Schmidt et al. also introduced an FPGA for acoustic feature extraction for the purpose of Music Information Retrieval [142]. MFCCs and spectral descriptors were extracted from the signal, including spectral centroid, spectral flux and spectral rolloff (which are also used in this dissertation, as shown in Section 4.3. In another relevant work the FPGA was used for acoustical simulations according to the Digital Huyghens Model for predicting the sound field within a room [143].

Another work, rather distantly related to the system presented in this dissertation, was described by Blechmann in his master's thesis [144]. An audio synthesis engine is introduced which exploits the supercomputing abilities through a specialized environment called SuperCollider. Blechmann introduced an extension to the SuperCollider framework, which allowed for parallel execution of SIMD instructions, pipelining and other mechanisms. The results show that nearly 4 times speedup is achieved when 4 processors are used.

Technologies which lie on the boundary of local and distributed processing are also present. An example of such solution is SoundGrid developed by Intel and Waves [145]. It uses the Audio-over-Ethernet technique to connect devices like mixing consoles, computers, broadcasting stations and processing servers. The processing is offloaded to

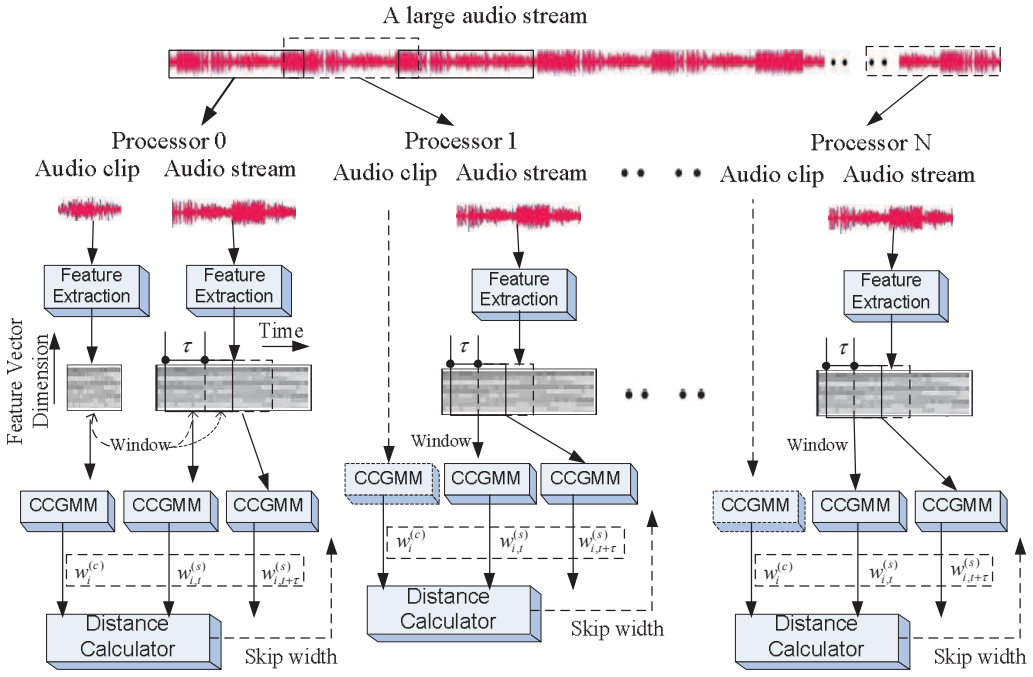


FIGURE 3.1: Parallel processing employed for audio data retrieval [137]

dedicated servers which handle the operations on audio data stream. The system yields high performance and very low latency (0.8 milliseconds) and is intended for professional live audio applications.

Interesting results have been published concerning parallel speech recognition. However it is not the topic of this dissertation, there is some connection between the field of speech recognition and sound event recognition. As it was shown in Section 2.2, some methods, including feature extraction and classification, are common. Kisun You et al. utilize an Intel i7 multicore processor (with 4 hyper-threaded cores) and an NVIDIA manycore processor (with 128 pipelines) [146]. The speedup of 3.4× on multicore and 10.5× on manycore processor is achieved. The operation which benefits the most from parallel processing as far as speech recognition is concerned, is Viterbi decoding [147]. This can be considered the contrary of the problem addressed in this thesis, in which feature extraction consumes significantly more processing time that classification (see Section 8.2).

3.3 Audio processing in distributed architectures

In the recent years supercomputing platforms have gained much popularity. Since the massive breakthrough in the 1990s more and more powerful machines have been used to accelerate computations in numerous fields, including particle physics, chemistry, genetics and meteorology. Grid computing is a considerable trend, which extends the processing power by connecting a few supercomputer clusters together in a computational grid.

Applications of supercomputing to processing audio data are not frequent. In a very early work, dating back to 1989, the Cray supercomputing platform is used to model the sound propagation in oceans [148]. In 2000 a research was reported, in which the simulation of ultrasonic field is accelerated [149]. A cluster of 24 dual-core Pentium III processors is used, which compared to the current state of the art can hardly be considered supercomputing. In another publication concerning acoustic modelling the finite element method is implemented on two clusters [150]. The first cluster consists of 10 nodes of 2 processors, whereas the second cluster consists of 8 nodes of 2 processors. The OpenMP library and MPI protocol are employed. It is also worth noting that the DEISA (Distributed European Infrastructure for Supercomputing Applications) initiative, funded in 2002, reports the research on employing supercomputers to study the noise in vehicles by means of computational aeroacoustics [151].

Music Information Retrieval (MIR) is a field which often employs supercomputing. However, in most architectures, the feature extraction operation, closely related to the algorithms described in this thesis, is performed on the client's side. The supercomputing servers handle the pattern recognition by comparing the signature of an audio file with the signatures of the files stored in the database. Relevant research was published by Jang et al. [32]. In the works of Downie and Futrelle real large scale supercomputing is employed [31]. A grid of five supercomputers is used with a total of 5000 processors and 40 TB of RAM. The computing power of the grid is estimated to reach 30 Teraflops.

Nowadays the *cloud computing* paradigm is becoming more and more popular. Such services as Google Cloud [152] or Microsoft Azure Cloud [153] facilitate the implementation

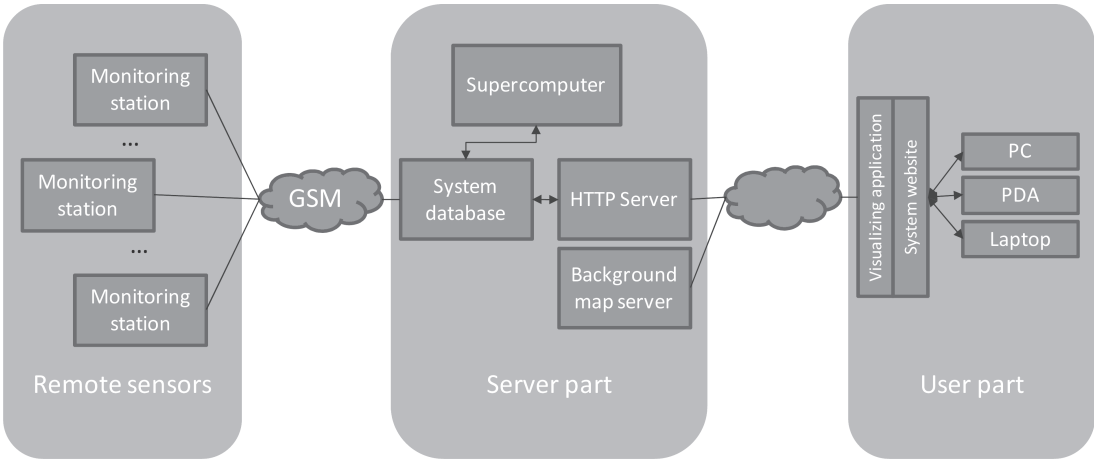


FIGURE 3.2: Architecture of the system employing a supercomputing platform for dynamic noise map creation [109]

of various algorithms in the computing cloud. In a related field Wenyu et al. implemented an audio fingerprinting algorithm in the cloud [154]. The audio fingerprinting was employed to search for an audio clip similar to the one presented by the service user.

A notable example of employing supercomputing to sound processing is the work of Czyżewski et al. [109]. A supercomputing cluster platform is used to calculate dynamic noise maps. The diagram of the system is presented in Figure 3.2. The data from noise monitoring stations located in selected places in the metropolitan area of Gdańsk are collected. Next, the measurement data are sent via GSM network to a server. To obtain the noise map for the whole metropolitan area a sound propagation model is employed to estimate the noise levels between the measurement points. The very computationally demanding operation of calculating the sound distribution in free air, considering the terrain model, is performed on a supercomputer. Finally, the HTTP server provides the user with the visualization of acoustic maps. Thanks to the employment of the supercomputing platform the dynamic noise map is refreshed in an unprecedentedly short time. The methods employed in the mentioned work are different than those used in this dissertation, however the resources of the same Academic Computer Center TASK in Gdańsk are employed, including the Galera cluster. A more detailed description of this particular supercomputing environment is provided in Chapter 5.

The "MAYDAY Euro 2012" project, carried out between 2009 and 2012 in Gdańsk University of Technology, introduces the framework for parallel processing of multimedia

data streams in a supercomputer environment [3]. The hardware platform, namely Galera and later Galera+ cluster, is located in Gdańsk University of Technology. One of the features of the developed framework (named KASKADA) is online processing of audio and video data streams [155]. During the project, with the participation of the thesis author, multimedia stream processing services have been implemented on the supercomputing cluster. Among other services, acoustic and visual surveillance services are implemented. The usage of the supercomputing cluster allows for significant increase in the efficiency of multimedia stream processing, which is proved by the results presented in this thesis.

3.4 Remarks on audio supercomputing

It is a common conviction that audio processing is not the most computationally demanding task, especially compared to video processing. However, it is shown in this chapter that there are several applications in which very time-consuming operations from the audio domain benefit greatly from parallelism (e.g. sound synthesis, room acoustics). The question in the context of this dissertation is: is sound event recognition one of such operations?

It has to be admitted that the signal processing methods which are used for sound event recognition (mentioned in Chapter 2) and outlined in Chapter 4) are not so computationally costly that they *require* parallelism to be executed online. However, parallelism is still considered in this thesis. As it is shown in Chapter 8, the advantage is *scalability* and *acceleration of decision making*. Thus, acoustic event recognition is proved to be another successful application of audio supercomputing, which is discussed again at the end of the dissertation.

Chapter 4

Developed sound recognition engine

In this chapter the sound recognition engine developed in the work is described in detail. The variety of algorithms and methods which can be used for the task of acoustic event recognition have been mentioned in Chapter 2. Basing on the literature studies the author chooses the best methods for recognizing the considered hazardous events in practical conditions. Before the specific methods employed can be explained in detail, some assumptions need to be made regarding the developed engine.

1. **Recognized events** - It is assumed that the system should recognize 4 classes of events related to danger, i.e. *explosion, broken glass, gunshot* and *scream*. The fifth class comprises other, typical sound events which do not indicate threatening situations and is denoted *other*. The chosen events represent the most frequently encountered dangerous situations. Such choice of events is also followed in many related works [42, 65, 80]. Nevertheless, the developed engine is not *limited* to the mentioned event classes. It can be easily adopted to recognize other types of events simply by adding them to the training set. In fact, an example application in which the proposed methods are used to recognize other types of sound is featured in Section 7.2.3 (bank operating hall surveillance). A deepened discussion on the recognized signals is provided further on in Section 6.1.

2. **Working conditions** - The developed engine should work online on real world data. It is one of the objectives of the thesis that the algorithms should adapt to the changing acoustic conditions. Therefore, the engine is capable of both indoor and outdoor operation. As far as outdoor operation is concerned, the system is intended to work in urban soundscape. The example indoor environment featured in the experiments is a hall in which public events take place. As it was mentioned in Section 2.8, recognizing the events in real-world circumstances is very challenging and is not addressed by all researchers working in the field of acoustic event detection.

3. **Sample format** - Throughout the dissertation we assume that the samples are provided in floating point 32-bit format and with a sampling rate equal to 48000 samples per second. All engineered algorithms work in the digital domain.

4. **Program code** - due to the implementation in online mode all algorithms are written in C++. The library LibSVM [85] is used for Support Vector Machine classification and FFTW [156] is used for computing the Fourier Transform. The sound source localization code is the work of a coworker from Multimedia Systems Department, dr. Józef Kotus. The KASKADA framework, explained in depth in Chapter 5, was created by the team of the project Mayday Euro 2012. Apart from that, the code for all algorithms, including the tools for detection, parameterization and validation of results was developed by the author of the thesis.

5. **Localization** - The localization of the sound source is treated as an *addition* to the recognition system. Hence, the methods for sound source localization are not featured in this chapter. However, some of the experiments outlined in Chapter 7 feature the localization of acoustic events. The techniques used for localizing the acoustic events are explained in the sections devoted to pertinent experiments.

The general concept diagram of the designed engine is shown in Figure 4.1. The input samples are first processed by the detection algorithm. The aim of this algorithm is to determine, whether the currently processed frame contains the acoustic background (i.e. typical sounds in a given environment) or a foreground event, which we understand as a

sound whose features are different from typical sounds. The detector's output equals 1 if a foreground event is detected and 0 otherwise. At this stage it is yet unknown whether the event is threatening or not. The frames, for which the detection algorithm yields positive values, are stored in a buffer. Next, once the event has finished and detector yields 0 again, feature extraction is performed on the buffered samples. Finally, the extracted features are fed into the classifier which determines if the event is considered hazardous and to which class it belongs.

The following sections of this chapter are devoted to the main building blocks of the engine, namely: *detection, buffering, feature extraction* and *classification.*

4.1 Detection

In Section 2.4 the approaches to sound event detection known from the literature were reviewed. The two prevailing techniques are detection-and-classification and detection-by-classification. It is more favorable to use the detection-and-classification approach in this thesis. Firstly, this approach is computationally lighter since it does not require the classifier to constantly process the input audio data. Moreover, it enables threshold-based event detection mechanism which is highly flexible. As it is shown in the following subsections, to design a new detector, the decision parameter has to be redefined only, with no need for establishing a new model (as it is required e.g. in GMM-based detection). Finally, the threshold-based detection enables computationally simple and highly efficient adaptation of detection threshold, which, as the experiments featured in the dissertation prove, contributes to the robustness of the recognition engine.

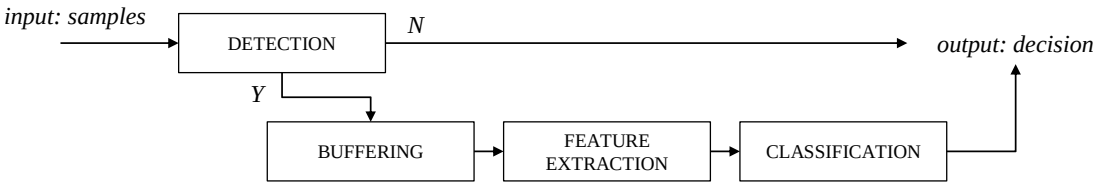


FIGURE 4.1: General concept diagram of the sound recognition engine

4.1.1 Detection principle

According to the principle known from the literature, the detection of acoustic events is performed by comparing the value of the *detection parameter* d with the threshold value t. The detection parameter can be of various nature (e.g. sound level, periodicity, kurtosis etc.) and its choice depends on the type of event which we want to focus on. If the detector should recognize e.g. short impulsive sounds, sound level calculated in short frames would be a good choice of the detection parameter. However, if the detector should respond to tonal sounds, a periodicity-related parameter would be a better choice. Nevertheless, the detection methodology and the adaptation mechanisms presented in this section are universal, as far as the choice of the feature which determines the detection is concerned. Henceforth, we will refer to the detection parameter as d. The definitions of specific detection parameters are featured in Section 4.1.2. The decision function, which yields 1 if the currently analyzed sound is a foreground event and 0 otherwise, can be defined as follows:

$$D(i) = \begin{cases} 1 & \text{if } d \geq t \\ 0 & \text{if } d < t \end{cases} \tag{4.1}$$

where i is the index of the sample frame for which the detection parameter is calculated. The length of the detection frame is also dependent on the purpose of the detector. Short frames (25-100 ms) are useful for detection of impulsive sounds whereas longer frames (up to 1 second) will yield better results of detection of periodic sounds.

The block diagram of the detection algorithm is presented in Fig. 4.2. The initial phase is learning, in which the detector is insensitive to acoustic events and gathers the profile of the acoustic background. The length of the learning phase can be adjusted depending on the characteristics of the acoustic background and the complexity of the model. The typical learning time is 30-60 seconds. Once the learning phase is completed, the detector compares the detection parameter of the current frame with the threshold value. If the threshold is exceeded, the algorithm yields a positive detection result. If the current detection parameter value is below the threshold, it is used to update the background

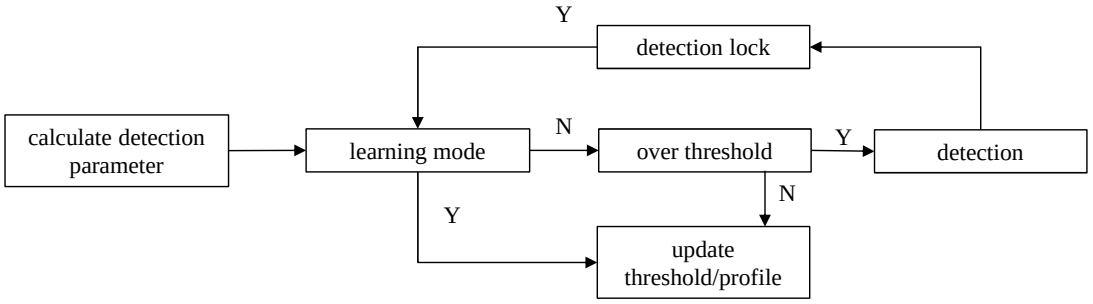


FIGURE 4.2: Block diagram of the acoustic event detection algorithm

profile, thus influencing the adaptive threshold. It is possible in this architecture that the detection parameter exceeds the threshold for too long and since no update is performed while detection is triggered, the algorithm becomes locked in detection state. To avoid this deadlock, a detection time counter is employed. If the detection is triggered for an excessive amount of time (e.g. 30 seconds), the detector goes back to learning mode in order to adapt to the new background.

The key parameter of the detection algorithm is *sensitivity*, denoted s. This parameter determines how *far* from the typical values of detection parameter the threshold should be set. The lower the sensitivity, the lower number of events events will be detected. The higher the sensitivity, the more events will be detected, but at the cost of increased false alert rate. Therefore, the sensitivity parameter influences the rates of *True Positive* (TP) and *False Positive* (FP) detections. It is defined that the sensitivity of the detector ranges from 0 to 1. How these values are translated to the value of detection threshold t, it depends on the intrinsics of the detection algorithm and adaptation approach employed. The details concerning this mapping are featured in the following subsection.

4.1.2 Detection algorithms

The choice of detection parameter is an important aspect and should be matched to the types of events we intend to detect (see the work of Dufaux for deepened discussion of the subject [65]). In a series of draft experiments several features for detecting the considered hazardous events were compared. As a result, four detection parameters and thus four detection methods were arrived at. All the described detectors are compliant

with the principle outlined in Section 4.1.1. The differences lie in the definition of the detection parameter d and threshold t, as well as on the mapping of sensitivity s to threshold value t.

Impulse detector

The *Impulse detector* algorithm is designed to detect short impulsive sounds, e.g. gunshots. The detection parameter is the equivalent sound level in a time frame containing N samples:

$$d = L_{eq}[dBSPL] = 10 \cdot log\left(\frac{1}{N}\sum_{n=1}^{N} x[n]^2\right) + L_{norm} \tag{4.2}$$

where n is the sample index and the normalization constant L_{norm}ensures that the sound level is expressed in decibels relative to 20 μPa. Throughout the dissertation $N = 512$ samples is most often used, which corresponds to 10.6 ms at 48000 samples per second. The threshold level t is obtained by adding a margin to the equivalent sound level. For sensitivity s equal to 0, the margin is 20 dB, whereas for $s = 1$, the margin equals 5 dB.

$$t[dB] = d + 20 - s/15 \tag{4.3}$$

Speech detector

The detector suited for discerning vocal sounds from the acoustic background is based on the *Peak-Valley Difference* (PVD) parameter. The parameter is defined in (4.27) in Section 4.3. It expresses the distance between the peaks and troughs in the power spectrum of the signal. The PVD parameter yields high values for periodic signals, whereas it yields small values for noisy signals, which typically constitute the acoustic background. Therefore it is a good parameter for detecting such events as screams. To obtain the threshold t the PVD value is multiplied by 2 for sensitivity s equal to 1, and by 20 for sensitivity equal to 0.

$$t = d \cdot (20 - 18s) \tag{4.4}$$

The frame, in which the PVD parameter is calculated, should be long enough to ensure adequate spectral resolution of the Fourier analysis. Typically throughout the dissertation, 4096-point-frame is used. Therefore, the temporal resolution of this detection algorithm is worse than that of *Impulse Detector*.

Variance detector

The *Variance detector* is conceived to detect sudden changes in the spectral structure of the signal. It was established in draft experiments, that the occurrence of an acoustic event leads to changes in the relation of energy in some specific frequency bands. Also in the course of the work 8 features were identified, which reflect distinctive energy ratios. Hence, in this detection algorithm we use spectral energy features $SE_1 - SE_8$, defined later on in Section 4.3, to calculate the detection parameter. The variance of the SE features is examined, defined as:

$$d_n = \frac{1}{I} \sum_{i=1}^{I} \left(SE_n(i) - \overline{SE_n} \right)^2 \tag{4.5}$$

where $SE_n(i)$ is the value of the n-th spectral energy feature in frame i, I is the number of recent frames considered, and $\overline{SE_n}$ is the mean value of SE_n from last I frames. In fact, 8 detection parameters are calculated, for 8 spectral energy features. The threshold is obtained by multiplying the current variance by 2 for sensitivity s equal to 1 and by 16 for sensitivity equal to 0.

$$t_n = d_n \cdot 2^{4-3s} \tag{4.6}$$

A sudden change in the spectrum of the signal causes the feature variance from last I frames to rise. If variance of any of the 8 examined features exceeds the threshold, the detector's output equals 1.

Histogram detector

The *Histogram detector* algorithm resembles the GMM detector [42]. The signal is analyzed in 30 1/3-octave bands. A histogram of the SPL values in each 1/3-octave band is created. The detection parameter is an estimate of the probability of the current spectrum of the signal:

$$d = -\sum_{i=1}^{30} h_i(X_i) \tag{4.7}$$

where $h_i(X_i)$ is the value of the normalized histogram for band i and for SPL value X_i in i-th 1/3-octave band. The minus sign is added for compliance with the definition 4.1. The threshold t is obtained by dividing the current detection parameter by 2 for maximum sensitivity and by 32 for minimum sensitivity.

$$t = d \cdot 2^{-5+4s} \tag{4.8}$$

4.1.3 Adaptation

It is shown in related works that adapting the detector to the changing conditions is beneficial in the context of sound recognition [40]. Several adaptation strategies were also employed in voice activity detectors [64]. In this work original approaches to adaptation of detection thresholds are featured. In general, the aim of adaptation is to automatically update values of the threshold t to match the changes of the acoustic background. The assumptions for this operation are:

- the threshold should follow the changes in the level of the detection parameter in the environment, i.e. it should be lowered when the background level of d is low and elevated when the background level of d is high;
- the threshold should react to a general trend over time rather than to instantaneous characteristics of the acoustic background, hence a time constant needs to be introduced in order to achieve smooth course of t;

- the threshold should be adapted more slowly when the background is quiet and more rapidly if there are sudden changes of d present in the background;
- the threshold should also be matched to the dispersion of d values in the environment in order to control the rate of false alerts.

From these assumptions, three different strategies for adaptation of the detection threshold are implemented.

Single adaptation

In the case of *single adaptation* the value of t is updated to match the changes in the acoustic background using exponential averaging. The initial value of the threshold is calculated as in Equation 4.3, 4.4, 4.6 or 4.8. Next, for every frame of sound samples a new value of threshold is determined by substituting d with the up-to-date value, calculated in the current frame. The current value of t is calculated as follows:

$$t = t_{new} \cdot \alpha + t_{old} \cdot (1 - \alpha) \tag{4.9}$$

The constant $\alpha \in (0; 1)$ is related to the *adaptation time* of the detector, i.e. the time constant of averaging. The adaptation time in seconds is defined as:

$$T_a[s] = \frac{N}{SR \cdot \alpha} \tag{4.10}$$

where N denotes the number of samples in the detection frame and SR is the sampling rate of the acoustic signal (here 48000 samples per second). The lower the value of T_a (higher α), the faster the threshold is adapted to the changes in the acoustic background. The time step in seconds between new and previous value equals N/SR.

Double adaptation

The concept behind *double adaptation* is that the threshold should be adapted not only to match the mean level of the detection parameter, but also to match its dispersion. If the dispersion of d is large, the threshold should be higher, in order to reduce the rate

of false alerts. Contrary, if the dispersion is low, the threshold can be lowered, so as to detect more subtle changes in the acoustic background. First, we assume a normal distribution of random variable D with the probability density function (PDF):

$$p(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d-\mu)^2}{2\sigma^2}} \tag{4.11}$$

The PDF is adapted by exponential averaging of the mean μ and standard deviation σ of the distribution:

$$\mu = \mu_{new} \cdot \alpha + \mu_{old} \cdot (1 - \alpha) \tag{4.12}$$

$$\sigma = \sigma_{new} \cdot \alpha + \sigma_{old} \cdot (1 - \alpha) \tag{4.13}$$

The advantage of this approach is that the probability of false alarm (P_{FA}) can be controlled by setting the threshold to a value according to the cumulation distribution function (CDF):

$$F(t; \mu, \sigma) = P(D > t) = \frac{1}{2}\left[1 + erf\left(\frac{t-\mu}{\sigma\sqrt{2}}\right)\right] = P_{FA} \tag{4.14}$$

where erf denotes the error function. Hence, the value of t is obtained using the inverse CDF (quantile function):

$$t = F^{-1}(1 - P_{FA}) = \mu + \sigma\sqrt{2} erf^{-1}(1 - 2P_{FA}) \tag{4.15}$$

while the following approximation of the inverse error function is used ([157]):

$$erf^{-1}(x) \approx sgn(x)\sqrt{\sqrt{\left(\frac{2}{\pi a} + \frac{ln(1-x^2)}{2}\right) - \frac{ln(1-x^2)}{2}} - \left(\frac{2}{\pi a} + \frac{ln(1-x^2)}{2}\right)} \tag{4.16}$$

with $a \approx 0.147$. The μ_{new} and σ_{new} are calculated over a time constant, equal to one second.

Another benefit of this adaptation strategy is that there is no longer need for mapping the sensitivity to threshold arbitrarily, as it was defined in Section 4.1.2 (see Equation 4.3, 4.4, 4.6 and 4.8). Thanks to the employment of the quantile function, only the false alert

rate has to be determined *a priori*, regardless of the definition of detection parameter. We assume that the minimum sensitivity $s = 0$ maps to $P_{FA} = 10^{-6}$ whereas $s = 1$ translates to $P_{FA} = 10^{-2}$.

Triple adaptation

In the first two approaches exponential averaging of both threshold values, mean μ and standard deviation σ was performed with the constant α related to the adaptation time defined in Equation 4.10. In the last presented approach the adaptation time also depends on the acoustic background. Let us consider a situation in which the average level of the detection parameter rises slowly, but fast enough to exceed the threshold, thus yielding a false alert. At some point in time, the mean μ starts to ascend due to a change in the environment. The rate of this rise can be determined by a linear regression:

$$a_\mu = \frac{I \sum_{i=1}^{I} i \cdot \mu(i) - \sum_{i=1}^{I} \mu(i) \sum_{i=1}^{I} i}{I \sum_{i=1}^{I} i^2 - \left(\sum_{i=1}^{I} i \right)^2} \tag{4.17}$$

where i is the frame index and I is the number of frames taken into calculation, which should be large enough to cover at least one second. We can assume that μ will meet the current threshold value t roughly after a time equal to:

$$T[s] = \frac{N\,(t - \mu_{new})}{a_\mu \cdot SR} \tag{4.18}$$

where N denotes the frame size in samples, SR denotes the sampling frequency and μ denotes the mean value of the PDF of the detection parameter. We assume that this value should be assigned to the new adaptation time in order to enable the threshold to update its value before the noise floor exceeds its level. Once again taking Equation 4.10 into consideration we obtain the following formula for α:

$$\alpha = \frac{|a_\mu|}{t - \mu_{new}};\ t > \mu_{new} \tag{4.19}$$

It is advisable to limit the adaptation time to predefined minimum and maximum values (e.g. 5-60 seconds) in order to avoid too fast adaptation or incorrect values of α. After obtaining the adapted α, the parameters (μ, σ) are adapted as in Eqs. (4.12) and (4.13). Next, the procedure presented in double adaptation approach is followed.

Adaptation example

The example of detection threshold adaptation is shown in Figure 4.3. The figure shows the changes of the detector's threshold during 24 hours of operation for three different adaptation times: 10 min, 30 min and 60 min. The detection parameter in this example is equivalent sound level in 10 ms frames. The employed adaptation approach is single adaptation. The analyzed audio data originates from a microphone installed near a busy street (traffic noise is present). It is visible that the detector adjusts its threshold to the changes in the acoustic environment. At night the detection threshold is ca. 15 dB lower than during rush hours. It enables the detection of more quiet sounds. The longer the adaptation time T_a, the smoother the threshold curve. However, the time the algorithm needs to react to a change in the acoustic environment is extended.

In Figure 4.4, an example of threshold changes depending on the adaptation strategy employed are presented. A signal with varying acoustic background and distinct spikes related to foreground events is utilized. In this example the *Impulse detector* is considered. It can be seen that the threshold obtained with double adaptation follows the changes in the acoustic background more closely than the detector with single adaptation, which would yield a few false detections in this case. In the triple adaptation approach the changes in adaptation rate are also apparent. The threshold rises faster for more quickly varying parts of the analyzed signal.

4.2 Buffering

Due to assuming the detection-and-classification approach, a buffer needs to be implemented, which stores the samples of the detected events before they are offered for feature extraction and classification. The principle of the buffering algorithm is simple.

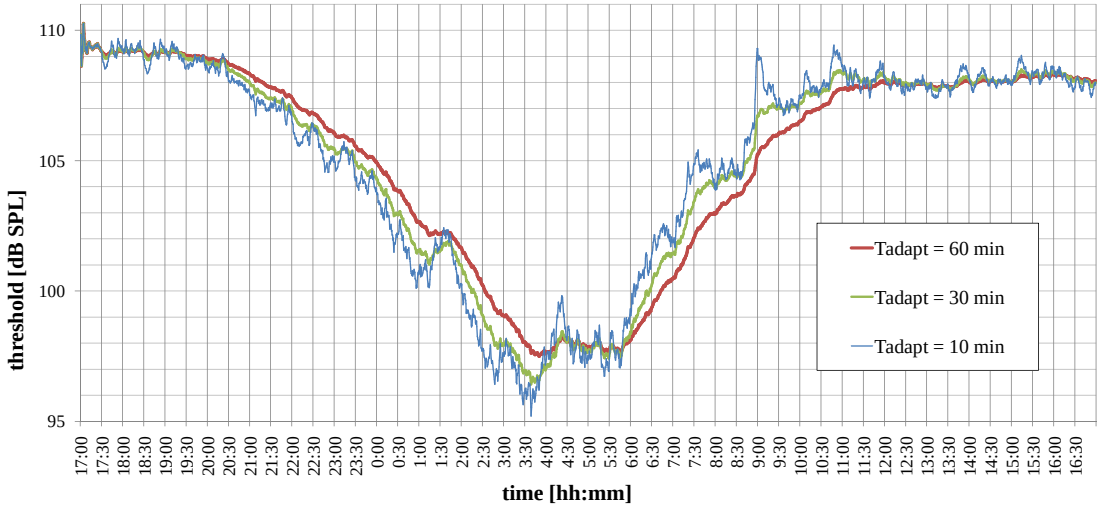


FIGURE 4.3: Changes of adaptive threshold during 24 hours of detector's operation

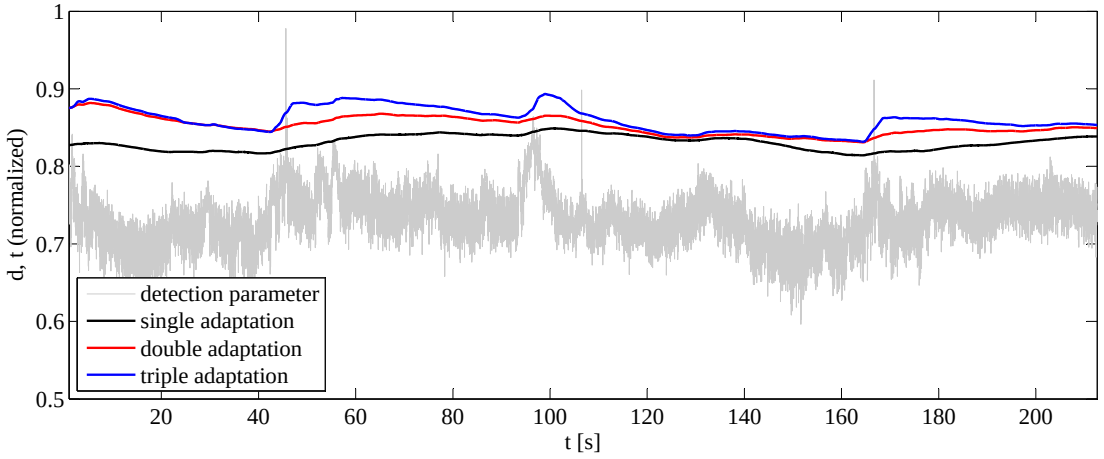


FIGURE 4.4: Illustration of the adaptive threshold changes in different adaptation strategies

The frames in which the detection parameter exceeds the threshold are stored in the memory until the parameter falls below the threshold, or until the maximum length is exceeded. The maximum length of the buffer (b_{len}) determines the maximum capacity of the buffer. If this length is exceeded, one buffer is closed and offered for feature extraction. whereas the next samples are written into the second buffer. From the point of view of the sound recognition engine, the samples in the second buffer constitute another sound event. The purpose of this mechanism is to avoid treating two events, which are close together in time, as one event. However, it also leads to dividing acoustic events which are longer than b_{len} into more than one event (see Figure 4.5). Fortunately, such events can be later joined after the final decision is obtained.

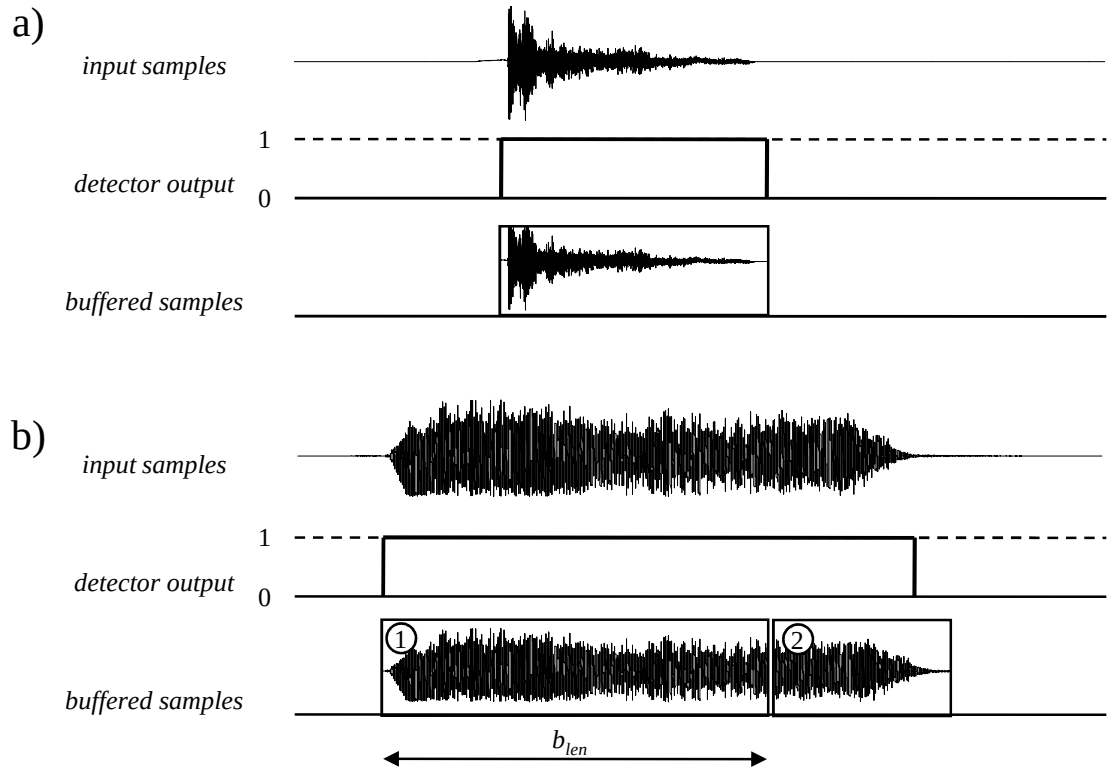


FIGURE 4.5: Example of buffering of acoustic events: a) buffer long enough to fit whole event b) event too long to fit in one buffer

4.3 Feature extraction

As it was outlined in the literature review, all considerable sound recognition applications follow the paradigm of describing the signal in the feature space (see Section 2.3). Moreover, there are several groups of features which are repeatedly used for sound event recognition (e.g. spectral shape features, MPEG-7 descriptors, MFCCs). This approach is also followed in this thesis. We consider the typical features used in related work with an addition of some original parameters formulated as a result of draft experiments. All the features defined in this section constitute the large feature vector, which is later subject to feature selection, performed in Section 6.2.2. The reduced feature vector is then employed for classifying the events.

In Section 2.3 it was mentioned that in online natural sound events classification it is beneficial to extract the features in short-time overlapping frames rather than from the whole signal containing the event. The feature vector is extracted from each short-time frame x_i and the vectors are put together in a feature matrix $\mathbf{F}$. If FE denotes the

feature extraction function, the process can be formally written as in Equation 4.20:

$$\mathbf{F(x)} = \begin{bmatrix} | & | & | & | \\ FE(\mathbf{x_1}) & FE(\mathbf{x_2}) & \cdots\ FE(\mathbf{x_i}) & \cdots\ FE(\mathbf{x_I}) \\ | & | & | & | \end{bmatrix} \qquad (4.20)$$
$$\mathbf{x_i} = \begin{bmatrix} x[i \cdot h] & x[i \cdot h + 1] & ... & x[i \cdot h + a_f - 2] & x[i \cdot h + a_f - 1] \end{bmatrix}^T$$

where a_f denotes the length of the analysis frame and h denotes the analysis hop, i.e. the distance between the two adjacent frames. In further consideration we will use the quantity of the overlap factor (OL) instead of the analysis hop. The overlap factor equals $(a_f - h)/a_f$. The length of the analysis frame and, in particular, the overlap factor have a strong influence on required processing time. For instance, changing the overlap factor from 25% to 75% leads to a triple increase in the number of short-time frames, which extends the time needed to extract the features from the buffered event.

In the following subsections we discuss the different types of features considered for the task of acoustic event classification. Both temporal, spectral and cepstral features are considered. It is shown in the literature that all types of features can provide important information in the process of acoustic event recognition. All features considered in the work are listed in Table 4.1. The reference to the equation with the feature definition or the literature from which it is adopted is also provided.

4.3.1 Spectral shape features

As it was shown e.g. in the work of Peeters [46], the spectral shape features reflect the shape of the power spectrum of the signal, which is useful for discerning between different types of events. Most of the considered spectral features are compliant with the MPEG-7 audio standard [56]. The MPEG-7 descriptors have been successfully used for acoustic event recognition in a number of works [34, 40, 42]. The calculation of the spectral features is preceded by Power Spectral Density estimation. In this thesis Welch's method is used [161]. The employed length of the DFT (Discrete Fourier Transform) is equal to 4096 points and 50% overlap is used. In notation we use either $P_x[k]$ (where k denotes spectral bin index) or $P_x(f)$ (where f is discrete and denotes frequency bin in

TABLE 4.1: List of all audio features

symbol	feature	count	reference
	Spectral Shape Features		
ASC	Audio Spectrum Centroid	1	4.22 [56]
ASE	Audio Spectrum Envelope	34	4.23
ASS	Audio Spectrum Spread	1	4.24 [56]
PVD	Peak-Valley Difference	1	4.27 [37]
SE	Spectral Energy	8	4.25 [158]
SFM	Spectral Flatness Measure	1	4.26 [56]
SFMa	Spectral flatness Measure in bands (A)	24	4.26
SFMb	Spectral flatness Measure in bands (B)	7	4.26
KRT	Spectral kurtosis	1	4.28 [159]
SRF	Spectral Roll-Off	1	4.31 [46][160]
SSL	Spectral Slope	1	4.32 [46]
SPE	Speech Energy	1	4.33
	Temporal features		
LAT	Log-Attack Time	1	4.34 [56]
CF	Crest Factor	1	4.35 [46]
PRD	Periodicity	1	4.36 [46]
TC	Temporal Centroid	1	4.37
ZCR	Zero Crossing Rate	1	4.38 [46]
	Cepstral features		
CCF	Cepstral Crest Factor	1	4.39
MFCC	Mel-Frequency Cepstral Coefficients	24	4.43 [51]
	total:	112	

Fourier analysis) to express the power spectral density function of signal x. The relation between k and f is as follows:

$$\frac{k}{N} = \frac{f}{SR} \tag{4.21}$$

where N is number of points in DFT (here 4096) and SR is the sampling rate (here 48000 samples per second). The resolution of the Fourier analysis is therefore equal to $r = SR/N = 11.7$ Hz.

The Audio Spectrum Centroid (ASC) feature is calculated as a 1-st order normalized spectral moment according to Equation 4.22.

$$ASC = \frac{\sum_f P_x(f) \cdot f}{\sum_f P_x(f)} \tag{4.22}$$

The Audio Spectrum Envelope group of features expresses the signal's energy in 1/3-octave bands relative to the total energy. Provided that the limits of the 1/3-octave band equal f_1 and f_2, the ASE feature in m-th band can be extracted according to Equation 4.23.

$$ASE_m = \frac{\sum_{f_1}^{f_2} P_x(f)}{\sum_{f} P_x(f)} \tag{4.23}$$

A total of 33 1/3-octave bands were taken into consideration spanning from 11 Hz to 24000 Hz.

The Audio Spectrum Spread Parameter equals the 2-nd order normalized central spectral moment and is calculated according to Equation 4.24.

$$ASS = \frac{\sum_{f} P_x(f) \cdot (f - ASC)^2}{\sum_{f} P_x(f)} \tag{4.24}$$

In his research Żwan found out that hazardous events (scream, broken glass or gunshot) have distinctive frequency bands, whose energy ratio differs from one event type to another [158]. This leads to the formulation of *Spectral Energy (SE)* features. They are calculated from the power spectral density $P_x(f)$ as a ratio of energy in two frequency bands - $[f_1; f_2]$ and $[f_3; f_4]$ according to the formula in Equation 4.25.

$$SE = \frac{\sum_{f_1}^{f_2} P_x(f)}{\sum_{f_3}^{f_4} P_x(f)} \tag{4.25}$$

The limits of the frequency bands are shown in Table 4.2.

The next descriptor, the Spectral Flatness Measure, contains the information about the shape of the power spectrum. The SFM features yields values close to 1 when the signal is noise-like and close to 0 when the signal has strong harmonic components. The formula for SFM calculation is given in Equation 4.26:

TABLE 4.2: The limits of frequency bands for Spectral Energy features calculation

feature	**f1 [Hz]**	**f2 [Hz]**	**f3 [Hz]**	**f4 [Hz]**
SE1	100	1000	0	24000
SE2	1000	2000	0	24000
SE3	1300	1700	0	24000
SE4	4000	7000	0	24000
SE5	7000	12000	0	24000
SE6	4000	7000	1000	2000
SE7	2000	4000	1000	2000
SE8	100	500	7000	12000

$$SFM_m = \frac{\prod_{k_1}^{k_2-1} P_x[k]^{\frac{1}{k_2-k_1}}}{\frac{1}{k_2-k_1}\sum_{k_1}^{k_2-1} P_x[k]} \tag{4.26}$$

where k_1 and k_2 denote the limits of the m-th frequency band. The indices k_1 and k_2 translate to frequencies f_1 and f_2 with the formula in Equation 4.21. The feature is calculated in 3 variants. The broadbanded parameter SFM is calculated over the whole spectrum ($f_1 = 0; f_2 = 24000Hz$). The $SFMa$ features are obtained from 1/3-octave bands (24 bands from 71 to 16000 Hz). The $SFMb$ parameters are determined in 7 1-octave bands from 500 Hz to 16000 Hz.

The *Peak-Valley Difference* feature is a modification of the parameter described in the literature [37] and is calculated according to Equation 4.27.

$$PVD = \frac{\sum_{k=1}^{N/2} P_x[k] \cdot V[k]}{\sum_{k=1}^{N/2} V[k]} - \frac{\sum_{k=1}^{N/2} P_x[k] \cdot (1 - V[k])}{\sum_{k=1}^{N/2} (1 - V[k])} \tag{4.27}$$

where $P[k]$ denotes the power spectrum of the signal, N equals the number of DFT points and $V[k]$ is a binary vector, in which *ones* are located in the points in which spectral peaks are detected. To find the locations of the spectral peaks, a grid search is performed. Assuming the peaks are equally spaced and the space between the peaks lies in the range between 80 Hz and 800 Hz, the space between peaks which yields maximum PVD is chosen.

Spectral Kurtosis is a useful parameter for non-stationary signals [159]. The feature is calculated from the 2-nd order centralized moment of the power spectrum M_{2c} and 4-th order centralized moment of the power spectrum M_{4c} according to Equation 4.28.

$$KRT = \frac{M_{4c}}{M_{2c}^2} - 3 \tag{4.28}$$

where:

$$M_{4c} = \frac{\sum\limits_{f} P_x(f) \cdot (f - ASC)^4}{\sum\limits_{f} P_x(f)} \tag{4.29}$$

$$M_{2c} = \frac{\sum\limits_{f} P_x(f) \cdot (f - ASC)^2}{\sum\limits_{f} P_x(f)} = ASS \tag{4.30}$$

The next parameter is *Spectral Rolloff* and it denotes the frequency under which 95% of the total energy is accumulated [46]. It was largely used by Kos et al. [160] for sound event recognition.

$$SRF = \min\{f_c : \sum_{0}^{f_c} P_x(f) \geq 0.95 \cdot \sum_{f} P_x(f)\} \tag{4.31}$$

The *Spectral Slope* parameter is the slope of the linear regression of the power spectrum [46]. It is calculated according to the formula:

$$SSL[1/Hz] = \frac{N \sum\limits_{k} f_k \cdot P_x[k] - \sum\limits_{k} f_k \cdot \sum\limits_{k} P_x[k]}{N \sum\limits_{k} f_k^2 - \left(\sum\limits_{k} f_k\right)^2} \tag{4.32}$$

where f_k denotes the center frequency of the k-th spectral bin.

The final spectral shape parameter introduced reflects the ratio of energy in the speech band to the whole energy of the signal. It is meant for discerning between speech and non-speech sounds. The feature *Speech Energy* is defined as follows:

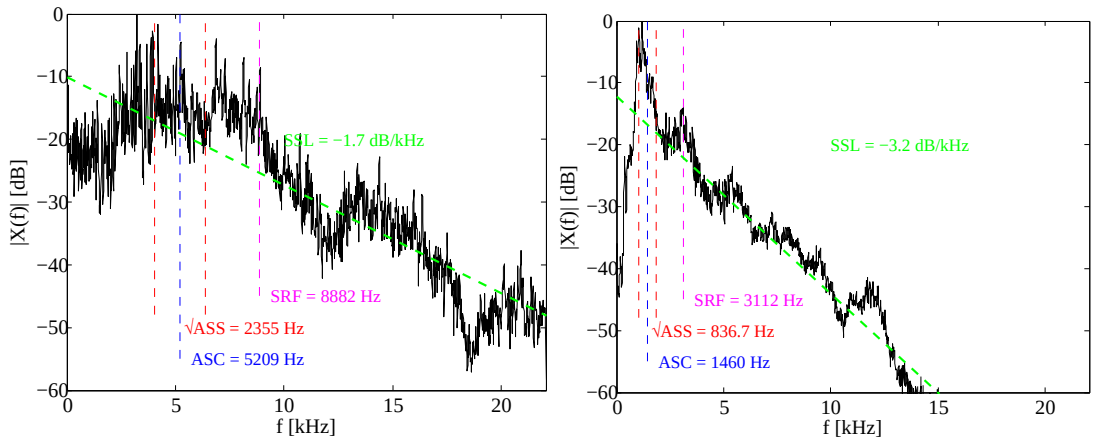


FIGURE 4.6: Example spectral shape parameters of breaking glass (left) and scream (right) event

$$SPE = \frac{\sum_{300}^{3400} P_x(f)}{\sum_{0}^{24000} P_x(f)} \tag{4.33}$$

An example interpretation of the chosen spectral shape features is shown in Figure 4.6. The power spectra of a breaking glass and scream event are plotted - normalized, in decibel scale. Four parameters are depicted: *Audio Spectrum Centroid ASC, Audio Spectrum Spread ASS, Spectral Roll-Off SRF* and *Spectral Slope SSL*. The difference in the values of the parameters are apparent. The energy of the breaking glass event is concentrated in higher parts of the spectrum, compared to scream event. Hence, it yields higher values of both *ASC* and *SRF* features. Moreover, the spectrum of glass sound is wider spread, which is reflected by the value of the ASS parameter. In contrast, the energy of scream is concentrated in lower frequencies, and the spread of the spectrum is smaller. Also, the energy of scream diminishes faster in the frequency scale, which is reflected by the value of the *Spectral Slope* descriptor. Such differences in the parameter values are a basis for training the classifier to discern between different types of events.

4.3.2 Temporal features

The temporal features are calculated directly from the time-domain representation of the event - the digital signal $x[n]$. They reflect the shape of the waveform and can be

very useful for discerning between different types of acoustic events.

The *Log-Attack Time* (LAT) feature is calculated from the signal's envelope. The RMS envelope is calculated in 512 sample frames (10.7 milliseconds at 48000 S/s). The maximum of the envelope is sought. According to the MPEG-7 standard, the LAT feature is determined according to the formula:

$$LAT = log_{10}(i_{max}) \tag{4.34}$$

where i_{max} is the index of the frame in which the maximum of the envelope is found. The lower the i_{max}, the closer to the beginning of the signal is the peak located. Thus, the attack time is considered shorter.

The *Crest Factor* parameter is a simple descriptor of signal shape. Here it is defined as a ratio of the signal's RMS (root-mean-square) value to the peak value:

$$CF = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}}{\max_{n=1}^{N} |x[n]|} \tag{4.35}$$

where N is the number of samples in the signal. The lower the CF parameter, the more impulsive the character of the signal. While many sources define crest factor as *peak to rms*, we use *rms to peak* instead, since it is by definition constrained to the interval [0;1].

Periodicity is a feature typically calculated from the autocorrelation function [46]. The autocorrelation of the digital signal is calculated according to the formula:

$$R_{xx}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x(n+m) \cdot x(n); \ m \geq 0 \\ R_{xx}(-m); \ m < 0 \end{cases} \tag{4.36}$$

Next, the maximum of the autocorrelation is sought between the indices $m_1 = SR/700, m_2 = SR/80$, where SR denotes the sampling rate. The limits correspond to the pitch range between 80 Hz and 700 Hz which is enough to cover the pitch range of human voice and

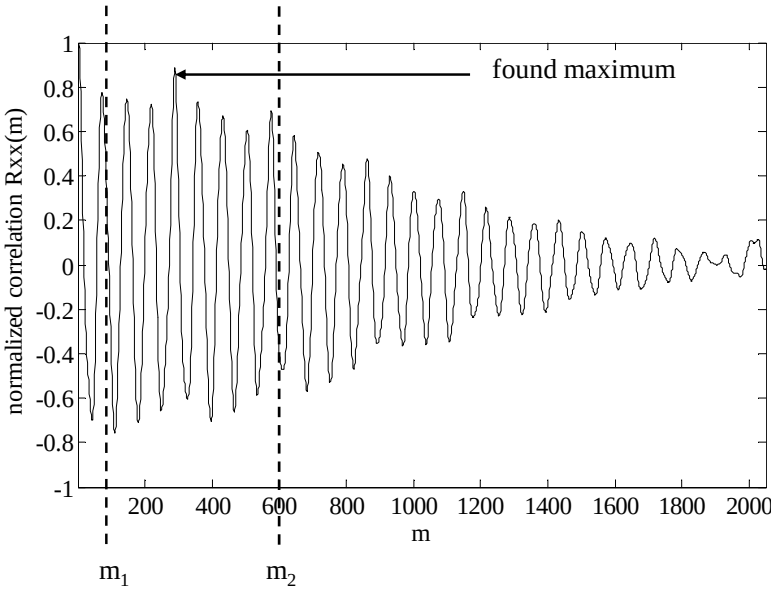


FIGURE 4.7: Example of searching for the maximum of the autocorrelation function

many other sounds. The *Periodicity* feature equals the maximum of the autocorrelation function in that range. The process of searching for the maximum of the R_{xx} function is shown in Figure 4.7.

The *Temporal Centroid* feature is calculated as a 1-st order moment of the signal's energy in the time domain:

$$TC[s] = \frac{1}{SR} \frac{\sum_{n=1}^{N} n \cdot x[n]^2}{\sum_{n=1}^{N} x[n]^2} \tag{4.37}$$

where SR denotes the sampling rate, n is the sample index and N is the number of samples in the signal.

Zero Crossing Rate is a well-known feature used for identifying noisy and deterministic signals. It is calculated according to the formula:

$$ZCR = \frac{1}{2N} \sum_{n=2}^{N} |sgn\left(x[n]\right) - sgn\left(x[n-1]\right)| \tag{4.38}$$

where N is the total number of samples in the signal $x[n]$.

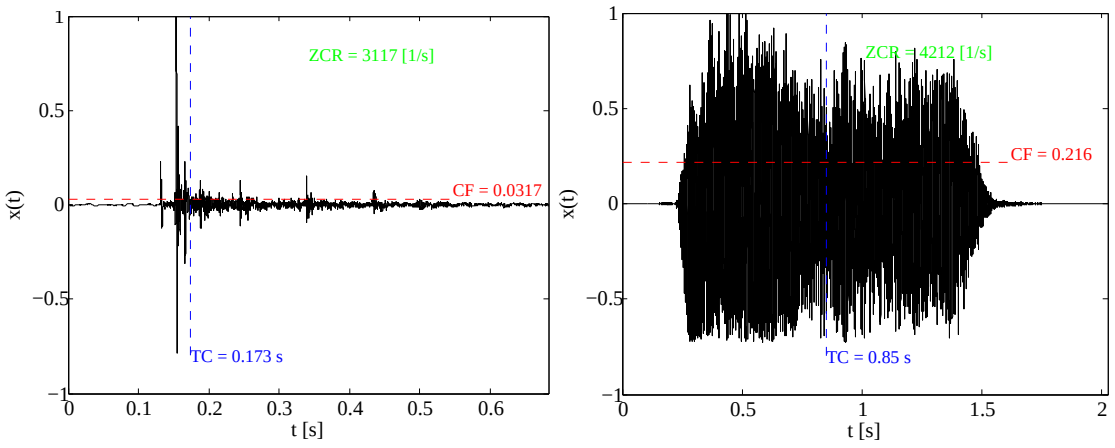


FIGURE 4.8: Example temporal parameter of gunshot (left) and scream (right) event

In Figure 4.8 example temporal parameters of gunshot and scream event are depicted. The differences in the time domain representations of these event are reflected by the values of the illustrated parameters: *Crest Factor CF, Temporal Centroid TC* and *Zero Crossing Rate ZCR*. The gunshot event is impulsive, so it yields lower crest factor (note in Equation 4.35 that crest factor is here defined as the reciprocal of its most common form). The temporal centroid of the scream event is also much higher and approximately in the middle of the event, which shows that the energy is evenly distributed in time. Finally, the ZCR parameter for the high-pitched tonal scream event is higher than for the noisy gunshot event.

A common technique to describe the temporal properties of sound bases on the so-called ADSR envelope. It divides the sound into four stages: Attack (A), Decay (D), Sustain (S) and Release (R). Even though the ADSR envelope was originally observed for musical sounds, the four phases can also be distinguished in other sound events. For example, as visible in Figure 4.8, scream event has all the ADSR phases, yet the gunshot sound lacks the sustain phase. The temporal properties of the considered sound are analyzed in more detail further on in Section 6.1. Thus, the analysis in means of the ADSR envelope could lead to some distinction between the mentioned classes of sounds. However, the ADSR parameters are not considered in this work due to the assumed approach to feature extraction. As it was defined in Section 4.3, the parameters are extracted in short-time frames and then fed into the classifier. Hence, the sound event is not analyzed as a whole and the features describing the overall shape of the event

are not considered in the classification process. Moreover, the signal envelope is easily distorted by additive noise. It could be difficult to extract the information about the ADSR envelope i noisy conditions. Nevertheless, the holistic temporal properties of sounds, including the ADSR envelope, should be considered in future research.

4.3.3 Cepstral features

The cepstral features are known to depict the shape of the spectral envelope and have been employed in a number of related works. Apart from the widely used MFCCs, we introduce *Cepstral Crest Factor - CCF*. Calculating the crest factor in the cepstral domain provides information about the noisiness of the signal. Tonal (or close to tonal) signals have a distinct peak in the upper part of the cepstrum. The CCF feature is calculated according to the formula:

$$CCF = \frac{\sqrt{\frac{1}{K}\sum_{k=1}^{K} C[k]^2}}{\max_{k=1}^{K} |C[k]|} \tag{4.39}$$

where $C[k]$ is the cepstrum of the signal:

$$C[k] = \mathcal{F}\left\{\mathcal{F}\{x[n]\}\right\} \tag{4.40}$$

where $\mathcal{F}$ denotes the Fourier transform (here DFT); k is the quefrency bin index and K is the number of quefrency bins (equal to half the number of frequency bins in DFT).

The feature vector is completed with *Mel-Frequency Cepstral Coefficients - MFCC*. The MFCC features are calculated with a generic method described in the literature [51, 91]. First, a mel filterbank is applied to the power spectrum of the signal to calculate the energy in each of the mel-frequency bands. The following formula for calculating the mel-scale frequency is used:

$$M(f) = 1127 \cdot ln\left(1 + \frac{f}{700}\right) \tag{4.41}$$

where f is the frequency in Hz. Next, $M = 24$ triangular filters placed uniformly in the mel-scale are applied. The energy of the m-th mel-frequency band is calculated as follows:

$$E_m = \sum_k H_m[k] \cdot P_x[k] \tag{4.42}$$

where $H_m[k]$ is the digital frequency response of the m-th triangular filter, $P_x[k]$ is the power spectral density in terms of frequency bin index k. Finally, the MFCCs are obtained by applying logarithm and DCT to the calculated energy coefficients.

$$MFCC_m = \sum_{k=0}^{M-1} w(k) \log E_k \cdot cos\left[\frac{\pi m(2k+1)}{2M}\right] \tag{4.43}$$

where k is the frequency bin index, m is the index of computed MFCC coefficient, M denotes the number of subbands considered and the weight $w(k)$ equals $\sqrt{\frac{1}{M}}$ for $k = 0$ and $\sqrt{\frac{2}{M}}$ for $k > 0$.

4.3.4 Normalization

The normalization of the signal features is essential for correct operation of the SVM classification algorithm employed in this work. We employ min-max normalization. The normalized parameter value v_{norm} is calculated according to the formula:

$$v_{norm} = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{4.44}$$

where v_{min} and v_{max} are the global minimum and maximum values of the parameter. The normalized feature ranges from 0 to 1. If possible, the v_{min} and v_{max} values should be defined *a priori*. If not, the maximum and minimum values of the signal feature are extracted from the training set and stored together with the classification model.

4.4 Classification

From a number of classification techniques featured in the literature and described in Section 2.5, the Support Vector Machine (SVM) classifier is employed for discerning between the different types of acoustic events. This algorithm is reported to yield comparable or better accuracy than HMM [75, 87]. Its great advantage is smaller computational cost than HMM or GMM and faster training than ANN. It renders the SVM algorithm particularly suitable for online operation, which is assumed in this work. Finally, SVM is a good tool for modeling the general properties of sounds, rather than the fine temporal structure. In the author's opinion it is a good trait as far as sound event classification is concerned.

The LibSVM C++ library is used [85] for implementation. The classifier is fired separately for a feature vector obtained from each short-time frame (whose length equals a_f and overlap factor equals OL). Thus we follow the *bag-of-frames* approach. A multiclass SVM model with *one-vs-one* technique is employed. So, in fact for 5 classes of sounds, 10 binary SVM classification models are created (for each pair of classes). LibSVM also enables the use of a probability model, which outputs class probabilities. The probability output reflects the certainty that the classified signal belongs to one of the predefined classes: *explosion, broken glass, gunshot, scream* and *other*. The example output of the classifier recognizing a scream event is presented in Figure 4.9. The certainties pertaining to each class for each short time frame are plotted.

The maximum certainty rule is used for making the decision based on results from the respective short-time frames. Moreover, the thresholds for each class need to be defined, to limit the rate of false alerts. It is required that the probability of event belonging to a certain class exceeds a predefined threshold for this class to be taken into consideration while making the decision. Thus, the sensitivity of the classifier can be adjusted. By lowering the threshold for a given class, more events can be qualified into this class, which potentially increases the TP rate, however at the cost of increasing the false alert rate. An experiment aiming at finding the optimum class thresholds is featured in Section 6.3.2.

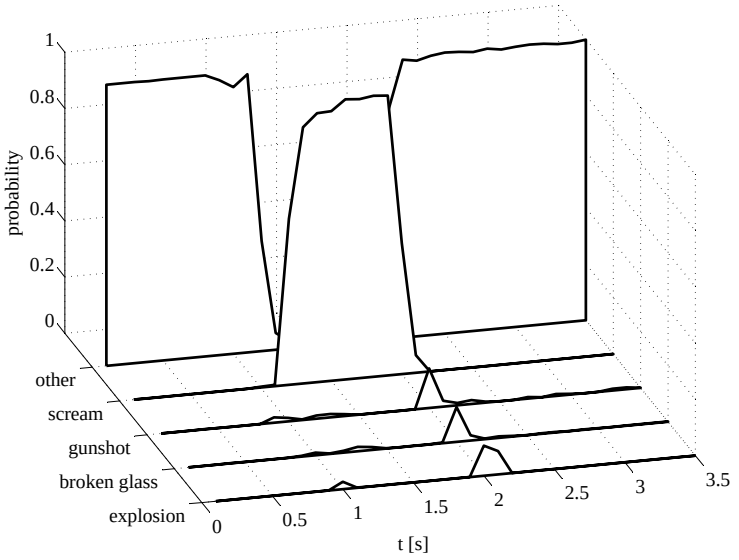


FIGURE 4.9: Example output of the SVM classifier

The important issue concerning classification is the *critical section* of the classifier. The operations required for recognition of acoustic events constitute a processing flow which is expressed with the following pseudocode:

- Detection thread

```
if detection_parameter > threshold
start buffering
end

if buffering & detection_parameter < threshold
stop buffering
end
```

- Classification thread

```
if buffer_ready
if classifier_busy
wait
else
classifier_busy := true
```

```
classify buffer
classifier_busy := false
end
end
```

The detection and the classification of acoustic events constitute separate threads. The analysis of the code indicates a possibility that, in cases where a new event has been detected and the classifier is still busy, the program has to wait until the classifier is free. The aim is to avoid conflicts in access to the memory, in particular in accessing the signal samples by the feature extraction functions. The sound events which are waiting for classification are organized in a FIFO queue. The phenomenon of the *critical section*, can in some cases lead to backlogs and increase the decision time. This problem is dealt with in Section 8.2.

Chapter 5

Implementation on a supercomputing cluster

The algorithms introduced in Chapter 4 are implemented in the environment of a supercomputing cluster. In this chapter the details regarding the implementation are provided. The technical infrastructure and the software framework utilized in the work are presented. Also, the supercomputing services created by the author of the thesis are introduced. The sources of audio stream, providing the input data, are specified. Finally, the client application with a specialized graphical interface developed in Multimedia Systems Department, which enables to the user to execute the services and observe the results, is presented.

It has to be noted that before the implementation on the supercomputing platform all the engineered algorithms were prepared for online operation. The created C++ code was analyzed and adjusted to ensure fast and reliable execution as well as cross-platform compilation.

5.1 KASKADA platform

The experiments featured in this dissertation are carried out in the environment of a supercomputing cluster, exploiting a specialized framework for parallel processing of

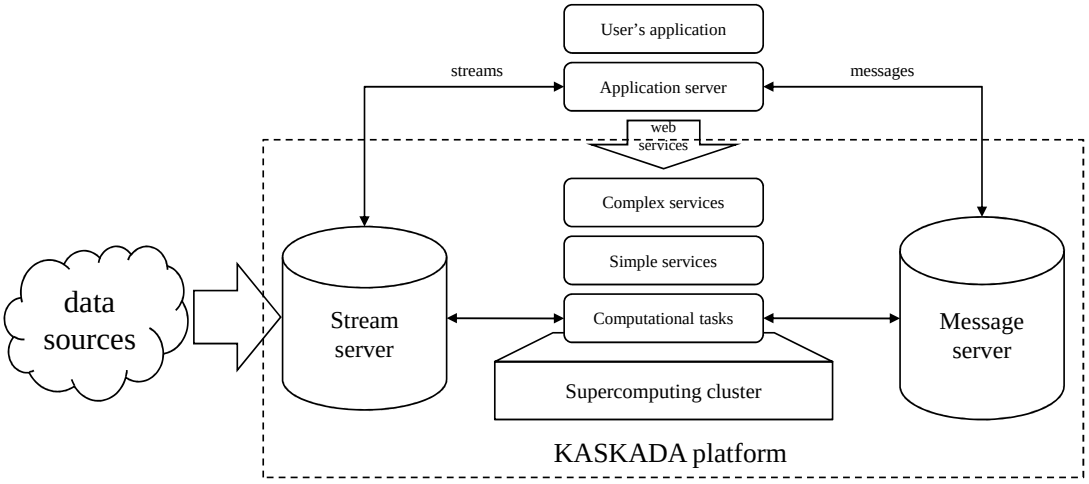


FIGURE 5.1: Concept diagram of the KASKADA platform [155]

multimedia data streams - KASKADA platform [121, 162–165]. The Polish abbreviation KASKADA stands for *Context Analysis of Camera Data Streams for Alert Defining Applications*. Both the framework and the supercomputer constitute the infrastructure of the project *MAYDAY Euro 2012*. In the following subsections the architecture of the platform and the mechanisms designed to facilitate multimedia stream processing, are described.

5.1.1 Platform architecture

The concept diagram of the KASKADA platform is shown in Figure 5.1. The data sources, i.e. cameras and microphones, provide the input data for the platform. The infrastructure of the platform comprises the supercomputing cluster and the stream and message servers. Outside the platform, an application server is set up. The application server is equipped with a user's console (UC), accessible by Internet. The application server offers the KASKADA services as web services, according to the *Software as a Service* principle. The user's applications also have the access to multimedia data streams and messages. The scenarios specified in the applications are realized by complex services. The complex services are composed of simple services, which comprise computational tasks. The tasks are executed on the supercomputing cluster, thus performing the processing required in the application.

For better understanding of this concept, let us consider a scenario related to sound recognition. The application offers a functionality of recognizing hazardous acoustic events in a chosen audio stream. The user chooses the source, configures the parameters (e.g. sensitivity of event detection, type of detected events) and starts the application. The application sends a request to the KASKADA platform to start a pertinent service. The complex service related to recognition of acoustic events comprises three simple services: one for preprocessing of audio data stream, one for sound recognition and one for presentation of results (see Section 5.1.3). The computational tasks required by the service are related to the actual operations on the input data which are needed to detect the acoustic events. These operations, i.e. detection, buffering, feature extraction and classification, are described in Chapter 4. The input data for the processing tasks is delivered by the stream server, which on the other end connects to the data sources (i.e. microphones). When a hazardous acoustic event is detected, a message is sent to the message server, which passes it on to the client's application. The user is able to listen (or watch) the input stream and to see the incoming events in the application's graphical interface.

The layer model of the KASKADA plaftorm, shown in Figure 5.2, gives more insight into the organization of the framework. The top layer is the application layer, which is seen from the user's point of view. The scenarios exploited in applications are realized by *complex services*. These comprise *simple services*, which execute *computational tasks*. The computational tasks are the specific implementation of the stream analysis algorithms [163]. The processing tasks may be single threads, or may be composed of several processes or threads. The process/thread layer exploits the standard parallelism mechanisms (e.g. POSIX threads) to execute computational tasks [163].

It is worth mentioning that, thanks to the ability to connect both acoustic and visual data sources, the KASKADA platform is a suitable environment for developing multimodal analysis (as mentioned in Section 2.9). The platform architecture enables services operating on audio and video data to communicate, exchange processing results and thus reinforce the event detection. Experiments devoted to such application of the framework were presented in related work [13].

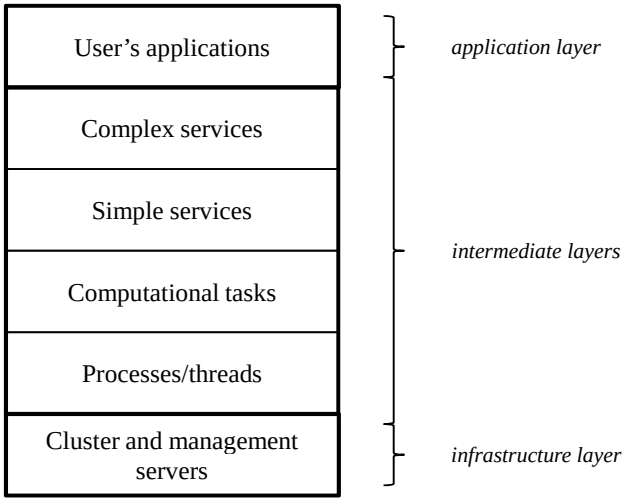


FIGURE 5.2: Layer architecture of the KASKADA platform [121, 162]

Apart from the sound event detection methods outlined in this dissertation, the KASKADA platform has also been successfully used for the following applications:

- video event detection, including object detection and tracking [166, 167] or crowd behavior analysis [168, 169];
- supporting medical examination, by analyzing the images from endoscopic capsules [170];
- protection of intellectual rights - comparing the document with the documents from the repository for detection of intellectual property infringement [170].

The experiments presented in the thesis were performed in the environment of the cluster *Galera* and later *Galera Plus*, both located in Academic Computer Network Center (TASK) in Gdańsk University of Technology. In the project, initially the *Galera* cluster was utilized, but later another cluster *Galera Plus* was purchased and employed specifically for the *MAYDAY Euro 2012* project. The resources of both clusters are compared in Table 5.1.

5.1.2 Resource allocation

The essential feature of the KASKADA platform is that it comprises mechanisms for allocating resources within the supercomputing cluster [165]. The developer, not to

TABLE 5.1: Resources of *Galera* and *Galera Plus* cluster

	Galera	**Galera Plus**
number of nodes	336	192
number of CPUs	1344	384
number of cores	5376	2304
cores per node	8	12
CPU type	Intel Xeon Quad Core 2.33 GHz	Intel Xeon Six-Core 2.27 GHz
physical memory	10752 GB	3072 GB
disk space	107.5 TB	512 TB
network	InfiniBand	InfiniBand
network bandwidth	20 Gb/s	40 Gb/s
operating system	Linux	Linux

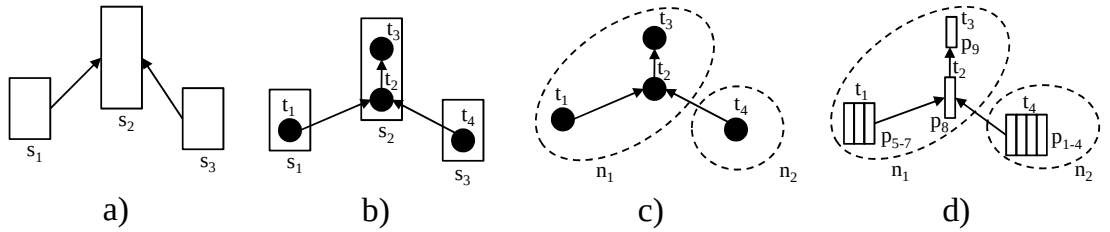
mention the user of the service, is not concerned with the CPU and memory availability. Moreover, separate simple services constituting a complex service or separate slave algorithms can be executed on different cluster nodes, depending on the available resources. The synchronization and communication between the tasks is handled by the framework.

The resource allocation process is presented in Figure 5.3. The process can be divided into the following steps [163]:

1. *Creation of simple services.* In the initial step the scenarios and complex services are divided into simple services. Each simple service handles a logical operation on the multimedia stream, e.g. detection of acoustic events, classification of acoustic events, etc. (see Section 5.1.3).

2. *Estimation of resource requirements.* The demand of a simple service for computational resources depends not only on the intrinsics of the algorithms employed, but also on the parameters of the input stream, e.g. resolution or bitrate. In this step the resource requirements of the individual simple services are considered and computational tasks are created. Note in Figure 5.3b that one simple service may comprise more than one computational task.

3. *Assigning computational tasks to cluster nodes.* Basing on the requirements computed in step 2., the processing tasks are assigned to available cluster nodes. It



s_i - simple service, t_i, computation task, n_i - computation node, p_i - process/thread

FIGURE 5.3: Allocation of resources in the KASKADA platform a) simple services, b) task graph, c) assignment of tasks to computation nodes, d) execution of tasks as processes or threads [163]

is important to mention that the architecture of the platform demands that all threads of a single simple service have to be executed on the same node.

4. *Starting computational tasks.* The processing tasks are executed on respective cluster nodes. Each task may consist of one or more process or threads (see Figure 5.3d). When a task consists of multiple processes or threads, parallelism mechanisms are employed to execute them in parallel on different cores.

5. *Monitoring of tasks.* The KASKADA platform monitors the running tasks in terms of physical memory and processor load. When the declared load is exceeded, the task is terminated and an exception is thrown.

6. *Termination.* The processing is finished at user's request or when the task is completed or end of stream is reached. The platform terminates all tasks belonging to the scenario and frees the allocated resources.

One service can be run in parallel on multiple sources. In the case of sound recognition algorithms, it is a key benefit. The supercomputer allows for processing a large number of streams and passing the information about the detected anomalies to the person responsible, employing the framework's messaging system.

The employed resource allocation mechanisms ensure that the KASKADA platform is scalable. With the increase of the number of sources and (proportionally) the processing nodes, the scalability of the cluster, as defined in Equation 3.3, is maintained at a constant level, which was proved by Proficz [121].

5.1.3 Services and algorithms

From the developper's point of view, The KASKADA framework provides the following components dedicated to processing multimedia data streams [155]:

- **Stream algorithms** implement methods for the real-time processing of audio and video data. Given the proper input, in the form of audio or video frames, they perform the pertinent data processing tasks. In this case a stream algorithm is used to handle the input audio data and execute sound recognition algorithms.
- **Slave algorithms** - in cases when a large portion of data needs to be processed, a slave process can be executed. After the processing is finished the result is returned to the master algorithm. However, slave algorithms are not dedicated to online stream processing. The execution of the slave task requires an amount of time, which is unacceptable in time-critical decision making.
- **Master algorithms** - master algorithms have the ability to start slave processes and receive their result. All stream algorithms also inherit from master algorithms.
- **Simple services** - a simple service can acquire stream output, process the data employing the algorithm attached to the service and produce stream output, as well as generate events. A service to be executed requires to run a task working according to a defined algorithm. The service can be made available to the user who has a demand for its result. A simple service is also the basic building block of complex services.
- **Complex services** - two or more simple services can form a complex service. The output of one service in this chain becomes the input of another. This schema is particularly useful in the case of connecting operations which are performed sequentially (e.g. detection and classification of acoustic events).

The mentioned processing units are exploited in the work described in the thesis. Therefore, the presented terms will be referenced throughout the dissertation, especially in Chapter 8.

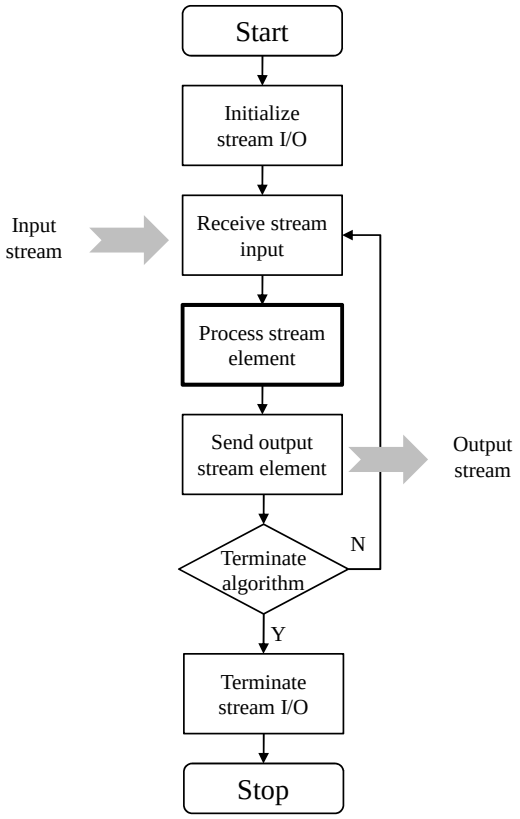


FIGURE 5.4: Block diagram of a stream processing algorithm in the KASKADA framework [121]

From the point of view of this work, the most important class of algorithm is the *stream algorithm*. The general block diagram of a stream algorithm is shown in Figure 5.4. The algorithm executes in a loop, in which stream elements are received, processed and sent to the output stream. In the case of audio stream processing algorithms, the stream element is an audio frame, comprising a packet of audio samples, interleaved if there are is more than one channel. The samples are represented by 32-bit floating point numbers (*float* datatype). The operations pertaining to sound recognition are embedded in the highlighted block *Process stream element*. Apart from outputting data stream, the stream algorithm can also output events.

5.1.4 Communication

The objects in the KASKADA framework communicate, e.g. by exchanging input/output data and processing results. Two types of communication can be distinguished [171]:

- stream communication - intended for transmitting multimedia data. The endpoints for this communication are two algorithms, most likely being parts of a complex service. Any serializable data object can be transferred between two algorithms using *kbin* protocol (internal protocol of the KASKADA framework). When the data object is an audio or video frame it can also be sent via RTSP (Real-Time Streaming Protocol). The *kbin* protocol is used to transmit uncompressed streams between tasks, whereas the RTSP protocol is used for communication outside the KASKADA platform. For transmitting the multimedia data from the sources to the platform and from the platform to the users' endpoints, compression is necessary.

- event communication - XML events can be sent from and received by every KASKADA algorithm. They are utilized e.g. to provide additional communication between blocks in a complex service. Events can also be sent from and received outside the platform, thus enabling the user of the service to alter its parameters or to receive the results. This mechanism is exploited in the client application presented at the end of this chapter in Section 5.4. An ActiveMQ server is used for queuing the messages. Due to the queuing mechanisms, event messages are not suitable for real-time communication between processing blocks. It is better to encode time critical data in the data object sent via *kbin* or RTSP protocol.

5.2 Audio signal acquisition

The input data streams which are available in the framework are provided by sources located within the Gdańsk University of Technology campus, i.e. cameras and acoustic sensors. As far as audio data is concerned, two types of data sources are considered: a pressure microphone and acoustic vector sensor (AVS), commonly referred to as a Microflown. A microphone provides one channel stream, whereas an AVS produces four stream of audio data, i.e. acoustic pressure p and three orthogonal components of particle velocity u_x, u_y, u_z. The data from the AVS are used in the *localization* services

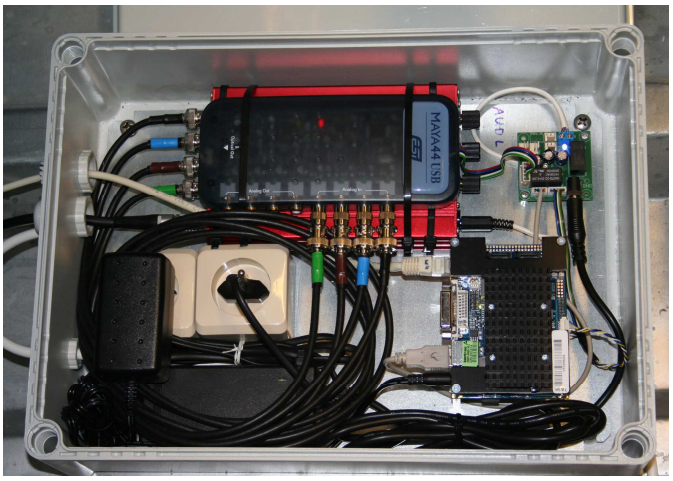


FIGURE 5.5: RTSP audio server comprising a single-board computer, external sound card and conditioning module for acoustic vector sensor

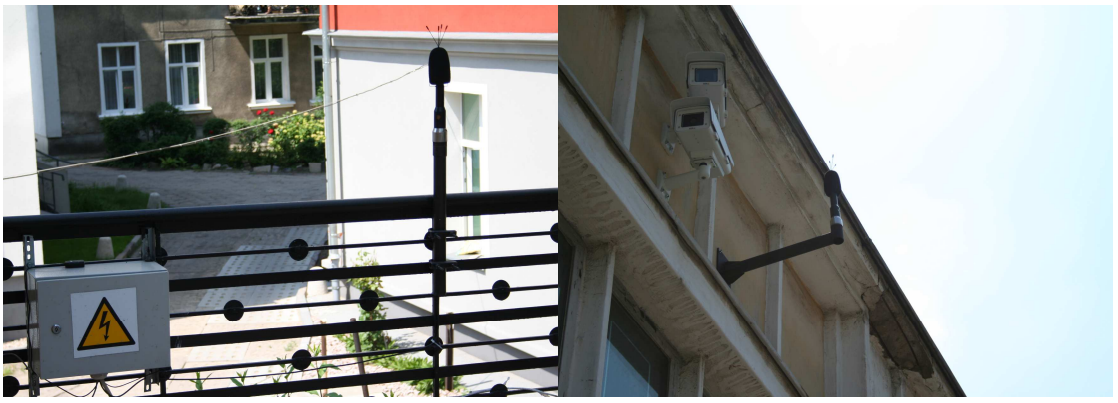


FIGURE 5.6: Example setup of microphones used for audio data acquisition

(see Section 7.2.4). Due to a high resolution of audio samples needed in the sound recognition algorithms, an assumption was made that the data should be transmitted from the sources to the cluster with a sampling rate equal to 48000 S/s and 16-bit per sample with PCM (Pulse Code Modulation) encoding.

In order to enable the transmission of four audio data streams with the desired resolution a specialized software and hardware was engineered in Multimedia Systems Department. The hardware consists of a multichannel sound card and a single board computer (Kontron pITX). A RTSP streaming application constitutes the software. The device is referred to as *RTSP server* [13, 20]. It is able to transmit up to 4 channels of audio at 48000 S/s and up to 32 bit per sample. The RTSP server and example microphone setups are visible in Figure 5.5 Figure 5.6 respectively.

5.3 Sound recognition services

In this section the supercomputing services intended for processing of audio data streams are introduced. The services have been designed employing the KASKADA platform and programming environment.

5.3.1 Simple services

AudioForwarder The component serves as a first step in the signal processing chain. Its purpose is to prepare the data stream for analysis with the sound recognition algorithms, in particular by alignment of data chunks or demultiplexing a multichannel audio stream. The output of this service is a stream of binary data, from which audio data chunks, i.e. *audio frames* are extracted by means of deserialization. The service also provides the functionality of registering the raw data stream on disk.

SoundRecognition The service is an implementation of the sound recognition algorithm introduced in Chapter 4. It operates according to the diagram in Figure 4.1, comprising both detection, buffering and classification operations. The input of the service is audio data stream, whereas the output is a stream of audio data with parameters concerning the results of detection and classification of events, as well as a stream of events passed to the message queue. The XML syntax of an example event produced by the service is shown in Figure 5.7. The XML event contains the data concerning the type and location of detected event, the start and end time of the event and the name of camera associated with the location, if available. The events which are considered threatening are also assigned high priority, whereas typical events are assigned low priority. The service requires as a parameter the path to the configuration file which stores the parameters needed to configure the sound recognition engine, such as types of used detectors, classification models etc.

sound_visualization The component is dedicated to visualizing the results of the detection and recognition of acoustic events. The input of the service is a stream of

```
<?xml version="1.0" encoding="UTF-8"?>
<kaskada timestamp="20121205T131625.596" type="user" subtype="kaskada.framework.event" from-
id="409122" to-id="manager">
    <event name="Sound event detected" service-
id="39170" hash="50ffef34d10ec49e51c857928d941517e5277b6f5c92eb0e15363f0c5675a058b5ab492e74893
fcaf5819d55202a3182c749abf27d7783c61868452515aa2552">
        <parameter name="camera">
        </parameter>
        <parameter name="class">
            scream
        </parameter>
        <parameter name="classIdx">
            4
        </parameter>
        <parameter name="eventEnd">
            20121205T141624.564724
        </parameter>
        <parameter name="eventStart">
            20121205T141622.583197
        </parameter>
        <parameter name="location">
            siedlicka
        </parameter>
        <parameter name="priority">
            1
        </parameter>
    </event>
</kaskada>
```

FIGURE 5.7: XML syntax of an example event produced by the service *SoundRecognition*

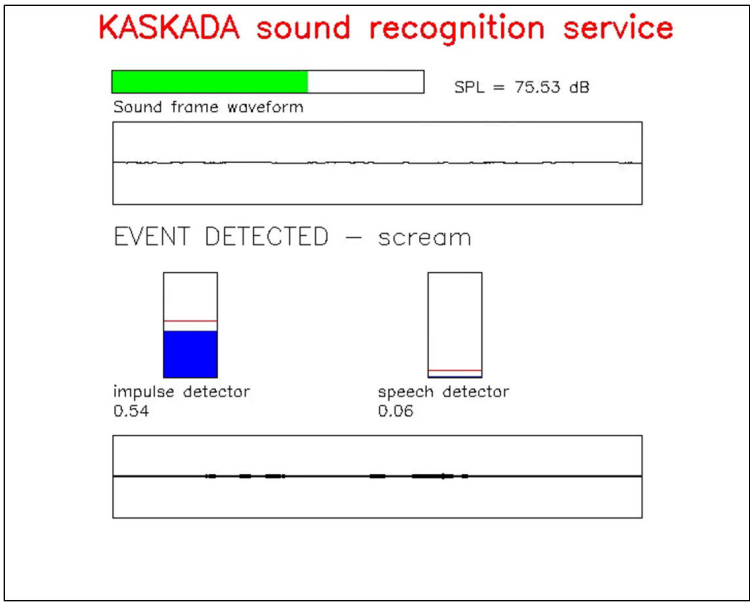


FIGURE 5.8: Example output of the *sound_visualization* service

audio frames with additional parameters concerning the results of events recognition. The service renders a video image presenting the results of the audio stream analysis. The example output is shown in Figure 5.8. The parameters visualized are i.a. current sound pressure level (SPL), detection parameters, detector thresholds, waveform of the detected event and type of detected event.

FIGURE 5.9: Graph of the *SoundRec_complex* service

audytorium The service is designed to analyze the multichannel audio data from the AVS with a view to detecting and localizing threatening events in a public interior space and pointing the PTZ camera in the direction of the detected event. The testbed was created in a lecture hall in Gdańsk University ff Technology. The method and its practical application is further described in Section 7.2.4 and in author's publications [6, 7].

5.3.2 Complex services

SoundRec_complex The service provides the functionality of recognizing acoustic events and visualization of results. As it is shown in Figure 5.9, the complex service comprises 3 simple services: *AudioForwarder*, *SoundRecognition* and *sound_visualization*. The input of the *SoundRec_complex* service is audio data, whereas the output is video data stream (with visualization) and event stream with information concerning the results of event recognition.

audytorium_complex The service is constructed similarly to the *SoundRec_complex* service. It comprises the *audytorium* simple service for detecting events in the audience of a public space and the *sound_visualization* service for presenting the results of the analysis in a visual form.

5.4 User interface and client application

In order to facilitate the exploitation of the audio and video processing supercomputer services a client application has been created [20]. The application is called *KliK* and was developed with the participation of the author in Multimedia Systems Department. In

FIGURE 5.10: Screen from the client application - choice of sources

this section the parts of the client application which are related to the sound recognition services are briefly introduced.

Choice of sources The first step performed by the user to execute the sound recognition service is the choice of signal sources. In Figure 5.10 the example screen from the client application is presented. The map of Gdańsk University of Technology campus is shown with the locations of the microphones marked. The service can be executed on any of the available streams, also on all streams at once. If a camera is available close to the microphone, it can be used to provide video feed when an acoustic event is detected.

Configuring the sound recognition engine In order to start the service, the parameters of the sound recognition engine must be specified. The user can either start the service with default configuration, choose one of the available presets or adjust the

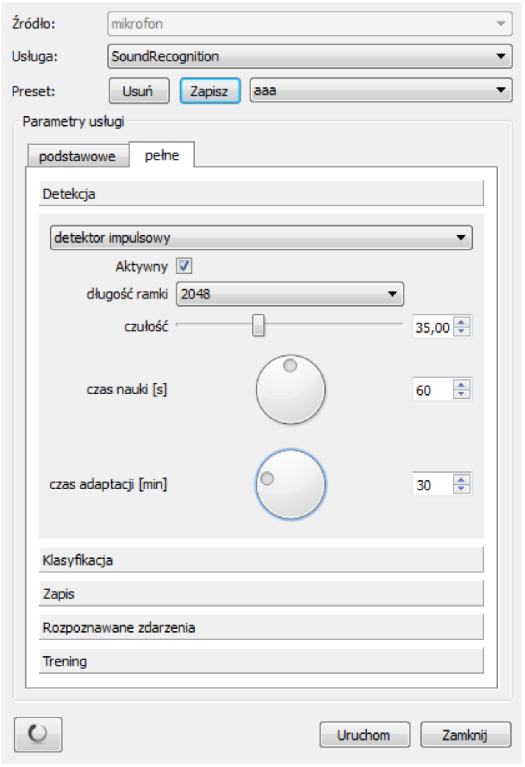


FIGURE 5.11: Screen from the client application - configuration of the service

parameters manually. In this mode, illustrated in Figure 5.11, the user can adjust the parameters concerning detection, classification, service output and more. In the figure the controls used for configuring the sound event detectors are shown. The temporal resolution (frame size), sensitivity, learning time and adaptation time can be finely tuned.

Presentation of results Once the service has been started from the client application's side, the results of the analysis can be observed in the following ways:

- by viewing the output stream of the *SoundRec_complex* service (according to Figure 5.8),
- by viewing the received events in the program window (see Figure 5.7),
- by visual notifications, i.e. flashing icons and visual feed from the cameras installed near microphones (if available).

Chapter 6

Evaluation on the training set

The methods for parameterization and classification of acoustic events featured in Chapter 4 are verified using a set of representative signals. The signals which are used for training the classifier are introduced. Subsequently, the optimum feature vector and the optimum parameters of the SVM classification model are sought. The obtained classification model is used in real conditions in the experiments outlined in the following chapters. Additionally, the results obtained in cross-validation on the training set are discussed.

6.1 Training signals

We use a database of self-recorded signals. The event samples were gathered during the work in various research projects conducted in both with and without the participation of the author of the dissertation. Five classes of events are considered: *explosion, broken glass, gunshot, scream* and *other*. The event samples were recorded in various conditions, both indoors and outdoors. The locations of the recordings include i.a. the outside and inside of the Faculty building, a street, a train station and the police training grounds (for gunshot and explosion recording). As far as gunshots are concerned, various types of weapons were used (both long- and short-barreled). For scream recording, calling for help in different languages (Polish, English, Spanish, Russian) was used. As far as scream is concerned, one has to distinguished between articulate speech and inarticulate

scream. The sound which people make in a dangerous situation may be both articulate (asking for help) and inarticulate (screaming in awe). In this work, speech is described by features which express the general properties of sound (see Section 4.3) and thus only the character of vocal activity is recognized, not the content. Both articulate and inarticulate scream are expected to share the same or similar spectral and temporal properties. In author's related work, a fuzzy classifier was employed to discern between different degrees of speech - from regular speech through raised voice - to scream [11]. To recognize the content of speech, a different approach should be used, i.e. the one known from the domain of speech recognition (see Section 2.2 for comparison between speech recognition and acoustic event recognition). Nevertheless, the content of speech is not considered in this thesis.

Sounds from the class *other* contain typical non-threatening events encountered in urban outdoor space and inside buildings. They were extracted from recordings of noise in such locations as: busy street in Gdańsk, railway station, university building, canteen and bank operating room. The types of events which are included in the *other* class are i.a.:

- vehicle noises registered in a busy street: e.g. trams, cars, trucks passing by;
- car alarms, horns, ambulance sounds etc.
- conversations of random people inside buildings;
- clanks and clatters of objects being put down or falling;
- doors closing, footsteps etc.

The signals were recorded using Bruel & Kjaer PULSE system with 65 kHz bandwidth and 24-bit depth. Next, the signals were converted to 48000 S/s and 32-bit floating point sample format. The summary of the training set is presented in Table 6.1. A total number of 1301 event recordings are used. Not all samples of each signal are utilized for classifier training. According to the principle described in Section 4.4, the short-time sample frames are extracted from the signal (here 200 ms long, with 50 % overlap). To avoid training the classifier against noise present at the beginning and the end of the

recording, only the sample frames whose energy exceeds the 0.05 of the maximum energy of each file are considered.

Table 6.1: Summary of the training set

no.	class	number of objects	number of frames	total length [mm:ss]
1	explosion	44	125	04:51
2	broken glass	193	452	19:43
3	gunshot	676	1370	66:56:00
4	scream	149	564	13:29
5	other	239	3694	60:22:00
	total:	**1301**	**6205**	**165:21:00**

In Figure 6.1 the example time domain forms and power spectra of the considered threatening events are depicted. The spectral representations were obtained from the whole signal by calculating the powers spectrum density estimate according to the Welch's method with 4096-point DFT and 50% overlap. It is visible in the plots that there are substantial differences in the shape of the events. Three of the hazardous event types (explosion, broken glass and gunshot) have impulsive character. The scream event has a sustain phase and is quasi-periodical. The shape of the power spectrum also enables distinction between the events. For explosions the energy drops quickly as frequency increases, whereas for broken glass it is sustained in the higher frequency range. In the spectrum of scream distinctive formants can be observed, in this case around 1000 Hz and 3000 Hz. The signal features should underline these differences in the properties of the signals from different classes.

Even though the respective events have different shape in both time and frequency domain, the distinction between them is not a trivial task. One of the reason is that it is easy to confuse a non-threatening event with a dangerous one. In a similar plot in Figure 6.2 the waveforms and spectra of non-threatening sounds are shown. Not that the stamping sound, depicted in Figure 6.2a, looks a lot like explosion. The sound of door closing, shown in Figure 6.2b, has similar properties to the sound of breaking glass. Next, if we judge only by the plots in Figure 6.1c and Figure 6.2c, it is not impossible to confuse the gunshot sound with the sound of a harmless object put down on desk. Finally, in Figure 6.2d the car horn sound is shown, which is similar to high-pitched human scream as far spectral and temporal properties are concerned. This example shows that the

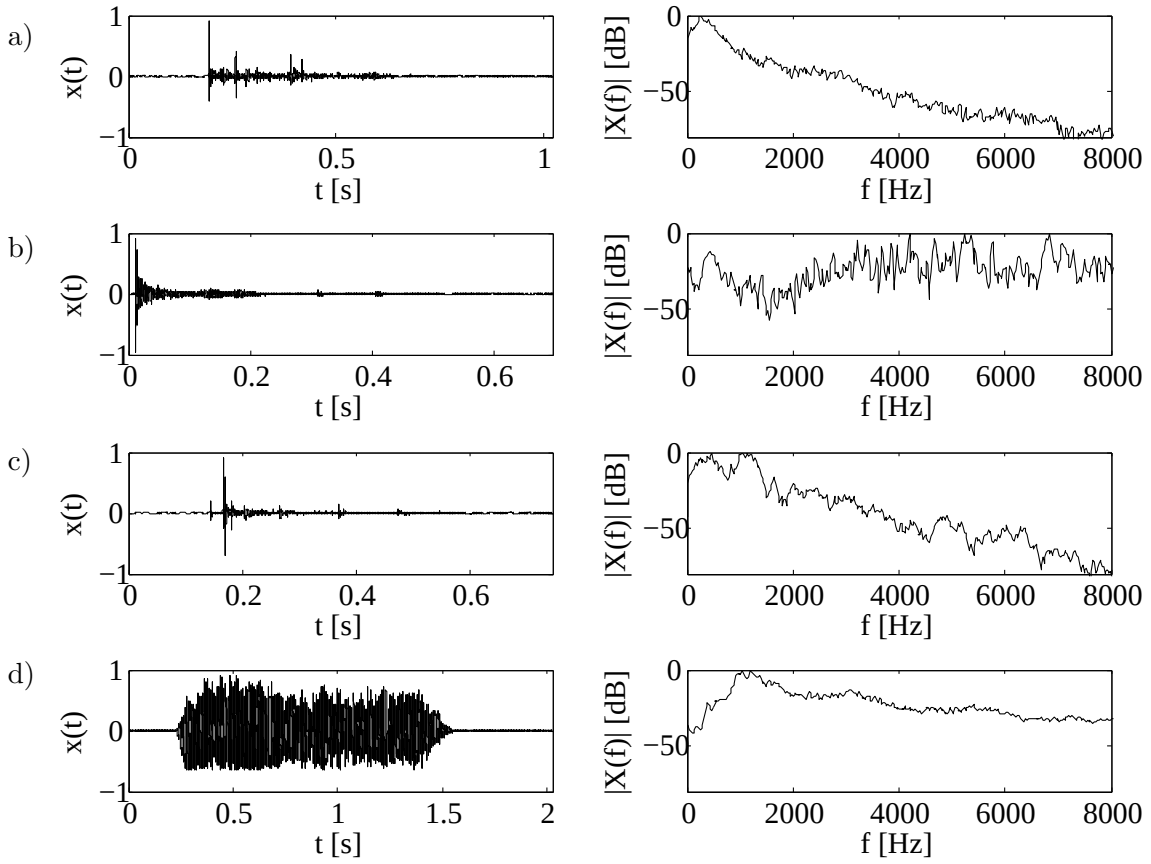


FIGURE 6.1: Example time-domain representations and power spectra of the hazardous events: a) explosion, b) broken glass, c) gunshot, d) scream

spectro-temporal representation of sounds can in some cases be ambiguous and that the choice of signal features should be very accurate to enable robust distinction between the respective event classes.

6.2 Features evaluation

In this section more attention is devoted to the signal features employed. First, examples of feature values for the respective event classes are evaluated. Subsequently, the experiment leading to an optimum feature vector is introduced.

6.2.1 Example feature values

To visualize the abilities of the considered signal features to discriminate between the respective event classes, 2D Cartesian plots are prepared. The four pairs of features

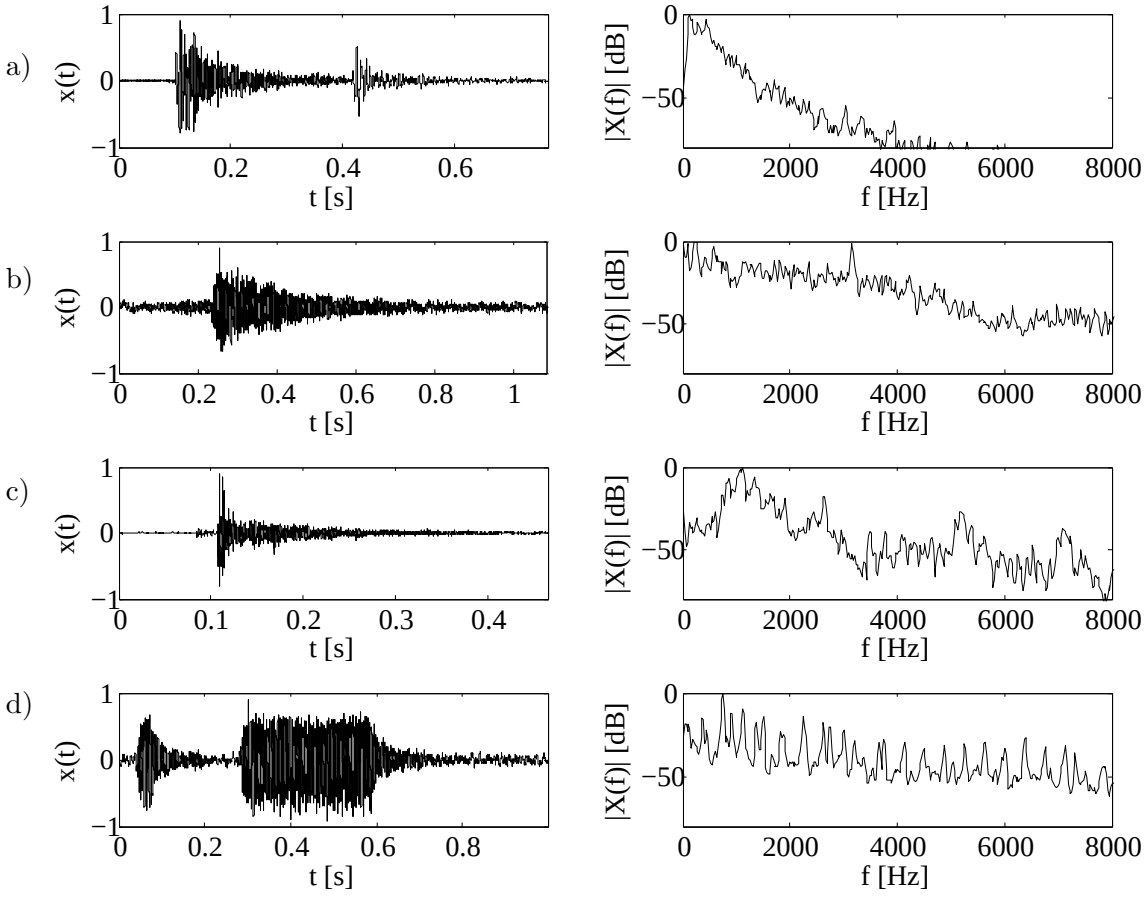


FIGURE 6.2: Example time-domain representations and power spectra of non-threatening events: a) object clatter, b) door, c) stamp, d) car horn

are depicted in Figure 6.3. For clear visualization, parameters from only 1 in 5 event samples are plotted. All parameter values are expressed in normalized scale [0;1] with respect to minimum and maximum value from the training set (see Section 4.3.4).

In a first pair, shown in Figure 6.3a, the features CF (Crest Factor - Equation 4.35) and ASC (Audio Spectrum Centroid - Equation 4.22) are compared. Explosions, gunshots and the sounds of breaking glass have lower CF, which shows their impulsive character. Screams on the other hand, yield higher ratios of *rms to peak*. The spectrum centroid feature - ASC - also shows some differences between the events. Note that explosions yield lower values of ASC than gunshots, since most of their energy is contained in the low-frequency regions. Broken glass sounds yield high ASC, since they are rich in high frequencies.

The features Zero Crossing Rate (ZCR) and Speech Energy (SPE), defined in Equation 4.38 and Equation 4.33, are shown in Figure 6.3b. It is visible that scream sounds yield higher values of SPE than other events, due to the fact that most of the scream's energy lies in the so-called speech frequency band (300 to 3400 Hz). The SPE parameter also enables some discrimination between gunshots and explosions. Note that gunshots also yield high values of SPE, since the energy of gunshots in more or less uniformly spread over all frequencies, whereas for explosions, most of the energy is in the low frequencies (see Figure 6.1). The ZCR parameter, on the other hand, yields higher values for screams than for gunshots, since the quasi-periodic signal of scream crosses 0 more often than the noisy sounds. Even higher values of ZCR are obtained for broken glass since these sounds are generally rich in high frequencies.

Another parameter which yields different values for noisy and harmonic events is spectral flatness. In a plot in Figure 6.3c the parameter SFMb2 (spectral flatness in band 1000 to 2000 Hz) is shown together with the spectral energy parameter, which expresses the ratio of energy accumulated in the band 1000-2000 Hz to the whole frequency spectrum. It is visible that scream events yield lower flatness than impulsive events like gunshots due to the comb-like properties of the power spectrum.

Finally, in Figure 6.3d, the parameters CF (Crest Factor - also *rms to peak*) and SE8 (spectral energy in range 100-500 Hz to range 7000-12000 Hz, see Table 4.2) are compared. This analysis shows the difference in the parameter values obtained form explosions and gunshots. While both these types of events yield low *rms to peak* ratio, explosion sounds have significantly more energy in the lower part of the spectrum, which is reflected by high values of the SE8 feature.

In a brief experiment the choice of auditory frequency scale is considered. Most of the published works utilize the mel frequency scale and MFCC coefficients [25, 34, 38, 40, 42, 52–55]. However, the Bark scale is also considered in some works. Dufaux utilized Bark spectrogram features and claimed that they yield comparable performance to mel-frequency features, MFCC being slightly better [65]. It was also shown by Shannon and Paliwal that bark scale features and mel scale features are equally efficient for automatic speech recognition [172].

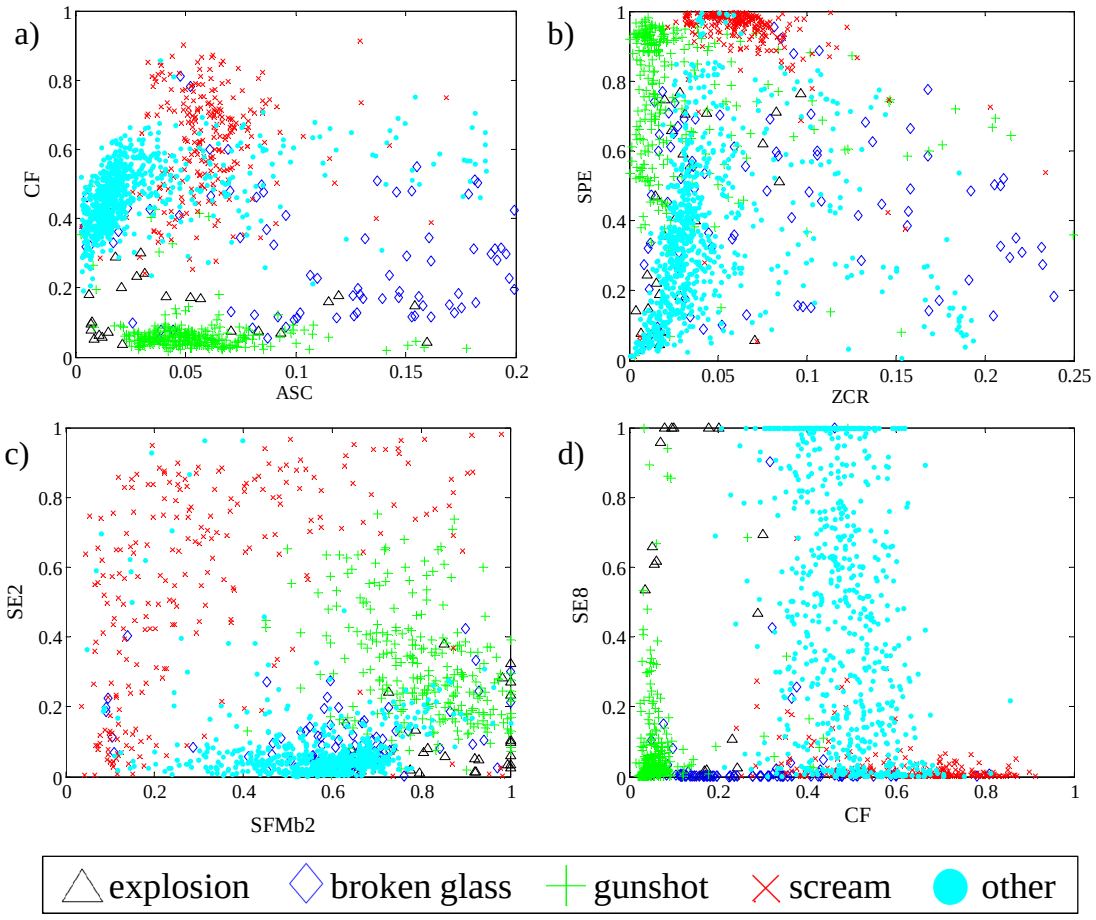


FIGURE 6.3: Example values of event parameters

A draft experiment to compare the Bark scale and mel-scale. The 16 MFCC parameters computed from the training set are considered and subsequently the mel scale is substituted by Bark scale. Thus, we compare the representation of the training signals in two frequency scales. The recognition results in 3-fold cross validation are evaluated. Mel-frequency features yield an overall accuracy of 84.93%, an average F1 score of 0.818 and $\kappa = 0.775$. The Bark scale features on the other hand yield accuracy equal to 74.25%, average F1 score of 0.673 and $\kappa = 0.625$. The results show that the mel scale performs better for the considered features and types of events. However, the aspect of auditory frequency scale should be evaluated more deeply in future research.

6.2.2 Feature selection results

The experiment for feature selection is conducted in two steps. In the first step, filter feature selection is used to find the best feature vector (see Section 2.3.2 for reference

on feature selection methods). It is temporarily assumed that a 50 element vector is used. The initial feature vector FV0 containing all 112 defined features is considered, as well as the vectors FV1 selected on the basis of χ^2 statistics, FV2 selected on the basis of information gain, and FV3 - obtained with SVM feature selection technique. The latter feature selection method was proposed by Guyon et al. with the participation of Vapnik [62]. It uses the weights assigned to individual features in the SVM optimization procedure as a criterion for ranking the features. Each feature vector is evaluated against the training set in 3-fold cross validation. The results are shown in Table 6.2. Classification accuracy and average F1-score are considered. It is visible that the vectors FV1 and FV2 deteriorate the classifier's performance, compared to the initial feature vector. The vector FV3 performs slightly better than the initial feature vector and therefore is the obvious choice. Two conclusions are drawn from this experiment. Firstly, it is visible that the information gain and χ^2 measures are not suitable for feature selection in this classification task and concerning the SVM classifier. In the process of feature elimination, some important information is discarded. The SVM feature selection technique does not worsen the classifier's performance. Secondly, there is only a minimal improvement after feature elimination. It shows that the SVM classifier is robust against redundancy in the feature vector. In fact, it can be said that a feature selection technique is already embedded in the SVM optimization (training) procedure. Feature selection is still profitable, though. As it will be shown later, feature extraction is the most costly operation in the engineered algorithm. Limiting the number of features reduces the computational requirements for this calculation step.

TABLE 6.2: Classifier's performance with different feature vectors

vector	description	accuracy	average F1-score
FV0	all 112 features (without elimination)	97.92%	0.972
FV1	50 top features according to χ^2	97.54%	0.958
FV2	50 top features according to information gain	97.16%	0.957
FV3	50 top features according to SVM selection	**98.23%**	**0.973**

In the next step, the optimum size of the feature vector is sought. We focus on the features selected by the SVM attribute elimination algorithm. The feature vector lengths from 70 to 10 are considered. The results of the classifier's performance in 3-fold cross validation mode are shown in Figure 6.4. The performance achieved with the 50 element

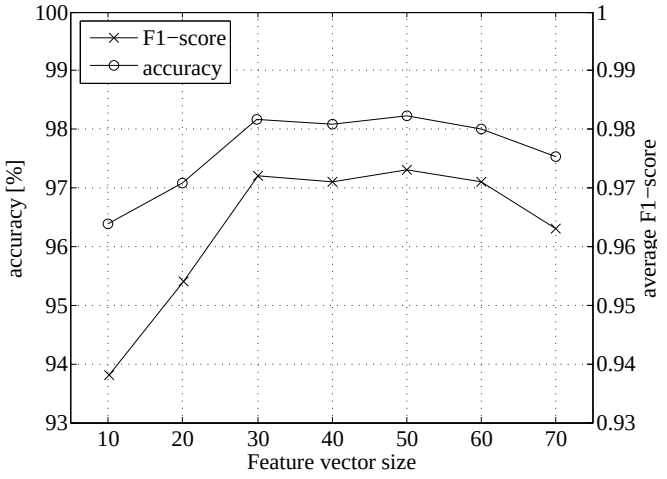


FIGURE 6.4: Classifier's performance vs. feature vector size

feature vector is the best (accuracy 98.23%, F1-score 0.973). The full list of selected parameters is given in Appendix A.

To illustrate the capability of the feature vector to discriminate between the considered classes of acoustic events, a visualization is featured. For the purpose of visualization the 50-element feature vectors are projected onto a 2D plane using Sammon mapping [173]. In Sammon's method nonlinear mapping (NLM) is used. Suppose there is a set of N vectors of L dimensions $\mathbf{x_i}$; $i \in \{1, 2, ..., N\}$. We perform the mapping of the vectors $\mathbf{x_i}$ to N corresponding vectors $\mathbf{y_i}$ in a space of 2 dimensions. The key of the Sammon's method is that the distance between vectors in reduced dimensionality space is kept as close as possible to the distance between vectors in the original space. The error of the mapping is defined by a formula [173]:

$$E = \frac{1}{\sum_{j=1}^{N}\sum_{i=1}^{j} {d_{ij}}^*} \sum_{j=1}^{N}\sum_{i=1}^{j} \frac{({d_{ij}}^* - d_{ij})^2}{{d_{ij}}^*} \tag{6.1}$$

where d_{ij} is the distance between vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ whereas ${d_{ij}}^*$ is the distance between vectors $\mathbf{y_i}$ and $\mathbf{y_j}$ and N is the number of data vectors. A Matlab implementation published by van der Maaten is used [63]. It utilizes gradient algorithm to find the parameters of the NLM algorithm which ensure the desired goal.

The results of Sammon mapping are shown in Figure 6.5. The training objects are

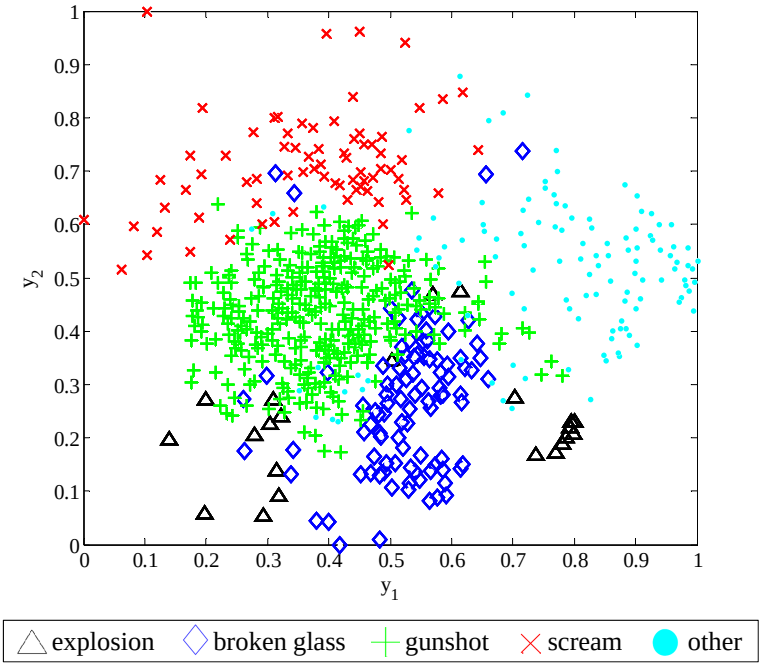


FIGURE 6.5: Results of Sammon mapping of the training set parameters

represented by two-element vectors (points) with coordinates $(y_1; y_2)$ in Cartesian space. The coordinates are normalized to the interval $[0; 1]$. The separation between the data points of different classes is visible, thus proving the features' ability to discern between the different types of hazardous events. In particular, the classes *gunshot, scream* and *other* are well separated. The explosion and broken glass data points are not that efficiently separable in 2D space. Please note that in Sammon's mapping the class labels are not taken into consideration. Therefore, the separation of data points in resulting 2D space is the result of the features' discriminative power, not the result of mapping.

6.3 Classifier validation

In this section the brief experiments aiming at choosing the best classifier parameters and the optimum feature vector are presented. The purpose is also to validate the SVM's ability to solve the given classification problem.

6.3.1 SVM parameters evaluation

The purpose of this particular test is to determine which values of the SVM model parameters are best for the given classification problem. The considered degrees of freedom, according to Section 2.5.3, are:

- kernel function,
- degree d,
- gamma γ,
- cost C.

The methodology is as follows. We utilize grid search technique to find the optimum kernel function, d, γ and C. Each time the classifier is evaluated in 3-fold cross validation fold against the training set, with a feature vector containing all defined parameters. We find it that accuracy, as defined in Equation 2.29, is not a good measure for assessment of classifier performance in this case. The number of vectors in each class is significantly different (see Table 6.1). Therefore, worse performance concerning the less numerous classes has a much weaker impact on accuracy than errors in the more numerous classes. For example, misclassification of 10 gunshots only lowers the gunshot recall rate by ca. 1.5%. On the other hand, 10 errors in the explosion class corresponds to nearly 1/4 of the entire class. However, in the terms of accuracy, both errors are equally significant. Hence, we use average F1-score rather than accuracy. The F1-score is calculated for all 5 classes according to Equation 2.31 and then the average value is computed.

Three kernel functions are considered: polynomial, RBF and sigmoid (see Equation 2.15, Equation 2.16 and Equation 2.17). Four values of d and γ and five values of C are used for comparison. For the polynomial kernel the degree was changed, whereas for the RBF and sigmoid kernels - γ was considered. The γ of the polynomial kernel function was set to 0.1 (which was found to be an optimum value in a brief experiment). The results are shown in Table 6.3. The best performance achieved is in bold font. The performance obtained with the sigmoid kernel is significantly worse. The polynomial and RBF kernel yield comparable results, yet the polynomial kernel is slightly better.

TABLE 6.3: Evaluation of SVM model parameters by means of average F1-score

	polynomial kernel			
	$d = 2$	$d = 3$	$d = 5$	$d = 8$
$C = 10^-2$	0.900	0.921	0.965	0.958
$C = 10^-1$	0.907	0.954	0.961	0.962
$C = 1$	0.954	0.963	**0.972**	0.968
$C = 10$	0.961	0.959	0.956	0.971
$C = 10^2$	0.955	0.959	0.959	0.954
	RBF kernel			
	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 10$
$C = 10^-2$	0.888	0.890	0.906	0.775
$C = 10^-1$	0.894	0.891	0.922	0.789
$C = 1$	0.889	0.912	0.950	0.839
$C = 10$	0.913	0.970	0.971	0.873
$C = 10^2$	0.954	0.962	0.968	0.861
	sigmoid kernel			
	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 1$
$C = 10^-2$	0.545	0.868	0.286	0.062
$C = 10^-1$	0.873	0.877	0.420	0.062
$C = 1$	0.874	0.881	0.435	0.062
$C = 10$	0.872	0.885	0.412	0.062
$C = 10^2$	0.872	0.884	0.369	0.062

6.3.2 Class probability thresholds

As it is explained in Section 4.4, the SVM classifier outputs the probabilities for each class. The decision is made based on these probabilities, according to the maximum probability rule. The predefined class probability thresholds influence the false positive and false negative rates. Here, an experiment is introduced to establish the optimum values of probability thresholds for each class.

The classifier is tested in 3-fold cross validation mode on the whole training set. For each class, the probability threshold is set to the following values: from 0.05 to 0.85 with 0.05 step and from 0.86 to 0.98 with 0.02 step. Then, the false positive (FP) and false negative (FN) rates for this class are computed. These points are plotted in Figure 6.6.

The data in the DET plots are scattered, due to the following reasons. Firstly, adjusting the class threshold does not only affect the results for the considered class, but also for other classes. Secondly, the number of objects in some classes is relatively small. Note that the data obtained for gunshots (676 objects) show a lot more regularity than

those obtained for explosions (44 objects). Thirdly, there is some randomness in the cross-validation operation, due to random division into training and testing set.

To obtain the DET (Detection Error Tradeoff) curves, trend estimation is performed. The least squares method is adopted. The coefficient of determination R^2 is used as a measure of goodness of fit. The R^2 measure is defined as [174]:

$$R^2 = 1 - \frac{SS_{req}}{SS_{res}} \tag{6.2}$$

where SS_{reg} is the explained sum of squares and SS_{res} is the residual sum of squares. To introduce the formulae for sums of squares, let us consider the observed data points y_i and the estimated data points $\hat{y_i}$ where $i \in \{1..N\}$ and N is the number of data points. The residual sum of squares is defined as:

$$SS_{res} = \sum_{i=1}^{N} (y_i - \hat{y_i})^2 \tag{6.3}$$

and the explained sum of squares is defined as:

$$SS_{reg} = \sum_{i=1}^{N} (\hat{y_i} - \bar{y})^2 \tag{6.4}$$

where $\bar{y}$ is the arithmetical mean of y. The trend functions considered were: exponential, logarithmic, linear, power and polynomial. It is established that the best fit is achieved with the power function:

$$y = a \cdot x^b \tag{6.5}$$

where the coefficients a and b are determined during the least squares optimization procedure and the variables x, y correspond in this case to false positive and false negative rate respectively. The obtained trend lines are shown in Figure 6.6 as solid lines.

The trend lines in the plots in Figure 6.6 enable the determination of equal error rates (EER). Points of equal error are indicated by a dashed line. By definition, the EER is

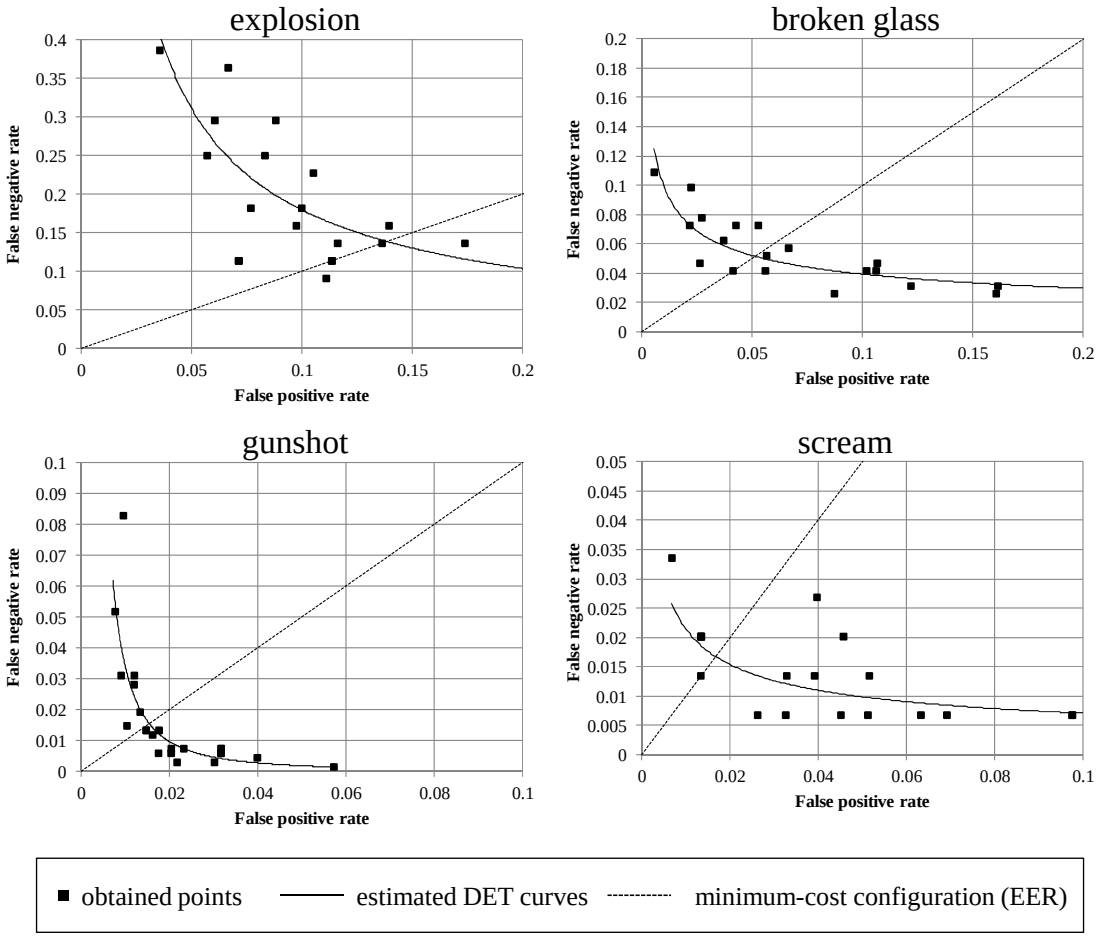


FIGURE 6.6: DET curves for classification of the events from the training set.

achieved when the false positive and false negative rates are equal. On the plot, it is the point in which the solid line crosses the dashed line. The approximate EERs obtained are: 0.13 for explosion, 0.05 for broken glass, 0.015 for gunshot and 0.017 for scream. The class thresholds which yield those EERs are considered optimum and are equal to: 0.1 for explosion, 0.45 for broken glass and 0.75 for both gunshot and scream.

It should be noted that the EER is a point for which the loss of the decision system is at a minimum, although it allows for missing some hazardous events. The classifier could be configured in such a way that no threatening event would be missed. However, it would in exchange yield a vast number of false positives. Therefore, even though the recall rate of a certain threatening event would be close to 1, the overall loss achieved in the classification process would be enormous.

6.3.3 Performance on the training set

The final classification configuration for this problem is:

- polynomial kernel
- degree $d = 5$
- gamma $\gamma = 0.1$
- cost $C = 1$
- feature vector comprising 50 elements

With these parameters, the classifier is tested against the training set in 3-fold cross validation procedure. The obtained results are shown in Table 6.4. The achieved performance is very good. Minor errors are observed, mostly concerning classifications of other events as dangerous events (false positives). False negatives (i.e. assignments of threatening sounds into *other* class) are less frequent.

The example false positive classifications are results of the following errors:

- ambulance siren or car alarm classified as scream,
- whistle classified as scream,
- loud clatter classified as gunshot,
- machine hiss classified as broken glass.

Such errors could be eliminated by gathering more training examples for the classifier. The more signals similar to the above there are in the training set, the less likely they are to result in an incorrect classification.

To investigate the false positive classification results, an analysis of feature values is presented. In Figure 6.7 the results of Sammon mapping of the training vectors are depicted. In addition to the points shown in Figure 6.5, the false positive classifications are marked by circles and the classes assigned to these events are printed. Each circled

TABLE 6.4: Evaluation of the classifier on the training set in 3-fold cross validation - confusion matrix

	explosion	glass	shot	scream	other	**recall**	**precision**	**F1-score**
explosion	**43**	0	1	0	0	0.977	0.935	0.956
glass	1	**187**	2	0	3	0.969	0.969	0.969
shot	1	0	**675**	0	0	0.999	0.990	0.994
scream	1	0	0	**147**	1	0.987	0.980	0.983
other	0	6	4	3	**226**	0.946	0.983	0.964
						accuracy: 1278/1301 [98.23%]		
						average F1-score: 0.973		
						κ: 0.973		

cyan dot represents an event from class *other*, which was erroneously classified as a threatening sound. It can be seen that the false positive results are obtained for vectors of the class *other* which lie close to threatening events in the transformed 2D space. Due to the nature of Sammon mapping we deduce that these vectors are also close to those classes in the original feature space. In practice it may mean that a non-threatening event resembles a threatening one, as far as the considered sound parameters are concerned. Such resemblance in terms of spectro-temporal properties has already been indicated in Figure 6.2. To reduce the number of such false positive classifications, a closer investigation of the feature values and appropriate feature selection should be carried out in future work.

Basing on the presented results, it can be stated that the analyzed hazardous acoustic events can be correctly discerned using the adopted methods. The results obtained on the training set will be used as reference in further experiments. The sounds on which the classifier is tested in the experiments described in this chapter, are considered *clean*. In Chapter 7 the evaluation of the recognition engine in noisy conditions is provided.

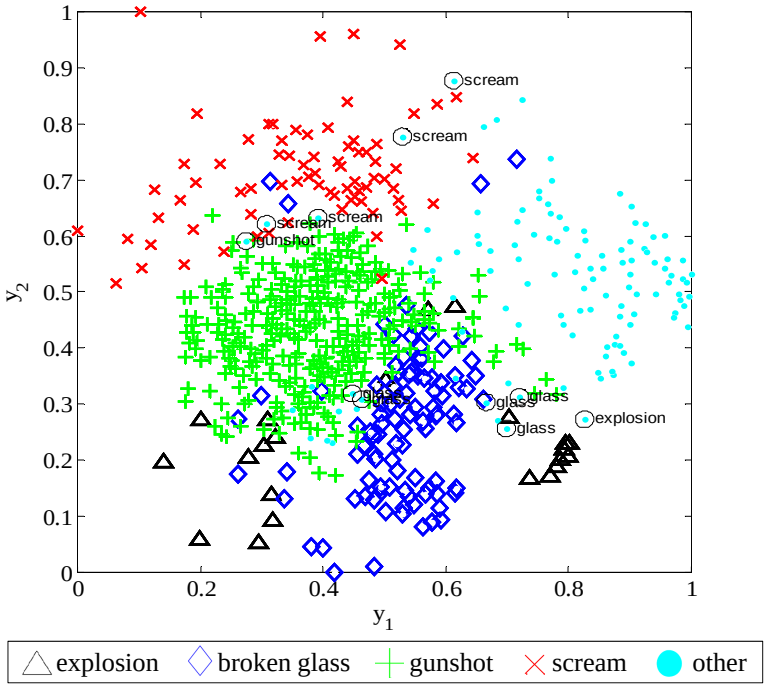


FIGURE 6.7: Results of Sammon mapping of the training set parameters with false positives indicated by circles

Chapter 7

Evaluation on noisy data

The evaluation presented in Chapter 6 concerned the classification of clean signals, without additional noise. Deployment of the sound recognition engine in real conditions raises new challenges. The noise present in the environment adds to the signal and influences the values of the signal features. Such unpredictability can lead to false detections. Moreover, in open acoustic space a number of phenomena are present which influence the acoustic wave emitted by the event source. Some of these phenomena are outlined in Section 2.8. Hence, we can anticipate that the performance of the algorithms in real conditions will probably be worse than the performance on the training set. The aim of the experiments presented in this chapter is to evaluate this performance and to determine how different types of noise and different environmental conditions influence the efficiency of sound event recognition. Basing on each experiment, the conclusions regarding the possibility of practical application of the engineered methods are drawn. The chapter is concluded by a series of general findings derived from the practical experiments.

7.1 Evaluation in simulated conditions

In order to properly evaluate the influence of noise on the performance of the sound recognition engine, the type and intensity of noise should be known and controllable. In real life conditions it is very difficult to achieve. If the experiment is conducted

in real life environment, the experimenter has no control over the level of background noise. It is only possible, in some cases, to control the intensity of the acoustic events. Nevertheless, the estimation of the *Signal-to-Noise Ratio* (SNR) would be prone to significant errors. Hence, we propose a simulation environment which recreates the acoustic ambiance of selected real-life surroundings and makes it possible to control and estimate the SNR with adequate precision. The experiment and most of the figures and tables featured in this section are reported in a related publication [9]. The setup of this simulated environment, acoustic signals employed, experimental methodology and results are presented in the following subsections.

7.1.1 Setup of the test environment

The setup of the measurement equipment employed in the experiment is presented in Figure 7.1. In an anechoic chamber, 8 REVEAL 601p speakers, an Acoustic Vector Sensor (AVS) and a type 4189 measurement microphone by Bruel & Kjaer (B&K) were installed. The USP probe was fixed 1.37 meters above the floor. The measurement microphone was placed 5 mm above the AVS. In the control room a PC computer with Marc 8 Multichannel audio interface was used to generate the test signals and record the signals from the AVS. The signals from the AVS were used to evaluate the accuracy of sound source localization. Two SLA-4 type 4-channel amplifiers were employed to power the speakers. Channels 1-4 and 5-8 were connected to respective speakers in the 8-speaker matrix. In addition, PULSE system type 7540 by B&K is used to record the acoustic signals. The PULSE measuring system was calibrated before the measurements using a type 4231 B&K acoustic calibrator. Part of the equipment utilized in the experiment can be seen in the photograph in Figure 7.2. Five of eight loudspeakers, arranged in a circle, the AVS and measurement microphone are visible.

7.1.2 Test signals

Audio events were combined into a test signal consisting of 100 events, randomly placed in time, 20 examples of each of the 5 classes (explosion, breaking glass, gunshot, scream, other). The average length of each event equals 1 second, and there is a 10 second space

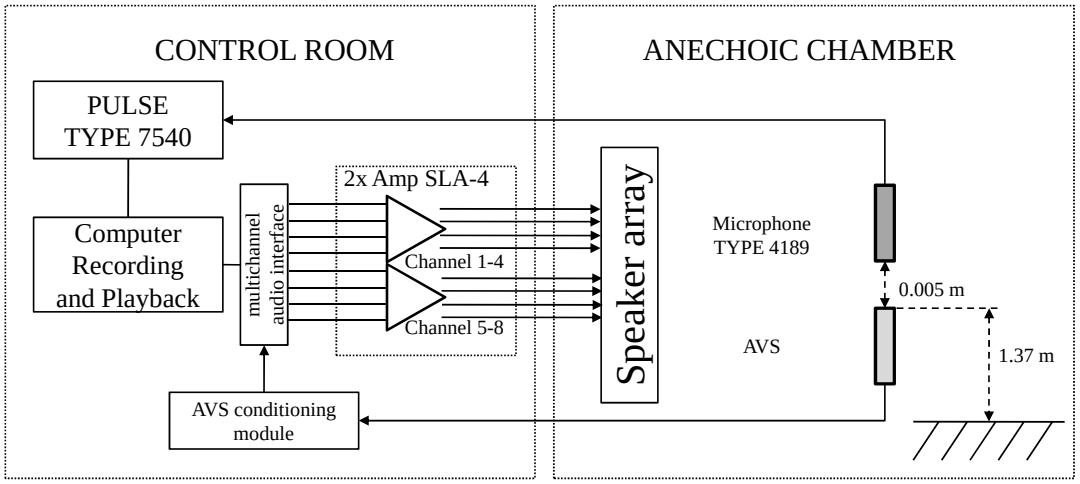


FIGURE 7.1: Setup of the experiment for testing the sound recognition engine in simulated conditions

FIGURE 7.2: Photograph of the equipment employed for testing the sound recognition engine in simulated conditions

between the start and end of adjacent events. The length of the test signals equals 18 min 20 s. Four disturbing signals were prepared, each with a different type of noise:

- traffic noise, recorded in a busy street in Gdansk;
- cocktail-party noise, recorded in a university canteen;
- railway noise, recorded in Gdansk railway station;
- indoor noise, recorded in the main hall of Gdansk University of Technology.

7.1.3 Experimental methodology

In the test signals the events are randomly assigned to one of four channels: 1,3,5,7. The order of the events with the numbers of channels they are emitted from and classes they belong to are stored in the Ground Truth (GT) reference list. At the same time, the other channels (2,4,6,8) are used to emit noise. Each noise channel is shifted in time to avoid correlation between channels. The gain of the noise channels is kept constant, while the gain of events is set to one of four values: 0 dB, -10 dB, -20 dB and -30 dB. This yields 16 recordings of events with added noise (4 types of noise x 4 gain levels). In addition, the signals of four types of noise without events and 4 signals of events without noise with different gain levels are recorded. These events are used to measure the instantaneous SNR. On the whole 24 signals have been gathered. The total length of the recordings equals 7 h 20 min. The summary of the recordings is presented in Table 7.1. The recording of events at -30 dB with indoor noise (no. 24) was later excluded from analysis due to too low level of SNR.

In recordings 9-24 the noise is added to the events acoustically. Each event in this recording has a unique SNR. We are able to measure this SNR by comparing the energy of clean events (in recordings 1-4) and noise (in recordings 5-8). The SNR is calculated for each event according to the formula in Equation 7.1:

$$SNR = log_{10} \frac{\sum_{m=m_1}^{m_2} s[m]^2}{\sum_{m=m_1}^{m_2} n[m]^2} \tag{7.1}$$

where $s[m]$ and $n[m]$ are the signals with event and noise respectively and $[m_1; m_2]$ is the range of samples in which the event is present. Thus, the SNR is calculated in the whole frequency spectrum and with varying time constant. To evaluate the influence of SNR on the detection and classification metrics, we sort the events with respect to SNR. The SNR values are then divided into 8 intervals. The limits of these intervals and the number of events found in the respective SNR intervals are shown in Table 7.2.

TABLE 7.1: List of recordings performed in simulated conditions

no.	**recording**	**events gain**	**number of events**	**time [hh:mm:ss]**
1	Events without noise	0	100	00:18:20
2	Events without noise	-10	100	00:18:20
3	Events without noise	-20	100	00:18:20
4	Events without noise	-30	100	00:18:20
5	traffic noise only			00:18:20
6	cocktail-party noise only			00:18:20
7	railway noise only			00:18:20
8	indoor noise only			00:18:20
9	events with traffic noise	0	100	00:18:20
10	events with traffic noise	-10	100	00:18:20
11	events with traffic noise	-20	100	00:18:20
12	events with traffic noise	-30	100	00:18:20
13	events with cocktail-party noise	0	100	00:18:20
14	events with cocktail-party noise	-10	100	00:18:20
15	events with cocktail-party noise	-20	100	00:18:20
16	events with cocktail-party noise	-30	100	00:18:20
17	events with railway noise	0	100	00:18:20
18	events with railway noise	-10	100	00:18:20
19	events with railway noise	-20	100	00:18:20
20	events with railway noise	-30	100	00:18:20
21	events with indoor noise	0	100	00:18:20
22	events with indoor noise	-10	100	00:18:20
23	events with indoor noise	-20	100	00:18:20
24	events with indoor noise	-30	100	00:18:20
		total	1600	**07:20:00**

TABLE 7.2: Number of events in assumed SNR intervals

SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**	**sum**
explosion	11	35	23	38	30	43	24	96	300
broken glass	7	33	27	36	33	40	29	95	300
gunshot	2	21	15	35	26	47	29	125	300
scream	13	16	15	12	31	21	42	150	300
other	5	15	6	26	21	39	26	162	300
sum	38	120	86	147	141	190	150	628	1500

The measures of detection accuracy are the *True Positive* (TP), and *False Positive* (FP) rates. The TP rate equals the number of detected events which match the events in the GT list divided by the total number of events in the GT list. The matching of event is understood as the difference between detection time and GT time of the event being not greater than 1 second. A FP result is considered when an event is detected which is not listed in the GT reference and is classified as one of the four types of event that are considered alarming (classes 1–4 in Table 6.1). The assumed measures of classification accuracy are precision and recall rates. For more details concerning the evaluation metrics, please refer to Section 2.7.

The direction of arrival of the acoustic wave was computed as a product of sound pressure and particle velocity components received at the AVS, according to the methodology presented in Section 2.6. The output of the AVS is 4 synchronized signals: acoustic pressure p_a and particle velocity in three orthogonal directions u_x, u_y, u_z [7]. The information about the air particle velocity components is used to calculate the sound intensity vector $\mathbf{I}$, as defined in Equation 7.2:

$$\mathbf{I} = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \sum_{n=1}^{N} p_a[n] \cdot u_x[n] \\ \sum_{n=1}^{N} p_a[n] \cdot u_y[n] \\ \sum_{n=1}^{N} p_a[n] \cdot u_z[n] \end{bmatrix} \tag{7.2}$$

where N equals the number of samples in the analyzed frame. Here $N = 4096$ samples (85 ms at 48000 samples per second). The direction of the sound intensity, received at the AVS, is identical with the direction of arrival of the acoustic wave. Hence, it points to the source of sound. Thus computed Cartesian coordinates can be transformed into the polar coordinate system in order to express the direction of arrival as azimuth ϕ and elevation θ.

$$\begin{cases} I = \sqrt{I_x^2 + I_y^2 + I_z^2} \\ \phi = arctan\frac{I_y}{I_x};\ 0 < \phi < 2\pi \\ \theta = arcsin\frac{I_z}{I};\ -\frac{\pi}{2} < \theta < \frac{\pi}{2} \end{cases} \tag{7.3}$$

The ground truth values of the angle ϕ between the loudspeakers which were used to emit the sound events and the AVS were measured and stored. The localization accuracy is evaluated in terms of error of azimuth angle, i.e. the difference between the measured value and the ground truth value:

$$\Delta\phi = \phi_{meas} - \phi_{GT} \tag{7.4}$$

The elevation angle is not considered in this evaluation due to the setup of the experiment, namely all sources being located roughly on the same plane.

7.1.4 Results

Detection results

First, we present the average results of event detection for all noise types. The TP rates of each detection algorithm with respect to SNR are plotted in Figure 7.3. Please note that the displayed SNR value corresponds to the upper limit of the interval, e.g. the label 5 dB denotes the interval (0 dB ; 5 dB]. The combination of all detection algorithms yields high detection rates. The TP rate decreases significantly with the decrease of SNR. The algorithm which yields the highest detection rates in good conditions (SNR $>$ 10 dB) is the *Impulse Detector*. It outperforms the other algorithms, which are more suited to specific types of signal. However, the *Impulse Detector* is most affected by added noise, since it only reacts to the level of the signal. Other algorithms, namely *Speech Detector* and *Variance Detector*, maintain their detection rates at a similar level while SNR decreases. It is a good feature, which allows the detection of events even if they are below the background level (note the TP rate of 0.37 for SNRs smaller than -5 dB). It is also evident that the combination of all detectors performs better than any of them alone, which proves that the engineered detection algorithms react to different features of the signal and are complementary. The *Histogram Detector* is disappointing, since its initial TP rate is the lowest of all detectors and falls to nearly 0 at 5 dB SNR. The total number of detected events equals 1055 out of 1500 (for all SNRs combined) which yields an average TP rate of 0.7.

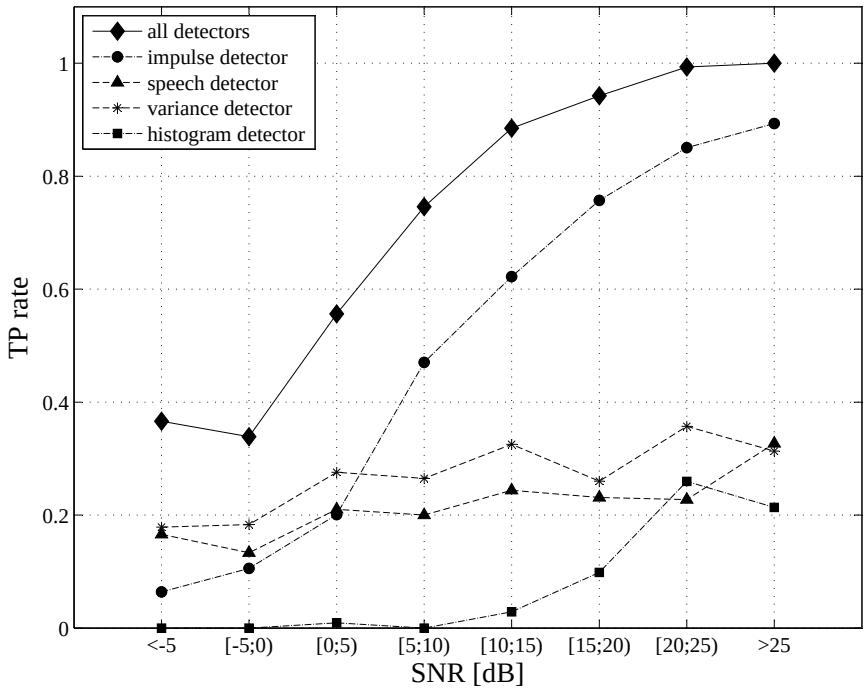


FIGURE 7.3: Averaged results of event detection in simulated conditions

The next analysis allows us to examine how different detectors react to different types of event. In Figure 7.4 the TP rates of the detectors for different event class are shown. The average results for all SNR values are presented. The presented dependencies once again prove that the developed detection algorithms complement one another and are well suited to recognizing specific types of events. The *Speech Detector* reacts to tonality which is present in screams, while *Variance Detector* reacts to sudden changes of spectral features related to the event of breaking glass (see Section 4.1.2). It proves the assumptions made while designing the detectors.

Subsequently, the evaluation of event detection in the presence of different noise types is presented in Figure 7.5. The average results for all SNR values are shown. On average, the detectors perform best in the presence of cocktail-party noise. The worst detection rates are achieved in the simulated indoor environment. It can also be observed that some classes of acoustic events are strongly masked by specific types of noise. Gunshots, for example, have a TP rate of 0.45 in the presence of traffic noise and 0.74 in the presence of railway noise.

A very important aspect, as far as sound event detection is concerned, is false alarms. In our experiment a detection is treated as a FP value when the detected event was

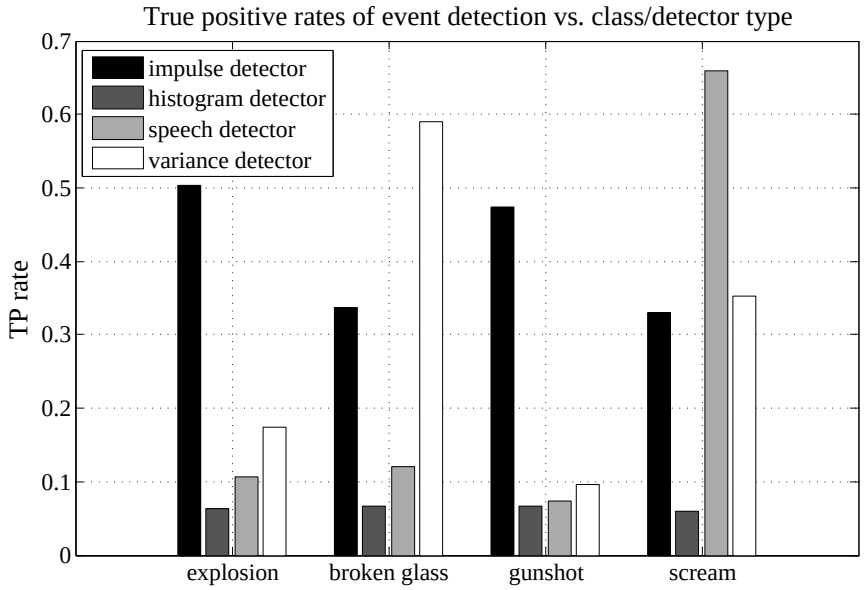


FIGURE 7.4: Results of event detection in simulated conditions for different detection algorithm and noise type

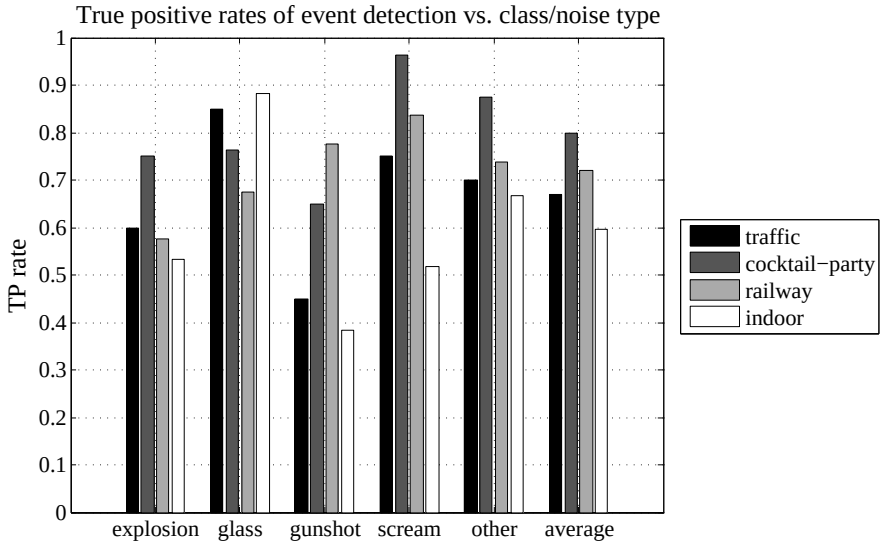


FIGURE 7.5: Results of event detection in simulated conditions for different event class and noise type

not present in the Ground Truth reference list and is recognized as one of the classes related to danger (classes 1–4). The number of false alarms produced by each detection algorithm and the classes that are falsely assigned to them are presented in Table 7.3. The presented FP rates are calculated with respect to the total number of events detected by the specific detector, for all SNR levels. It can be seen that *Speech Detector* and *Impulse Detector* produce the majority of the false alarms. The fact is understandable, since these algorithms react to the level of the signal and to tonality. Sudden changes in the signal's level and tonal components appear in the acoustic background frequently. The lowest FP rate is achieved by the *Histogram Detector*, however it also yields the lowest TP rate. The *Variance Detector* achieves satisfactory performance, as far as FP rate is concerned. It is a good feature, demonstrating the fact that its TP rate is robust against noise. The overall FP rate equals 0.08, which can be regarded as good performance. In this experiment the relation between FP and TP detections is not studied. We investigate the DET (Detection Error Tradeoff) curves of the detectors and attempt to lower the FP rate in Section 7.2.2.

TABLE 7.3: False positive detections in simulated conditions

	impulse detector	**histogram detector**	**speech detector**	**variance detector**	**all detectors**
explosion	12	0	1	0	**13**
broken glass	26	0	27	8	**50**
gunshot	12	0	7	1	**20**
scream	3	1	9	1	**9**
sum	53	1	44	10	**92**
FP rate	0.07	0.01	0.12	0.02	**0.08**

Classification results

The adopted measures of classification accuracy, i.e. precision and recall rates, are calculated with respect to SNR. The results are presented in Figure 7.6. The general trend observed is that the recall rate descends with the decrease in SNR. It can be seen, as far as explosion and broken glass are concerned, that the precision rate ascends with the decrease in SNR. In very noisy conditions these classes are recognized with greater certainty. The class of event which is least affected by noise is broken glass. The recall rate remains high (ca. 0.8 or more) for SNRs greater than or equal to 5dB.

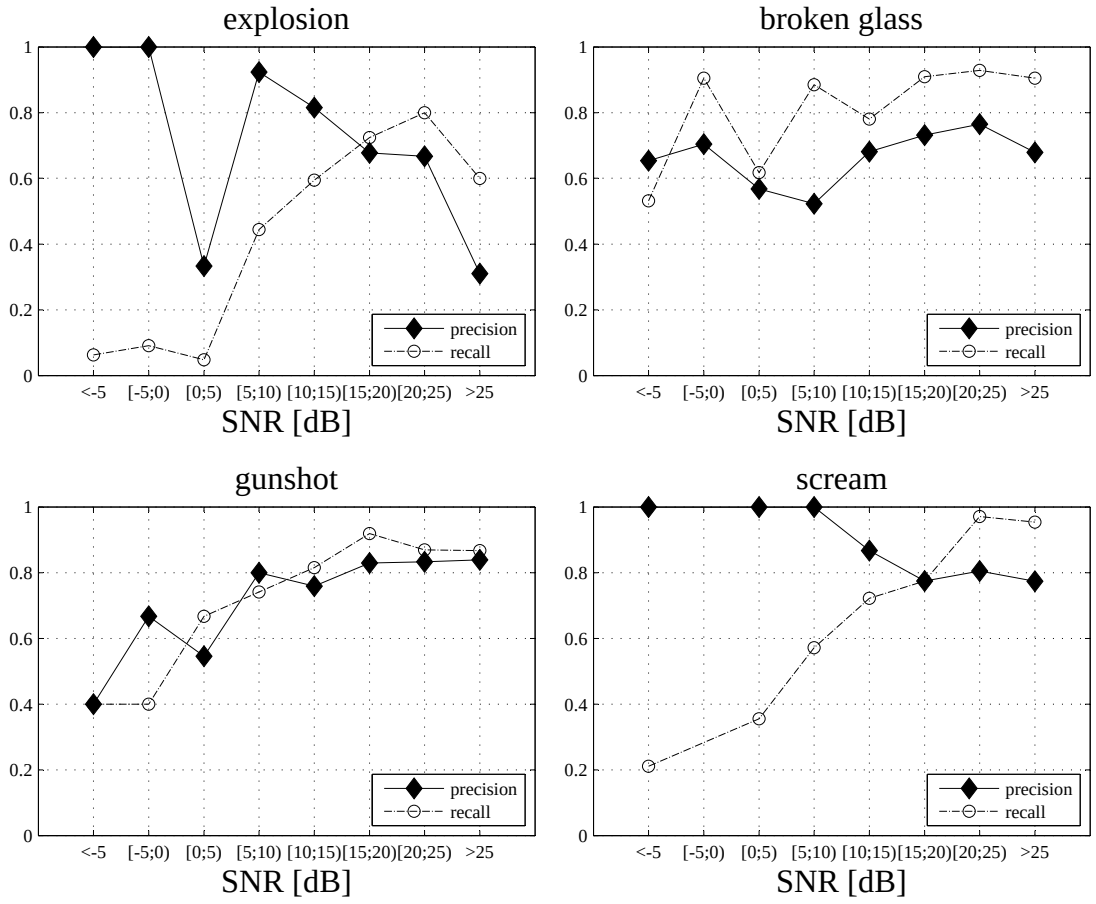


FIGURE 7.6: Precision and recall rates achieved in simulated conditions

The low overall recall rate of explosions is caused by the fact that the events were reproduced through loudspeakers, which significantly changes the characteristics of the sound. This aspect is discussed further in the concluding paragraph. To examine the event classification more thoroughly, we present more data. In Table 7.4 and Table 7.5 two confusion matrices are presented – at 20dB and at 0dB SNR respectively. It is apparent that when the noise level is high, the threatening events are often confused with other, non-threatening events. The errors between the classes of hazardous events are less frequent. It can also be seen that at 20dB SNR there are frequent false alarms, especially falsely detected explosions (in 10 cases) and screams (8 cases). In audio surveillance, however, such false alarms should always be verified by the human personnel, therefore such error is not as grave as classifying a hazardous event as non-threatening (false rejection).

TABLE 7.4: Confusion matrix obtained in simulated conditions at 20 dB SNR

	explosion	glass	shot	scream	other	recall	precision	F1-score
explosion	**24**	2	1	0	3	0.800	0.667	0.727
glass	0	**26**	0	0	2	0.929	0.765	0.839
shot	1	1	**20**	0	1	0.870	0.833	0.851
scream	1	0	0	**33**	0	0.971	0.805	0.880
other	10	5	3	8	**12**	0.316	0.667	0.429
						accuracy: 115/153 [75.16%]		
						average F1-score: 0.745		
						κ: 0.696		

TABLE 7.5: Confusion matrix obtained in simulated conditions at 0 dB SNR

	explosion	glass	shot	scream	other	recall	precision	F1-score
explosion	**1**	5	1	0	14	0.048	0.333	0.083
glass	0	**21**	2	0	11	0.618	0.568	0.592
shot	0	0	**6**	0	3	0.667	0.545	0.600
scream	0	6	1	**11**	13	0.355	1.000	0.524
other	2	5	1	0	**16**	0.667	0.281	0.395
						accuracy: 55/119 [46.22%]		
						average F1-score: 0.439		
						κ: 0.309		

Moreover, the Cohen's κ metrics achieved at different SNR levels is examined. In Figure 7.7 the relation between the measured SNR and obtained κ is presented. The similar trend is visible as observed for the precision and recall rates. The greater the SNR, the higher the κ score. The maximum value of κ obtained equals 0.696 for the SNRs between 20 and 25 dB. It can be noted that for SNRs greater than 25 dB the measure drops slightly. This irregularity is caused by the fact that the events were not evenly distributed with respect to SNR. As it can be seen in Table 7.2, the events with the highest SNR are in this experiment mostly *scream* and other events, whereas in other SNR intervals impulsive events were more frequent.

The other aspect of the experiment is the performance of classification in the presence of different types of noise. The results have been gathered in and in Table 7.6 (recall rates) and Table 7.7 (precision rates). It can be observed that different types of noise affect the results of classification of some event types more than others. The recall rates of screams is e.g. lower in the presence of cocktail-party and railway noise compared to traffic and indoor noise. This is caused by the spectral content of these types of noise,

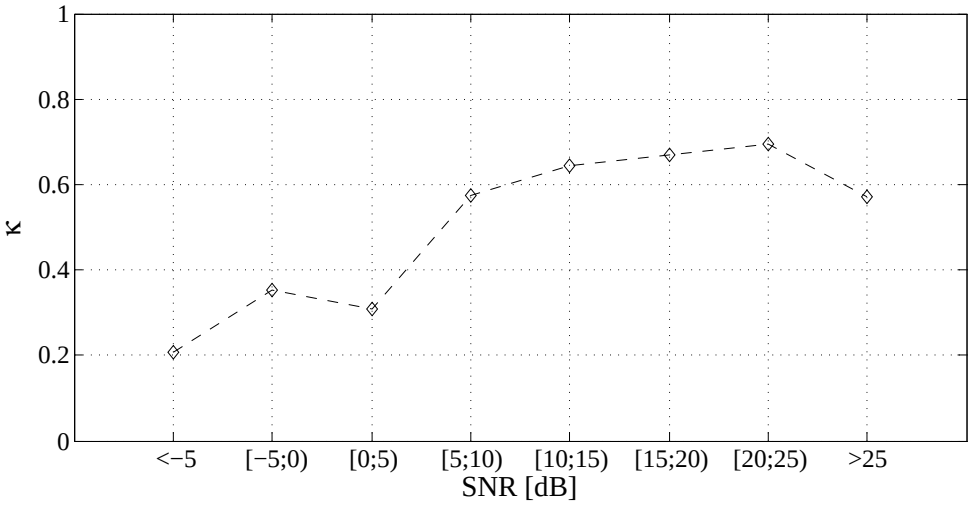


FIGURE 7.7: Relation between measured SNR and obtained κ measure

TABLE 7.6: Recall rates obtained in simulated conditions for different noise type and event class

traffic noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0	0.2	0.64	0.88	0.78	0.5
broken glass	0.55	0.83	0.7	1	0.75	1	1	1
gunshot	0	0	1	0.75	1	1	0.92	0.8
scream	0.22	0	0.71	0.29	0.86	0.67	1	0.91
other	0	1	0.5	0.62	0.71	0.38	0.33	0
cocktail-party noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0	0.33	0.3	0.7	0.7	0.6
broken glass	1	1	0.8	0.86	1	0.82	1	0.92
gunshot	0	0	0.67	0.71	0.83	1	0.88	1
scream	0	0	0.18	0.44	0.64	0.67	1	1
other	0	1	0.82	0.63	0.92	0.25	0.42	0.11
railway noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0.17	0.33	0	0.17	0.63	0.57	0.9	0.67
broken glass	0.2	0.86	0.25	0.75	0.5	0.91	0.75	0.8
gunshot	0.57	0.44	0.5	0.85	0.67	0.81	0.67	0.78
scream	0.14	0	0.13	0.71	0.5	0.86	0.92	1
other	0.6	1	0.67	0.83	0.67	0.43	0.31	
indoor noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0.25	1	0.88	0.75	1	0
broken glass	0.54	1	0.6	1	0.79	1		
gunshot	0	0	0.67	0.33	0.73	1		
scream	0.5	0	0.6	1	1	1	1	
other	1	1	0.4	0.67	0.54	0.2	0.14	0.67

which are rich in frequency components that are important for recognizing screams (i.e. high and mid-high frequencies). Generally, traffic noise yields the highest recall rates. A possible reason is that the training samples were recorded in the vicinity of a street.

TABLE 7.7: Precision rates obtained in simulated conditions for different noise type and event class

traffic noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0	1	1	0.64	0.78	0.14
broken glass	0.46	0.63	0.54	0.47	0.75	0.73	0.79	0.5
gunshot	0	0	0.5	0.75	0.78	1	0.92	1
scream	1	0	1	1	0.86	0.75	1	0.67
other	0	0.33	0.17	0.57	0.56	0.63	0.67	
cocktail-party noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0	1	1	0.7	0.78	0.43
broken glass	1	0.5	0.5	0.43	0.64	0.69	0.82	0.86
gunshot	0	0	0.67	0.83	0.83	0.71	0.88	0.79
scream	0	0	1	1	1	0.86	0.79	0.77
other	0	0.2	0.36	0.38	0.61	0.67	0.63	0.67
railway noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	1	1	0	1	0.83	0.8	0.64	0.29
broken glass	0.33	0.86	0.17	0.67	0.33	0.83	0.67	0.67
gunshot	0.57	1	0.5	0.79	0.5	0.87	0.5	0.88
scream	1	0	1	1	0.83	0.67	0.85	0.92
other	0.17	0.14	0.29	0.45	0.57	0.43	0.67	
indoor noise								
SNR [dB]	**(-∞;-5]**	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
explosion	0	0	0.33	0.88	0.64	0.6	0.25	
broken glass	1	1	0.9	0.67	0.92	0.6		
gunshot	0	0	0.5	1	0.8	1		
scream	1	0	1	1	0.8	0.86	0.57	
other	0.15	0.4	0.17	1	0.64	1	1	1

Localization results

In Figure 7.8 the localization accuracy in relation to SNR is depicted. The plot is prepared in the following way. For all 1600 events the azimuth angle error is computed. Moreover, the SNR is calculated for each event. The events are then sorted in descending SNR order. Thus, the SNR curve (dashed line) and the angle error plot (solid line) are obtained. The average angle error is also shown, which is close to 0. It means that the angle error distribution is symmetrical. It is visible that as the SNR decreases,

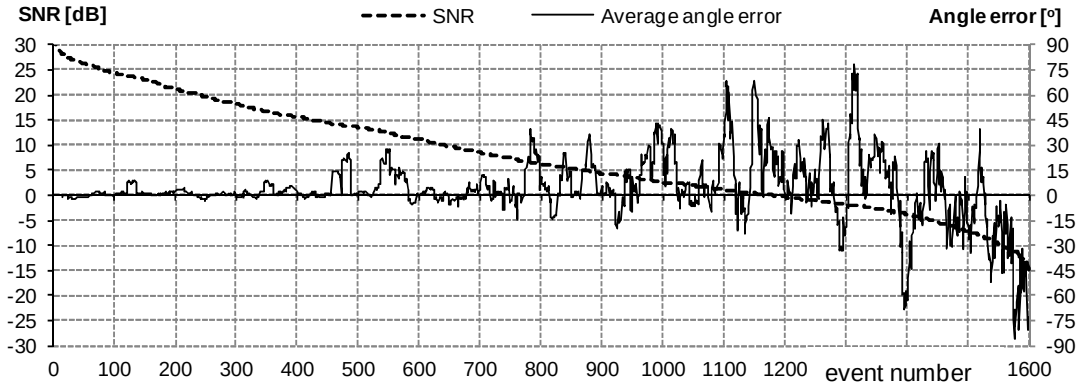


FIGURE 7.8: Relation between azimuth angle error and Signal-to-Noise Ratio

the localization becomes less certain. The error of computed angle is less than 10° for SNRs greater than 10 dB.It also has to be considered that the loudspeakers used for emitting the sound events are not point sources. The angular width of the loudspeakers' diaphragm, as seen from the AVS's reference frame, equals ca. 2-3°. Basing on these results, it can be assessed that the engineered algorithms are suitable for calculating the acoustic direction of arrival in moderate noise conditions.

A more insightful analysis of localization accuracy is provided in Table 7.8. The standard deviations of computed angle are presented with respect to event type and SNR level. Once again it is observed that the uncertainty of the localization substantially increases when the SNR falls below 10 dB. Moreover, it is visible that some events, i.e. those of longer duration (scream and broken glass) are localized better than the impulsive ones. This phenomenon is due to the integration time needed to obtain proper localization information (see Equation 7.2). It is also noticed that traffic and railway noise disturb the localization more than indoor and cocktail-party noise. This fact can be explained by the nature of the spectra of the said parts of noise - the wider the spectrum, the more the localization is impaired. It can also be predicted that it would be easier to localize events in interiors than outdoors. However, it is not always true, since in an indoor space room reflections are an important factor which influences the localization accuracy.

TABLE 7.8: Standard deviations for computed azimuth angle (in degrees) vs. event and noise type

SNR [dB]	**(-5;0]**	**(0;5]**	**(5;10]**	**(10;15]**	**(15;20]**	**(20;25]**	**(25;∞)**
Explosion							
Traffic	84.1	61.3	86.6	8.4	6.1	4	—
Cocktail-party	52	35.7	22.6	9.2	12.6	4.3	0.6
Railway	100.3	69.3	13.1	39.8	7.2	5.2	1.5
Indoor	91.1	52.4	44.1	18.4	5.5	—	—
Broken glass							
Traffic	66.6	44.6	53	9.4	4	6.4	—
Cocktail-party	37.7	37.3	8.6	28.3	5.6	6.5	1.2
Railway	64.1	70.5	18.1	26.8	4.8	3.8	—
Indoor	64.4	52.2	25	11.4	12.6	—	—
Gunshot							
Traffic	56.4	43	17.1	5.8	4.4	3.1	—
Cocktail-party	53.5	75.3	39.2	6.8	3.3	2.7	1.7
Railway	102.7	99	77.9	85.1	14.4	30.2	8.2
Indoor	64.4	34.4	54.8	16.4	22.3	—	—
Scream							
Traffic	48.7	52.7	7.4	3.2	2.7	1.4	1.8
Cocktail-party	16.8	34.6	6.2	3.7	4	2.2	3.5
Railway	72.3	66	8	5.8	4.6	3.8	4
Indoor	11.9	4.2	2.9	3.6	2.8	—	—
Other							
Traffic	68.2	48	7.7	3.8	2.6	3.7	1.9
Cocktail-party	25.1	27.1	3	4.2	2.8	3.4	1.5
Railway	91.1	65.3	6.5	12	4.4	3.2	1.5
Indoor	63.9	15.3	4.1	3.1	3.5	4.6	—

Conclusions

The analysis of the results points out that some conditions of the experiment impair the performance of the methods employed. The most significant aspect is that the acoustic events were played through loudspeakers. The characteristics of the sound which is reproduced by speakers (especially dynamic and spectral features) are different from those of real sounds. This yields a relatively low recall rate for gunshots and explosions. These types of event are practically impossible to be reproduced through speakers with enough fidelity to preserve the dynamics and spectral content of the sound. Therefore the training samples, which were recordings of real events, in some cases do not match the signals analyzed in this experiment in the space of acoustic features. The effect is that gunshots and explosions are either confused with non-threatening events, or confused with each other. The Signal to Noise Ratios of real gunshots and explosions

will also be much greater than the SNRs achieved in this experiment, unless the events are heard from a great distance. In the future research attempts will be made to analyze the efficiency of sound event detection, classification and localization by employing real signals. However, in such cases it is very difficult to measure and control the SNR, which was the key aspect of this work. The values of SNR in this experiment are realistic, i.e. such SNRs are encountered in environmental conditions. It appears that the precision and recall rates achieved in the cross-validation check performed on the training set are very difficult, if at all possible, to achieve in real conditions. The possible reasons for such degraded performance are:

- insufficient noise robustness of features, whose values change significantly when noise is added;
- low noise robustness of the classification algorithm (possibly overfitted to clean signals);
- coincidence of the important spectral components of noise with the components of the events which are substantial for recognizing them (low recall rate of screams in the presence of cocktail-party noise);
- conditions of this experiment, namely reproducing the events through loudspeakers.

Generally, the experiment shows that the developed sound recognition engine achieves adequate accuracy to detect the threatening acoustic events in moderate noise conditions. The detection rates can be considered satisfactory. The achieved precision and recall rates depend on the type of event and disturbing noise. The best performance was obtained for broken glass, gunshot and scream in the presence of traffic noise. Such results are promising concerning the possible practical deployment of the system in urban environment. The further improvement of the performance can be achieved by employing adaptation, which is considered in the following sections.

7.2 Evaluation in realistic conditions

Hitherto mentioned experiments provide the insight into the recognition engine's performance against the training set and noisy data in controlled conditions. For a complete evaluation of the engineered algorithms, the efficiency of recognizing real life events in realistic conditions is also examined. The experiments are focused on the following possible practical usage scenarios:

- **Outdoor urban space**, i.e. detection and classification of threatening events in an open area in the presence of traffic noise;
- **Bank operation hall**, i.e. surveillance of an indoor space, for detection of both typical and threatening events;
- **Public event**, i.e. monitoring of mass events to detect and localize possibly threatening sounds in the audience.

The following subsections are devoted to these usage scenarios. The conditions of the performed experiments are outlined and the achieved recognition results are discussed.

7.2.1 Adaptive detection results

An experiment was conducted to evaluate the adaptation of the detectors and its influence on the efficiency of the sound event detection system. The aim is to show that the proposed methods for adaptation of detection threshold, featured in Section 4.1.3 enable the reduction of false positive detections while maintaining the true positive detection rate, i.e. enables a decrease of equal error rate (EER).

Test signals employed in the experiment contain a 24-hour-long recording of traffic noise performed near a busy street in Gdansk. The signals were recorded using Bruel&Kjaer PULSE recorder in 48000 Hz 24-bit fixed point sample format. The sample format was then changed to 32-bit floating pint. Recordings of gunshots from the training set were added to the test set as example events to evaluate the detection. A gunshot signal was mixed with the background every 60 seconds with random gain varying from 0 to

30 dB. This yields 1440 total acoustic events with roughly random distribution of Signal to Noise Ratio (SNR). Due to this conditions, some events are more difficult to separate from the background and the better the detector's ability to adapt the threshold to the changing conditions, the more events are detected. Also, the efficiency of adaptation influences the false alert rate.

The evaluation is performed employing the *Impulse detector*. Such algorithm is practically useful for detection of impulsive sounds. However, the evaluated adaptation strategies can be successfully used with any other detection parameter, thus enabling detection of different events in different environments.

The evaluation with a DET curve [101] is performed. The sensitivity of the detector is changed from 0 to 1 with 0.05 step. The false alarm probability and miss probability measures are considered:

$$\text{miss probability} = \frac{\text{total number of events} - \text{true positive detections}}{\text{total number of events}} \tag{7.5}$$

$$\text{false alert probability} = \frac{\text{false positive detections}}{\text{false positive detections} + \text{true positive detections}} \tag{7.6}$$

The plot of miss false alarm probability vs. miss probability constitutes a *Detection Error Tradeoff* (DET) curve . Three adaptation strategies are evaluated, as introduced in Section 4.1.3: *single adaptation, double adaptation* and *triple adaptation*. The simple adaptation approach is based on following the average sound level. In the double adaptation method the variance of the sound level is also considered. Triple adaptation is based on both average sound level, its variance, and the rate at which the sound level changes with time.

The DET curves obtained with different adaptation strategies are plotted in Figure 7.9. The detector with triple adaptation performs best for medium sensitivities. Also, the equal error rate (EER) achieved with this adaptation technique is the lowest (0.064). The single and the double adaptation strategies yield similar results. The single adaptation method performs better for low and high sensitivities. However, the EER achieved with double adaptation is slightly lower (0.074 - double, 0.076 - single). The EER is

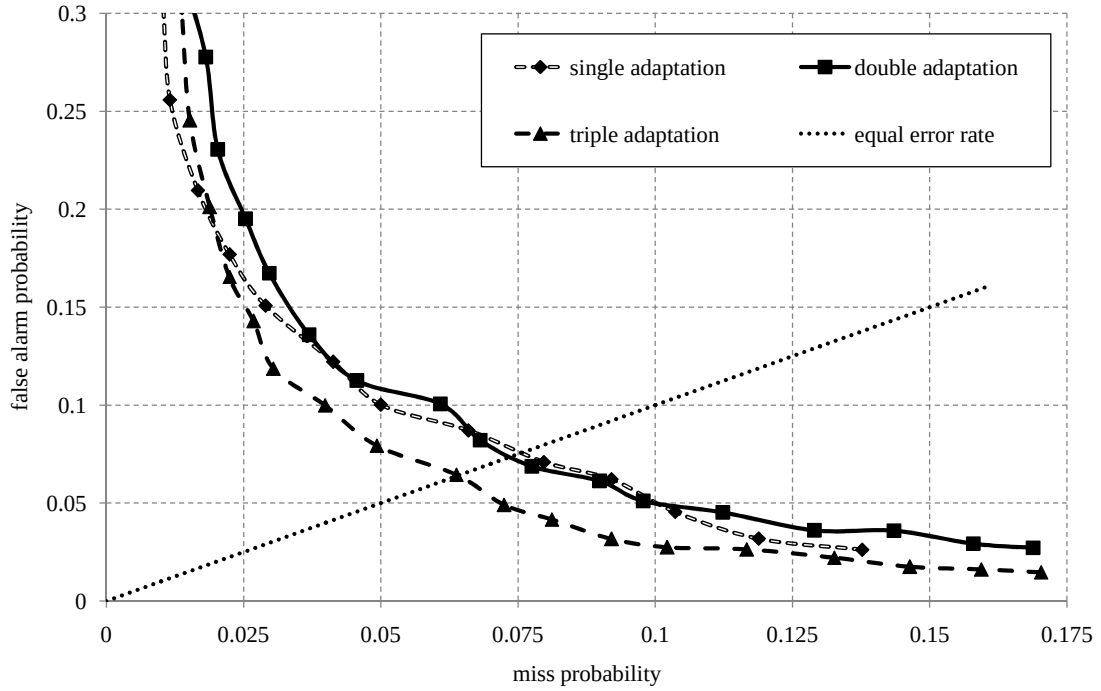


FIGURE 7.9: DET plots for sound detectors with different adaptation strategies

achieved for sensitivity equal to 0.22, 0.3 and 0.45 for single, double and triple adaptation respectively. The achieved reduction in EER justifies the need for adapting the detection threshold to variance and dynamics of the sound level instead of adapting the threshold only to the average sound level. It should again be noted that the minimization of EER is a task related to optimizing the cost of the decision system's operation. In a security surveillance system a missed event, as well as a false alarm cause losses, by either endangering life and property, or unnecessarily engaging security personnel. In such understanding, the system operating in EER conditions, yields minimum loss.

7.2.2 Recognition of events in outdoor conditions

An experiment was conducted to assess the accuracy of recognizing real life events in actual acoustic conditions. Gunshot, broken glass and scream events were recorded near a busy road in the presence of strong traffic noise. Explosion events were not evaluated due to difficulties in reproducing such sounds. For gunshot emission a noise gun was used.

The following signals were recorded: two takes of screams, two takes of broken glass and two takes of gunshots. In case of gunshots one recording contained shots heard from a small distance (2-30 m) and one - from larger distance (60-100 m). The signals were recorded with B&K PULSE system with type 4189 measurement microphones. Five microphones were placed approx. 2, 4, 6, 8 and 32 meters from the source of events. According to the inverse square law, a linear decrease in SNR is expected. The samples from 5 different microphones were treated as separate events in the evaluation methodology. The photographs from the recording process are shown in Figure 7.10.

As far as the accuracy assessment is concerned, a similar methodology to the one elucidated in Section 7.1.3 is employed. The signals are processed with the developed sound recognition engine and the results are compared with the GT description. The metrics for sound recognition accuracy are calculated and aggregated with respect to SNR. The SNR however, is calculated in a different manner, since in this case no reference noise signal and clean event signal is available. Thus, the calculated measure is in fact the relation of *signal plus noise* to *noise*. The process of SNR estimation is illustrated in Figure 7.11. The energy of the fragment containing the event is compared with the energy of a fragment of equal length just before the event.

Again, the estimated SNR values are divided into intervals, only in this experiments the intervals are wider, due to a smaller number of events and lack of precise control over the SNR values. The numbers of events in each SNR interval are shown in Table 7.9. The distribution is not uniform, but the number of events can be considered sufficient to provide the evaluation of recognition accuracy. Please note that the events from the class *other* are not indexed, since they were not emitted intentionally in this experiment. Therefore, the presented tables do not contain the row *other* concerning the results of classifying non-threatening events (as opposed to hitherto presented results). Nevertheless, the false positive classifications are still taken into consideration by analyzing the FP rate for each event class (see Table 7.10).

We begin by evaluating the detection rates. Two detectors are used in this experiment: *Impulse Detector* with sensitivity equal to 0.6 and *Speech Detector* with sensitivity equal to 0.7. The sensitivities were established in a draft experiment as a tradeoff between

FIGURE 7.10: Photographs from recordings of real-world events

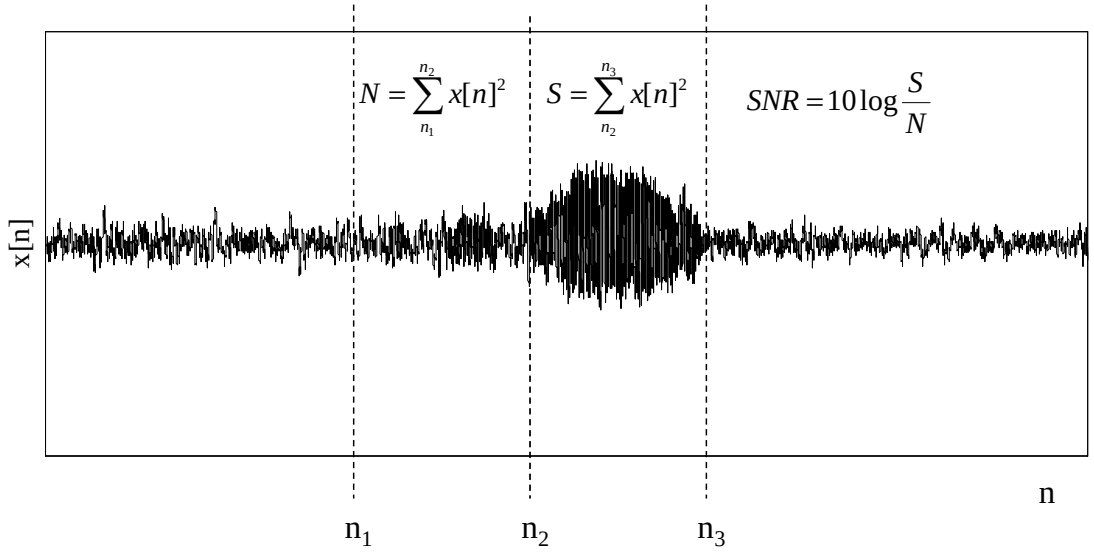


FIGURE 7.11: Method for SNR estimation in practical conditions

TABLE 7.9: Number of events recorded in real conditions in respective SNR intervals

SNR [dB]	**(-∞;0]**	**(0;10]**	**(10;20]**	**(20;∞)**	**sum**
broken glass	17	111	58	19	205
gunshot	78	96	19	172	365
scream	25	166	112	32	335
sum	120	373	189	223	905

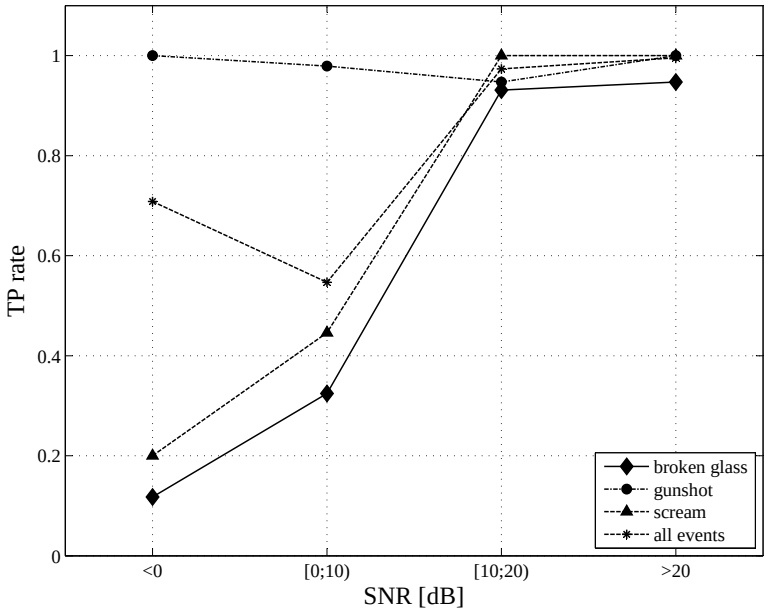


FIGURE 7.12: True positive rates of event detection achieved in practical conditions

false positive and false negative detections. The TP rates of event detection with respect to event type are shown in Figure 7.12. The detection rates are very high for SNRs larger than 10 dB. Unfortunately, the TP rate for screams and broken glass drops significantly when SNR falls below 10 dB. The detection rate of gunshots remains close to 1, which is a surprisingly good result. The spike of energy generated with the noise gun enables correct detection of this type of sound even if the average energy of the event is lower than the energy of noise. However, it should be noted that if the energy of the gunshot was calculated in shorter time base (i.e. from the spike only), the SNR estimate would be much higher.

In Table 7.10 the true positive and false positive detection rates for respective events are aggregated. The results for all SNR levels combined are presented. Again, gunshots are most reliably detected. The broken glass event yields the lowest TP and the highest FP rate. The high false positive rate means that a lot non-threatening events, not present in the GT list, were classified as broken glass. The classification algorithm should be improved to reduce such false alerts. Screams yield an acceptably low FP rate. The overall TP rate of screams can also be considered satisfactory, taking into consideration the fact that many of the scream events were strongly masked by noise.

TABLE 7.10: True positive and false positive detection rates achieved in practical conditions

	TP	FP
broken glass	0.537	0.225
gunshot	0.992	0.095
scream	0.666	0.063
average	695/905 [76.8%]	85/780 [10.9%]

In the next analysis, the classification results are considered. The precision and recall rates for broken glass, gunshot and scream are plotted in Figure 7.13. The lowest SNR range (SNR<0) was only considered for gunshot event since for scream and broken glass too few events were detected at this level. The recall rates achieved can be considered satisfactory under such difficult conditions. The lowest recall rate (equal to 0.622) is obtained for screams at SNR below 10 dB. The precision rate for broken glass at highest SNR is surprisingly low. To account for this fact, the confusion matrix needs to be examined - see Table 7.11. It is visible that 17 gunshots were classified as broken glass. This error highly influences the precision factor for broken glass, despite the fact that it only concerns ca. 10% of the analyzed gunshot events. The large number of gunshots in this SNR interval is obviously the consequence of high energy of gunshots in general. If the events were more evenly distributed, the precision rate for broken glass would not be lowered that much.

TABLE 7.11: Confusion matrix for SNR > 20 dB in practical conditions

	glass	shot	scream	other	recall	precision	F1-score
glass	**17**	0	1	0	0.944	0.500	0.654
shot	17	**154**	0	1	0.895	1.000	0.945
scream	0	0	**32**	0	1.000	0.970	0.985
					accuracy: 203/222 [91.44%]		
					average F1-score: 0.861		
					κ: 0.666		

In another confusion matrix, shown in Table 7.12, the classification results obtained for SNR from the interval [0;10) dB are shown. The achieved precision and recall rates, as well as the accuracy and F1-score yield high values. The comparison with the results achieved in simulated conditions (Table 7.5) shows that the recognition engine performs better than in the controlled environment. This is due to the fact that the reproduction

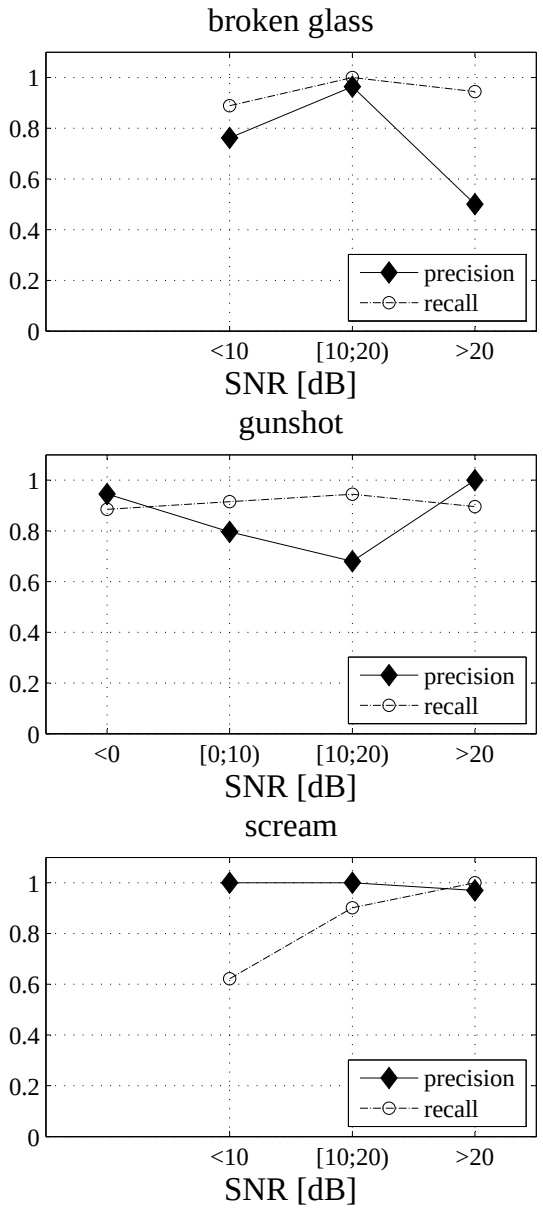


FIGURE 7.13: Precision and recall rates achieved in practical conditions

of signals by loudspeakers distorts the qualities of sound, which was already mentioned in Section 7.1.4. Finally, in Table 7.13 the confusion matrix for all events recorded in this experiment is presented. Please note that only the events which were correctly detected were taken into consideration (695 TP detections). To properly express how many of the detected events were properly recognized one has to take into consideration the TP and FP rates. The corrected values of precision and recall, taking into account the detection rates, are shown in Table 7.14. It can be seen how the low FP rate impairs the precision rate and how insufficient TP rate lowers the recall score.

Table 7.12: Confusion matrix for SNR from the interval [0;10) dB in practical conditions

	glass	shot	scream	other	recall	precision	F1-score
glass	**32**	1	0	3	0.889	0.762	0.821
shot	8	**86**	0	0	0.915	0.796	0.851
scream	2	21	**46**	5	0.622	1.000	0.767
					accuracy: 164/204		[**80.39%**]
					average F1-score: 0.813		
					κ: 0.693		

Table 7.13: Overall confusion matrix obtained in practical conditions

	glass	shot	scream	other	recall	precision	F1-score
glass	**105**	1	1	3	0.955	0.739	0.833
shot	33	**326**	0	3	0.901	0.906	0.903
scream	4	33	**179**	7	0.803	0.994	0.888
					accuracy: 610/695		[**87.77%**]
					average F1-score: 0.875		
					κ: 0.801		

Table 7.14: Recall and precision factors corrected with TP and FP rate

	TP	FP	recall	precision	corrected recall	corrected precision
broken glass	0.537	0.225	0.955	0.739	0.512	0.603
gunshot	0.992	0.095	0.901	0.906	0.893	0.819
scream	0.666	0.063	0.803	0.994	0.534	0.918

The conducted experiment leads to some important conclusions. Firstly, it is visible that the recognition rates achieved in practical conditions, using real-life events, are slightly better than the ones achieved in controlled environment during the simulation featured in Section 7.1. A brief comparison is shown in Table 7.15, in which the results achieved in real conditions at SNR from 0 to 10 dB are compared with the results achieved in the simulated environment at an SNR from the interval [5;10) dB. This finding can be explained by the fact that in the simulated environment the events were reproduced by loudspeakers. The introduction of a loudspeaker to such signal chain apparently inflicts significant distortions. Thus, the signal features are tampered and the recognition results are impaired. It is also confirmed empirically that an event, especially a gunshot, sounds differently when emitted from a loudspeaker than a real-life one. This leads to a conclusion that the recognition of acoustic events is a fragile process, in which even small changes in the signal can influence the classification accuracy and that to achieve

robustness, close attention should be paid to the acoustic conditions of the recognition process.

Secondly, frequent misclassifications are observed between the known classes of sound. It can be noticed by examining the results in Table 7.13, that during the experiment as much as 33 gunshot events were classified as broken glass and exactly the same number of screams were classified as gunshots. To investigate this kind of errors, the outputs of the classifier are presented. In Figure 7.14 the result of an incorrect classification in which a gunshot event is classified as broken glass is presented. It is visible that generally the analyzed sample frames are recognized as similar to gunshot, but they fail to exceed the threshold, which according to the experiment featured in Section 6.3.2 equals 0.75 for gunshots. On the other hand, the classifier output for broken glass exceeds the probability threshold for broken glass events (0.45) and thus leads to a false alarm.

An example of correctly recognized gunshot is shown in Figure 7.15. The output gunshot probability exceeds the threshold for gunshots and the event is correctly classified. It is visible that the initial phase of the event is crucial for correct recognition. In future work more attention should be devoted to analyzing the attack phase of the sound.

The author of this dissertation believes, that such errors could be reduced by retraining the model after including this incorrect classifications in the training set and repeating the experiment. Moreover, an effort could be made to identify the signal features which are suspected of contributing to such errors ad eliminate them from the feature vector. Such investigation is one of the topics for future research on the subject.

Despite the observed errors, the achieved performance can be regarded as good. Even in strong noise conditions, the considered threatening events are recognized with ca. 80% certainty. It shows that the engineered methods have the potential to be employed in a practical security surveillance system.

7.2.3 Recognition results in a bank operation hall

Several experiments were performed with the participation of the author of the dissertation in a bank operation hall. It served as a testbed for developing the technology of

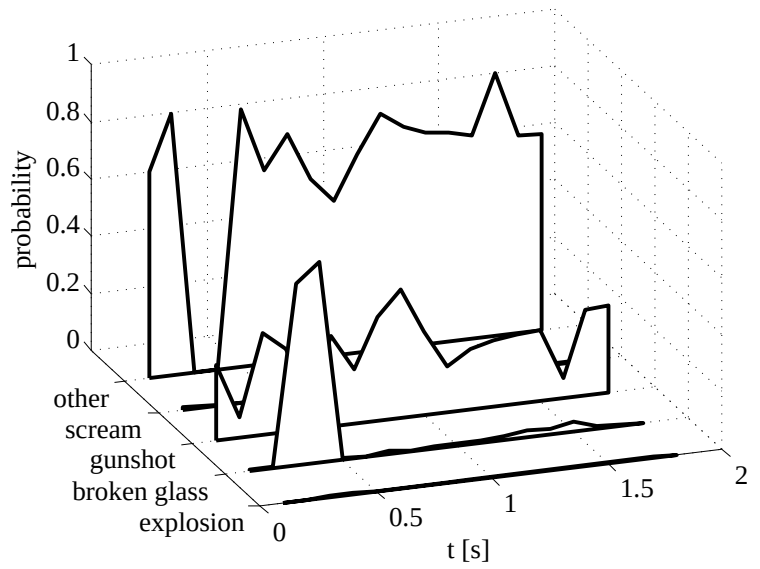


FIGURE 7.14: Example classifier output for a gunshot classified as broken glass

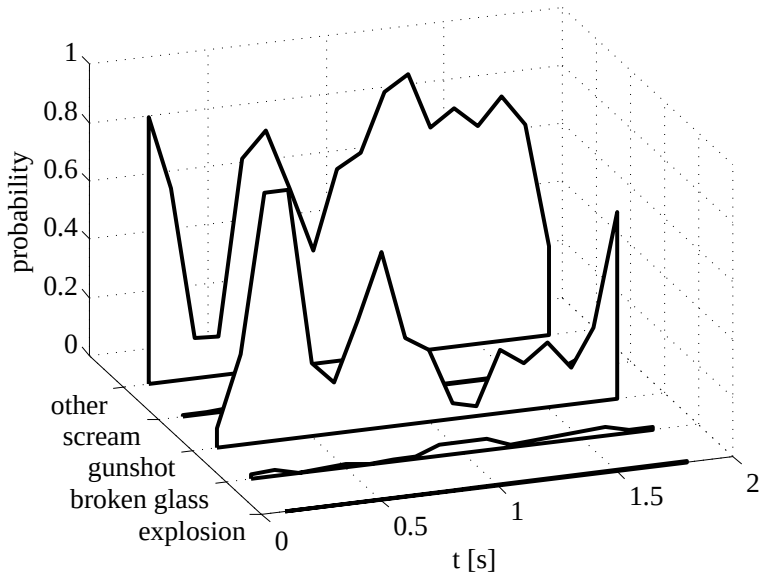


FIGURE 7.15: Example classifier output for a gunshot correctly classified as gunshot

TABLE 7.15: Comparison of classification results achieved in simulated conditions and in real acoustic environment

event	recall	precision
broken glass (real)	**0.889**	**0.762**
broken glass (simulation)	0.885	0.568
gunshot (real)	**0.915**	**0.796**
gunshot (simulation)	0.815	0.8
scream (real)	**0.622**	**1**
scream (simulation)	0.571	0.774

acoustic surveillance of an indoor space. In case of a bank space, as well as many other types of indoor spaces, the detection of both threatening and typical events is important. As far as the bank is concerned, the information about typical events gives an insight into the operational qualities of the institution. Such information can be processed from the management point of view. In the bank operation hall, the following events are considered typical:

- speech, often forming a so-called cocktail-party noise;
- stamping;
- other sounds, such as chairs moving with a squeak, safes beeping, people stepping, objects being put down on desks etc.

As far as the alarming events are considered, screams or raised voice should definitely raise attention. Gunshots are also considered, since they are sometimes present during armed robberies. Therefore, in the task of surveillance of the bank operation hall, we define the following events to be detected: *speech, scream, gunshot, stamp, chair, beep, other*. Since the set of events is different from what was assumed before, obviously a different set of training signals was employed in this experiment. Events recorded in the bank operation hall were utilized rather than those described in Section 6.1.

Detection

Two detectors are used to detect sound events in the bank operation hall: *Impulse detector* and *Speech detector*. The impulse detector uses a 512-sample frame, whereas the speech detector uses a 4096-sample frame. The sampling rate is equal to 48000 samples per second. The sensitivity of the impulse detector equals 0.2 whereas the sensitivity of the speech detector equals 0.3. Double adaptation is employed. The adaptation time of both detectors equals 10 minutes.

The changes of the detectors thresholds are presented in Figure 7.16. The results originate from the 13-hour online operation inside the bank operation hall. The bank is open for customers from 9:00 to 18:00. Around 8:00 the bank employees start preparing for

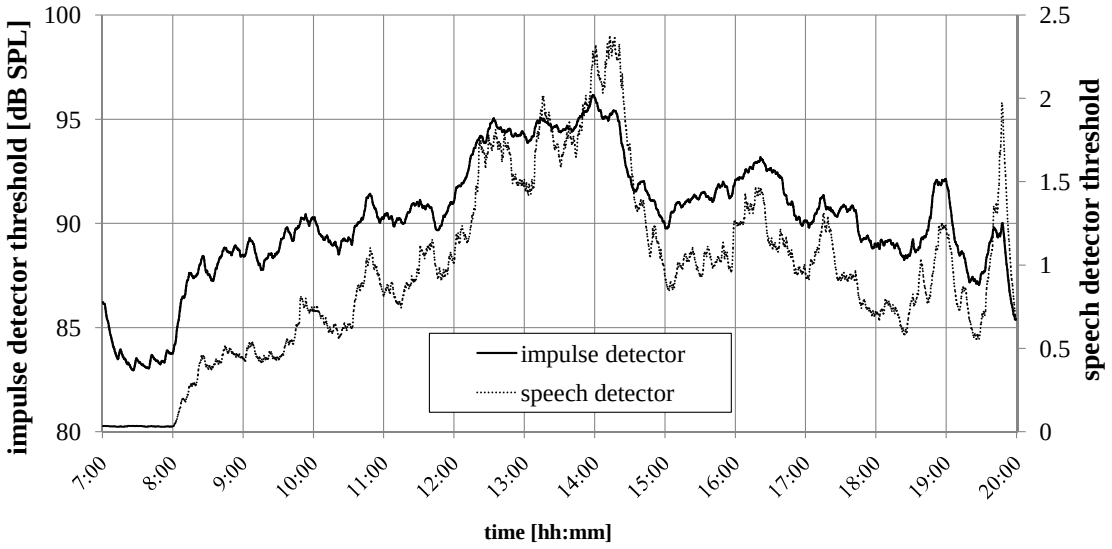


FIGURE 7.16: Adaptation of sound detectors in bank operation hall

work, thus generating some noise. It is visible in the threshold of the speech detector. It can also be observed that the detection threshold is elevated during peak hours (ca. 13:00 to 14:30). The impulse detector threshold is ca. 10 dB higher during peak hours than early in the morning. It shows that thanks to the adaptation technique employed the detector is capable of detecting quiet events at night and is not prone to false alerts when the level of the acoustic background is high. It is a very useful feature for the surveillance of the indoor space. Provided a person broke into the bank at night, even a very subtle noise would trigger detection. The same principle applies to the speech detector. During the day, when lots of people speak and generate cocktail-party noise, a regular voice would be less likely to trigger an alarm that outside opening hours.

Classification

During the same experiment as described in the above paragraph, the classifier was also operating. The events were detected and recognized during and outside bank operating hours. The numbers of detected events of each type are shown in Table 7.16. In total 1026 acoustic events were detected. It is visible that stamps (547 detections) are the prevailing sounds. Other events are also frequently present (368 detections). Only 84 speech events were recognized, which can be explained by the fact that speech is often present in the background as cocktail-party noise and the speech detector adapts to this

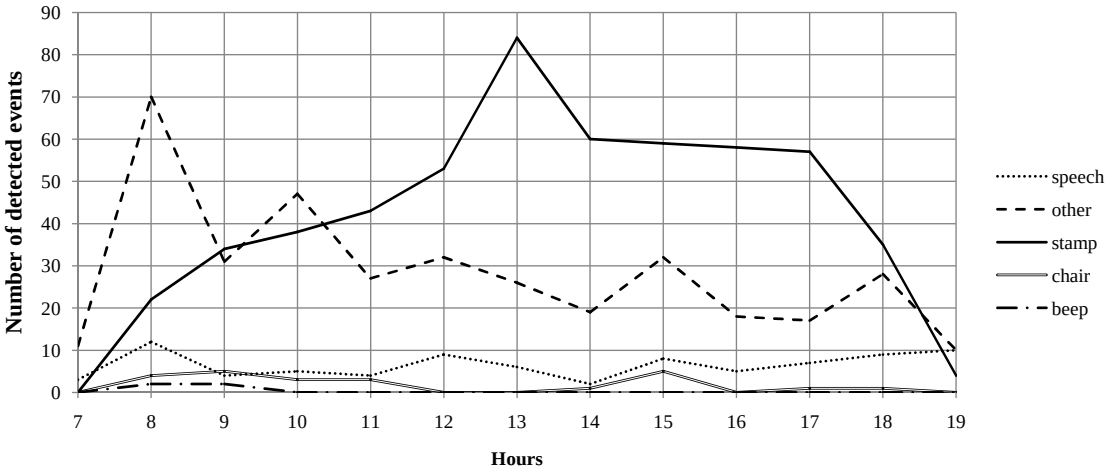


FIGURE 7.17: Histogram of events detected during bank operating hours

background by adjusting the detection threshold. The sound of moving chairs and money safe beeps were seldom detected (23 and 4 occurrences, respectively). No threatening sound events were detected during that period, which means that the system did not produce a false alarm.

TABLE 7.16: Number of events of each type detected in bank hall during operation

Type of event	speech	other	stamp	chair	beep
number of detected events	84	368	547	23	4
				total:	**1026**

In Figure 7.17 the distribution of events per hour of operation is presented. It is visible that during the early hours – 7:00 to 9:00. – other sounds are more frequent than stamping. Stamping becomes more frequent during the operating hours, which are 9:00 to 6 18:00. Information about the movement in the bank can also be derived from the event distribution. The peak hour appears to be on 13:00. Speech events are evenly distributed, since they are caused by both: the clients and the staff of the bank, who are present during the whole period.

To evaluate the classification of alarming events, a situation was arranged in which actors were asked to generate screams, talk in raised voice and fire guns. The goal was to recreate an armed robbery event. during the 30 minutes of the staging, a number of 74 alarming events were detected. Sounds belonging to the following classes were present in the test signal: *speech, scream, gunshot* and *other*. The analysis of the classification

results is presented in Table 7.17. It can be observed that the threatening events are recognized with an acceptable accuracy. However, the speech sounds are often confused with other acoustic events. All gunshots were correctly classified, although only four shots were emitted.

TABLE 7.17: Confusion matrix for recognizing threatening events in bank operation hall

	speech	scream	gunshot	other	recall	precision	F1-score
speech	**24**	7	0	10	0.585	1.000	0.738
scream	0	**21**	0	4	0.840	0.750	0.792
gunshot	0	0	**4**	0	1.000	1.000	1.000
other	0	0	0	**4**	1.000	0.222	0.364
						accuracy: 53/74 [71.62%]	
						average F1-score: 0.723	
						κ: 0.58	

7.2.4 Detection and localization of events in a public event space

The final use case considered is public event surveillance. During mass events such as concerts or sports games a large number of people is crowded in a limited space. If a threat occurs in such circumstances, the consequences may be fatal. Hence, the detection and warning about possible threats is of utmost importance. The project *MAYDAY Euro 2012* addressed such situations, e.g. by developing algorithms for detection of abnormal behavior of crowd [168, 175]. The author of the dissertation was involved in the work on the methods for detection and localization of acoustic events in the public event audience [6, 7]. In this section the results of this work are briefly outlined. The goal is to detect alarming events such as screams or gunshots and to localize its sources. The information about the localization of the event source in the audience is considered, i.e. row and seat number. Such data is particularly useful in the context of mass events, when row and seat number can be connected with the name of the person occupying the seat. In an example usage scenarios of such system, a gun is fired in a stadium. The localization instantly pinpoints the seat in which the shooter is present and the PTZ camera zooms in to the shooter. Such a functionality is realized by the developed KASKADA service called *audytorium_complex* (see Section 5.3).

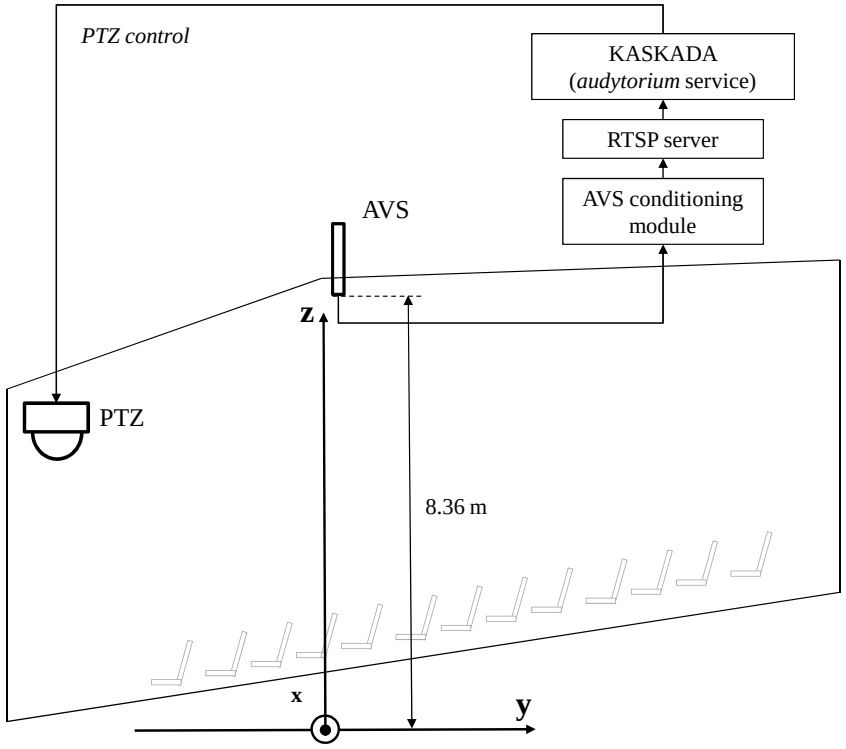


FIGURE 7.18: Vertical section of the hall and the setup of the system for detection and localization of events in the audience

Setup and equipment

The testbed for developing the algorithms for detection and localization of events in public event audience was a lecture hall in Gdańsk University of Technology. The vertical section of the hall, the hardware setup and the employed coordinate system are presented in Figure 7.18. The acoustic vector sensor (AVS) is mounted under the ceiling, 8.36 meters above the lowest floor of the auditorium. The AVS conditioning module, comprising filters and preamplifiers for the signals from the AVS, receives the signals from the AVS, being pressure signal p and three particle velocity signals u_x, u_y, u_z. The signals are encoded and sent to the KASKADA platform via a RTSP server mentioned in Section 5.2. On the KASKADA platform the *audytorium* service is executed which comprises signal processing algorithms capable of computing the localization of the detected event in the audience. The information about the location of the source is then translated to *pan,tilt* coordinates of the PTZ camera mounted in the hall. Thus, the camera is automatically pointed in the direction of the detected event.

The detailed layout of the audience is shown in Figure 7.19. The center of the coordinate

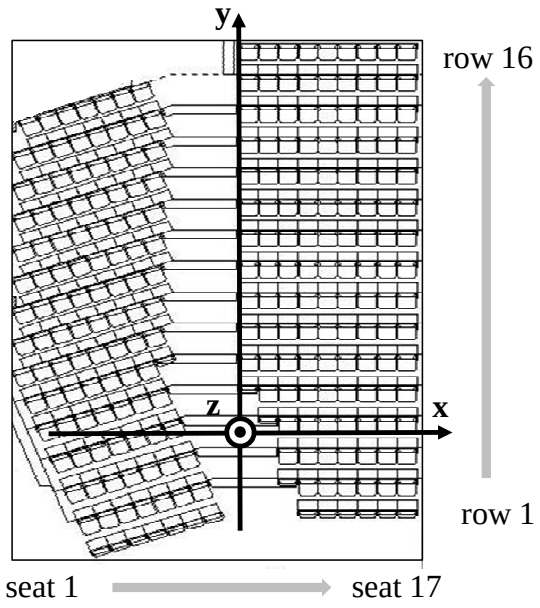


FIGURE 7.19: Top view of the audience with the employed coordinate system and row/seat numbering

system ($x = 0; y = 0; z = 0$) is located 8.36 meters below the AVS, in the aisle at the height of row 4.

Methods

As it was expressed in Equation 7.2, the signals from the AVS are utilized to compute the coordinates of the sound intensity vector **I**. Knowing the dimensions of the room, the exact seat from which the sound originates can be determined. The process of this geometrical calculation is illustrated in Figure 7.20. The AVS location is marked as red circle, whereas the sound source location is indicated by a blue filled circle. The height H on which the AVS is mounted and the coordinates on the floor plane p are known. The point of intersection of the direction of the sound intensity vector and the floor plane is found. Next, the coordinates (x, y) of this point are converted to seat number with the use of a look-up table prepared during the system calibration. The row and seat number are then converted into the coordinates of the PTZ camera.

To evaluate the accuracy of sound source detection and localization in the indoor space the following experiment was carried out. Noise gun was used to emit shots. Five shots were emitted from each seat in the auditorium. The signals were processed online and

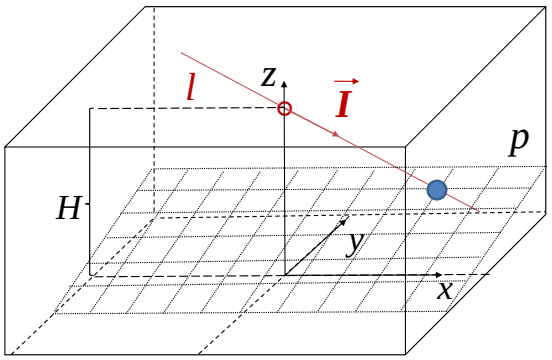


FIGURE 7.20: Determination of sound source location in the audience

for each signal the azimuth ϕ and elevation θ angle of the acoustic Direction of Arrival (DoA) were calculated.

The calculated coordinates of the sound source are compared with the ground truth values known from the mentioned look-up table used for converting coordinates to seat number. The absolute error is considered $\Delta x = |x_{comp} - x_{GT}|$ where x_{comp} denotes the computed value and x_{GT} denotes the reference value. The minimum of the five shots fired from each seat is considered, denoted Δx_{min} and Δy_{min}.

Results

The results are presented on 2D plots which show how the localization error depends on the location of the source in the audience. During the experiment it turned out that for some seats the localization is more accurate than for others. It led to a conclusion, that the observed errors depend on the qualities of the room. The possible ways in which the conditions in the hall deteriorate the sound source localization procedure are as follows:

- In the assumed model the surface of the seats is a regularly sloped plane. In fact the left side of the auditorium is sloped at a different angle than the right side.
- The air in the room is heated with heaters placed near the left wall. In such a room the air temperature is not constant in relation to height. Therefore, sound waves refract and do not propagate along straight lines, which leads to change in the direction of coming sound.

- The probe is located on the level of the sound directing panels hung under the ceiling. It is possible that the panels reflect the direct sound from some localization in such a way, that the reflections from the walls reach the probe first. Thus, the angle of coming sound is calculated with error. There can also be reflections from the ceiling present above the panels (possible cause of the elevation error in row 5).

- The error of calculating the y coordinate is probably caused by the inaccurate assumption of the seat surface plane. The characteristic points in the hall, used to find the formula of the plane p, were measured with some uncertainty.

- The error in localization can also be caused by the possible tilt of the AVS.

To compensate for the mentioned effects, a correction function was employed which incorporates the general trend observed in the localization errors [6]. The error of the calculated x coordinate of the sound source Δx_{min} is shown in Figure 7.21. The results with and without correction are presented. Initially, large errors are observed in the left side of the auditorium. It is visible that the employed correction procedure leads to an improvement of calculation accuracy of the position of the sound source in the lecture hall. Such a calibration should be performed after the system is installed in a room. The errors related to x, y coordinates which are smaller than 1 m can be interpreted as a good accuracy, since it yields resolution of 1-2 seats in the audience, which is usually satisfactory for the application of monitoring of public events.

In Figure 7.22 the accuracy of determining the y coordinate of the sound source is examined. Similarly to the case presented in Figure 7.21, before correction the localization is more accurate on the right side. Also, large errors are observed in the back of the hall. The employment of the correction function levels the accuracy. In most parts of the auditorium the error is less than 0.5 m (approx. 1 seat). The errors in the last rows still remain large after correction, though.

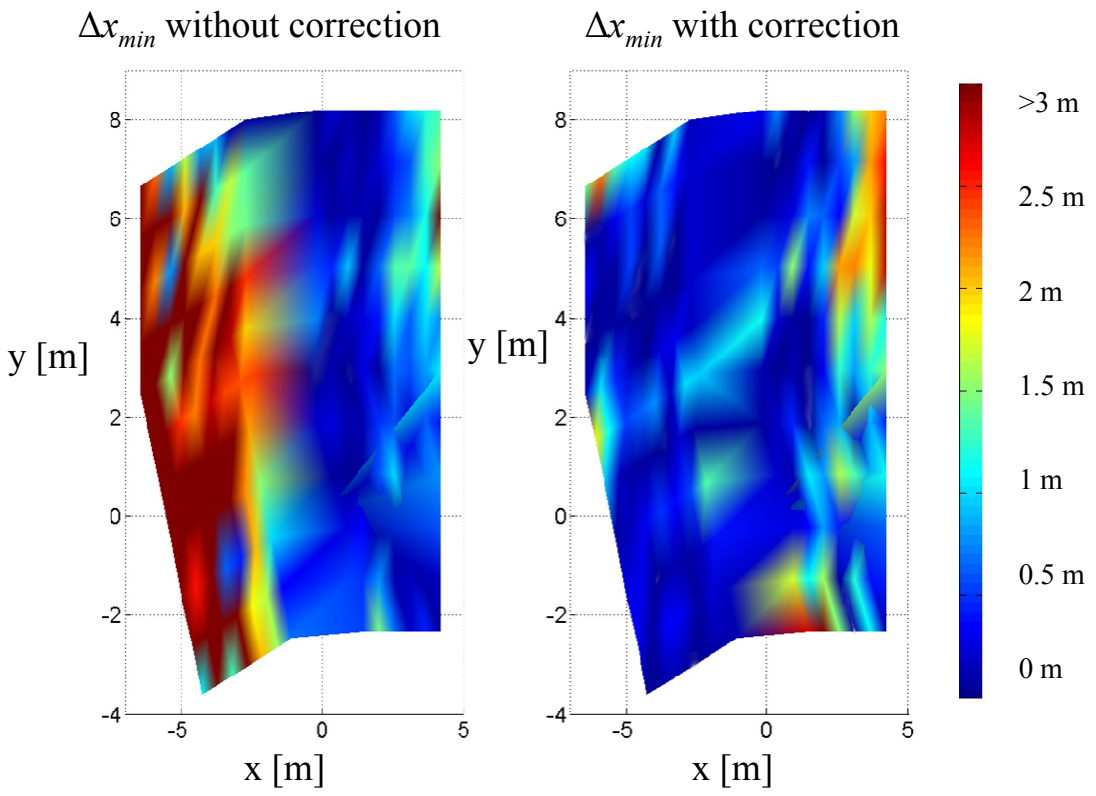


FIGURE 7.21: Error of determining the x coordinate of the sound source in the audience of a public event [6]

Example use case

A draft experiment is introduced to show how the developed technology, after required adjustments, can be used to monitor the crowds during sports events. Let us assume that a microphone (or an acoustic vector sensor) is installed in a stadium. The possible threatening events are e.g. gunshot or crowd panic. In case of a gunshot it has already been shown that the location of the shooter can be determined with adequate precision. It is also shown in Section 7.2.2 that the gunshot can be detected in most cases even in challenging acoustic conditions. The problem is however, the number of false positive detections. To examine how the developed algorithms react to the sound recorded during a football match, an example match recording was used, from an Internet database [176]. It is identified that the following events are typical during a football match:

- crowd cheering (reaction to goal, foul etc.),
- horn blowing,

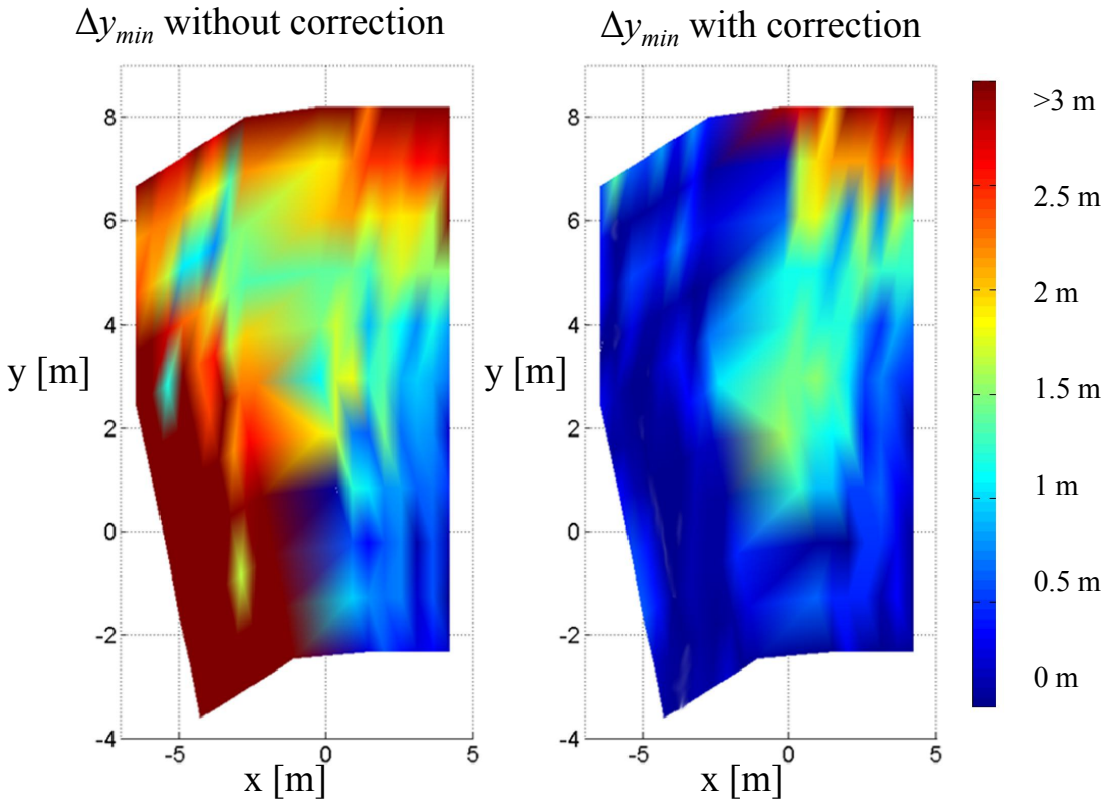


FIGURE 7.22: Error of determining the y coordinate of the sound source in the audience of a public event [6]

- clapping,
- drums beating.

The first analysis of the recording resulted in a vast number of false positives. Therefore, the events detected during the first half of the match, were used as examples of the *other* class and the classifier was trained again taking into account these new events. Subsequently, the analysis of the recording from the second half is performed. The following types of false positives were observed:

- regular voice of person in crowd classified as scream,
- drum classified as explosion,
- clapping recognized as gunshot, and once as broken glass.

In general, the false positives are not frequent. During the 45 minutes of the match, a number of 3 false explosions, 1 false broken glass event, 2 false gunshot and 5 false

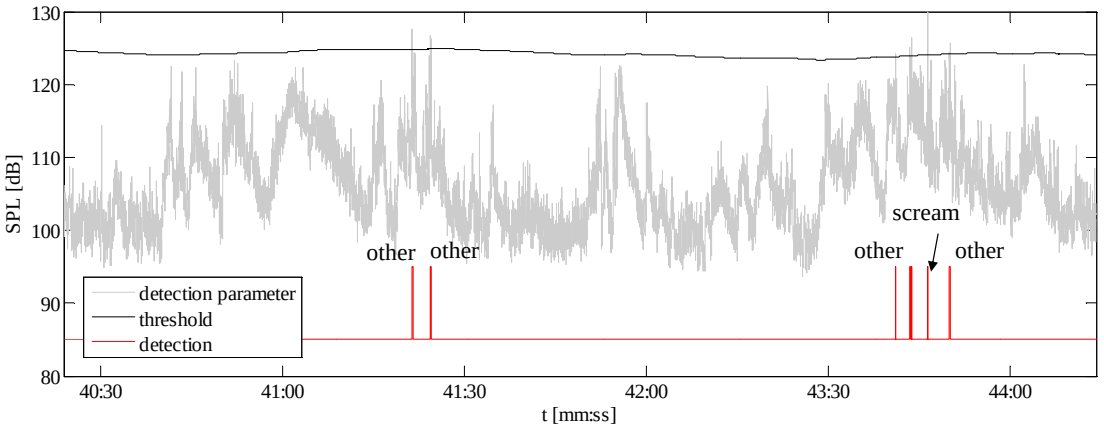


FIGURE 7.23: Example detection and recognition results from a recording of a football match

screams were detected. The count of false positives could be lowered by lowering the sensitivity of the detectors, which was not evaluated in depth in this draft experiment.

The example detection and recognition results of a fragment of match recording are shown in Figure 7.23. The detection parameter of *Impulse detector* and its threshold are shown. Some peaks and troughs are visible in the sound pressure level course, due to changes in the acoustic environment. Most of the peaks do not exceed the threshold, which is ca. 20 dB higher than the background level. However, around 43:40 the crowd gets louder due to a situation at the end of the match. Some events exceed the threshold. One false detection of scream is present when a person close to the microphone speaks loudly. The example shows that if the detector's sensitivity is low, the detection should not be triggered by normal behavior of crowd. The impulse detector reacts mostly to incidental bursts of acoustic energy.

7.3 Conclusions from practical experiments

In the chapter several experiments were featured whose aim was to assess the ability of the engineered methods to detect, classify and localize threatening acoustic events in practical conditions. The system can be considered flexible, since it can be adopted to operate in varying conditions, both outdoors and indoors. The detection, recognition and localization was examined employing the known methods and metrics. Satisfying

results were obtained in most conditions. However, some improvements are still needed to improve the robustness against noise and changing environment. The most important conclusions derived from the practical experiments are listed below:

1. High recognition results achieved during cross-validation on the training set are not attained during practical experiments. It is obviously caused by the fact that the acoustic environment influences the signal strongly. To achieve a more robust system, the samples in the training set should be more diversified, i.e. recorded in different environments, at different SNRs etc.

2. Detection is a crucial operation in the recognition process. If an event is missed at this stage, there is no correction possible. It is a consequence of following the *detection-and-classification* approach. The robustness of the detectors could still be improved, thus improving the performance of the recognition engine as a whole.

3. The advantage of the assumed detection approach is the ability to employ *adaptation*. Adaptation is a significant advantage of the engineered methods. In the featured practical experiments it was shown that the same recognition engine can be used in such conditions as: outdoor space, bank operating room and auditorium hall. In all the mentioned environments the events are correctly detected thanks to adaptation.

4. As far as classification errors are concerned, it is more often observed that a non-threatening event is classified as hazardous than vice versa. It is a good feature regarding possible application in security surveillance. A false alert can always be verified by human personnel, whereas a false negative value leads to an omission of an alarming situation. It is also shown that the number of false alerts can be reduced by employing adaptation of sound event detectors.

5. Localization is an improvement of the sound event recognition system. The information about the direction of coming sound is highly useful in security surveillance. It was shown that the employed methods, basing on the signal from the acoustic vector sensor, are capable of localizing the sound source with adequate accuracy.

Chapter 8

Parallel processing experiments

In this chapter the experiments related to parallel processing of audio data on the supercomputing cluster are introduced. Two experiments are featured. The first one concerns the speedup of the analysis of large amount of audio data in offline mode. In the second experiment the evaluation of decision making time in online mode is presented. It is shown that employing parallel processing leads to shortening of the time needed to recognize the event.

8.1 Speedup of offline analysis

In a possible usage scenario of the developed system, some security services need to search through a recording from acoustic surveillance to find a fragment of interest, e.g. pertaining to a hazardous event which happened in the past. Manual search of the long audio material, possibly containing hours or even days of registered data, would be extremely time-consuming. Automated hazardous sound event recognition can be used to indicate the times in which atypical events occurred, thus facilitating the search process. It is obvious that the faster such automatic search is, the better. Hitherto presented methods can be used for processing registered stream from acoustic monitoring [15]. In fact, a specific KASKADA service, called *SoundRecFile*, was created for such purpose (see Section 5.3). It utilizes data-level parallelism to speed up the offline analysis of large amount of registered audio data.

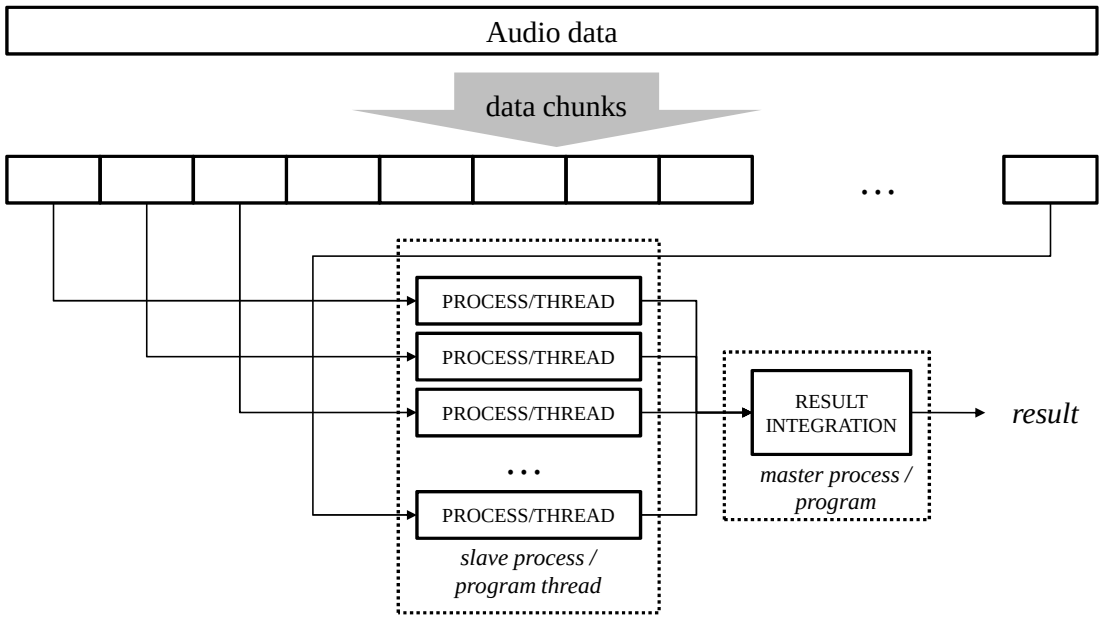


FIGURE 8.1: Approach to offline analysis of registered audio data

The approach to the task offline analysis of registered audio data is presented in Figure 8.1. The large portion of acoustical data (presumably in many sound files like *wave* or *mp3* format) is divided into equal chunks. Each chunk processed by a separate thread or process in parallel. After the processing tasks are completed, the results are sent back to the master process. The output of each chunk is the data concerning the detected acoustic events. The synchronization of results is performed and the final output is the list of detected events in the whole audio material.

8.1.1 Parallel processing approaches

Four different approaches to parallel offline analysis are considered. The differences lie in the assumptions regarding the employment of the supercomputer's resources and the KASKADA framework. The different strategies are evaluated later in the section. The illustration of the employed approaches is provided in Figure 8.2.

A. **Execution directly on supercomputing node** - The program is executed manually after logging on to the chosen supercomputer node. The program uses *boost thread* library to execute operations in parallel. Only the resources available on the

current node are available. At the date of the experiment evaluating this approach, 8 cores were available per supercomputing node.

B. **Execution in KASKADA framework with threads** - The program is executed in the KASKADA framework, using the KASKADA user console. However, it still uses *boost* threads and utilizes only the resources of a single node.

C. **Execution in KASKADA framework with master-slave processes** - The KASKADA framework provides master-slave mechanism for offline processing tasks. The master process is executed on a chosen node and then runs slave processes, which handle the actual data processing. The slave processes may be allocated on whichever node is available. The KASKADA framework handles the allocation of the resources (see Section 5.1.2 for reference).

D. **Execution in KASKADA framework with master-slave processes and threads** - In this approach the master-slave mechanism is utilized, but the slave processes are also divided to *boost* threads. The KASKADA allocates each slave process on a node with free resources. The threads of the slave process occupy the cores available on the node.

In all approaches the slave processes or threads are given the information about the time range from the registered data which they should operate on. All processes and threads access the same audio files on disk.

8.1.2 Experiment for evaluation of the processing time

First, a draft experiment is performed to compare the strategies employed for parallel computations. The experiment is conducted on *Galera* cluster (with 8 CPU cores per node, see Table 5.1). In the draft experiment only up to 10 threads or processes are employed. A 1-hour long recording is used as test signal. The results are shown in Figure 8.3. Three measurements of computation time are performed for each strategy. The median value is taken into consideration. The speedup of computations is determined in relation to the time of analysis on 1 CPU, according to the definition in Equation 3.1.

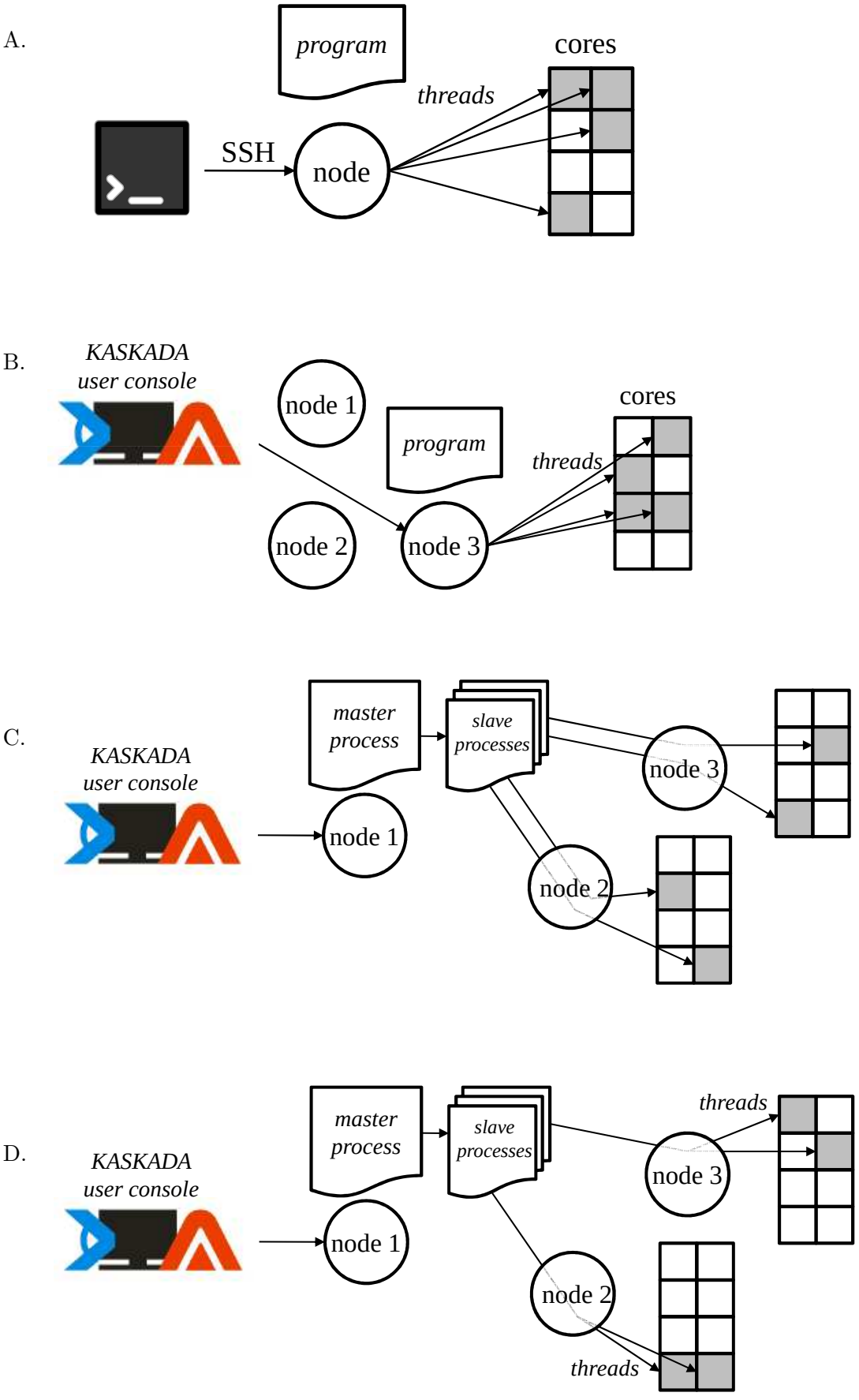


FIGURE 8.2: Approaches employed to parallel offline analysis

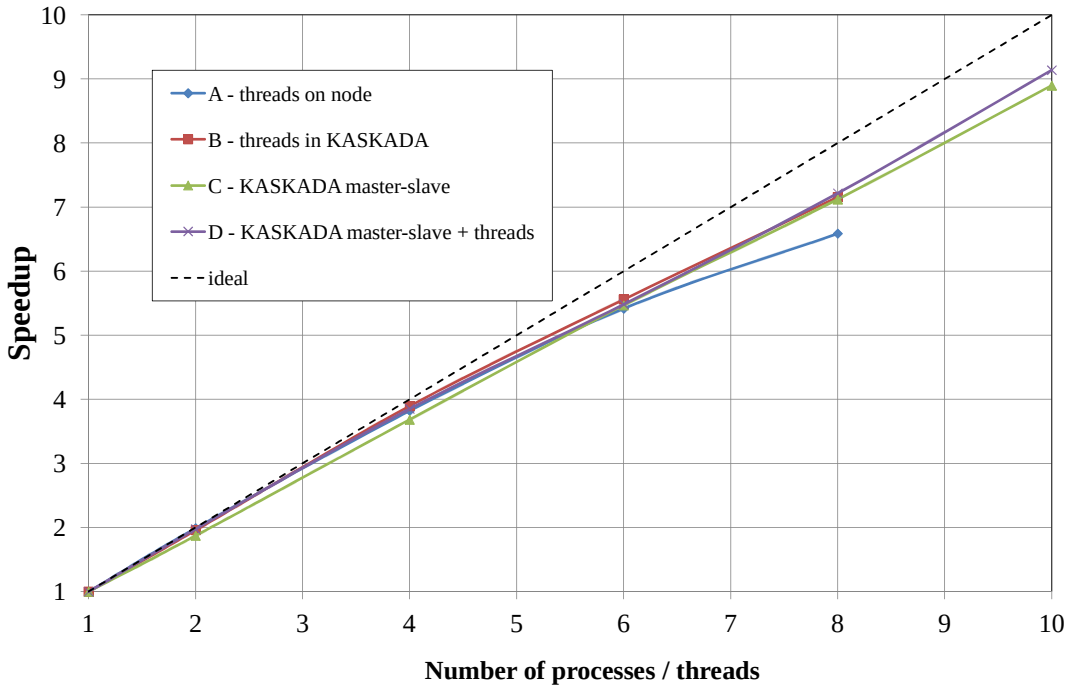


FIGURE 8.3: Draft comparison of offline processing strategies

More sophisticated methods for scalability evaluation are also presented in Section 3.1. However, in this case of offline data processing the speedup metrics is believed to be sufficient.

It is visible in Figure 8.3 that the results obtained with different parallel processing strategies are similar. However, it can be noted that the execution of the program outside the KASKADA framework (approach A) yields worse results. The remaining strategies perform similarly up to 8 threads. The B approach (threads in KASKADA framework) cannot be efficiently used with more than 8 threads, since only 8 CPU cores per node are available. The strategy D (master-slave processes divided into threads) appears better than C (master-slave processes), however the difference is not large. Therefore, in more precise evaluation, the approaches C and D are compared.

The next experiment is carried out to assess the scalability of the solution. Since a greater number of nodes is utilized, only strategies C (KASKADA master-slave) and D (KASKADA master-slave + threads) are evaluated. The *Galera Plus* cluster is employed. A 24-hour long recording from a noise monitoring station, the one shown in

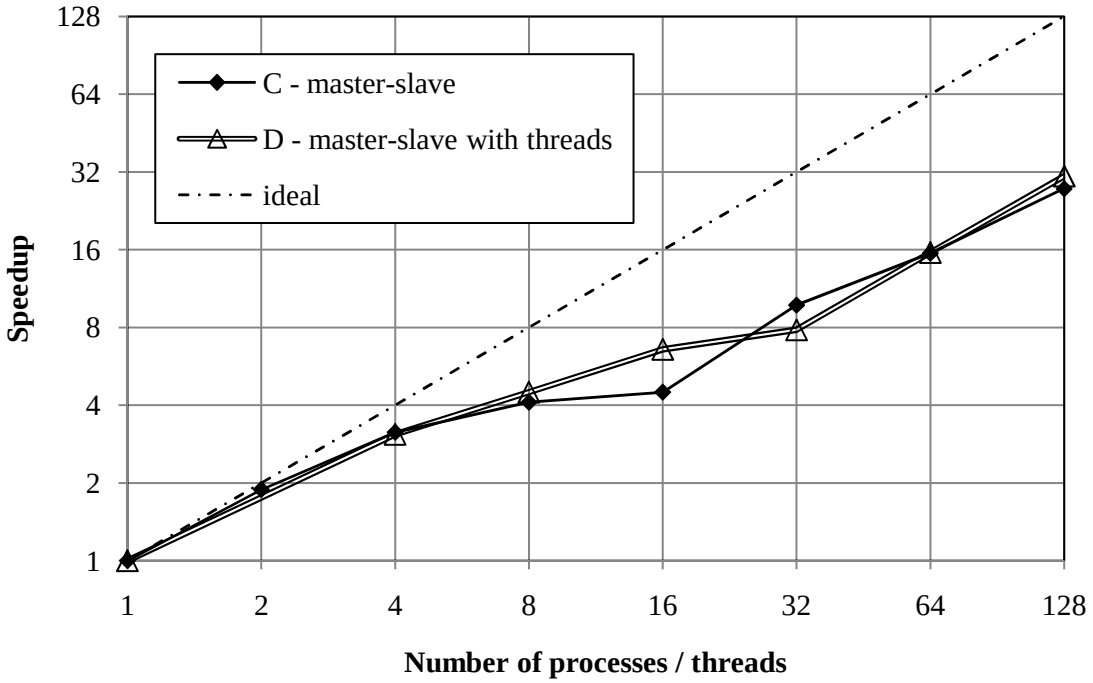


FIGURE 8.4: Acceleration of computations in offline analysis

Figure 5.6 is used as test signal. The analysis is performed 5 times for each parallel processing strategy and the median value of elapsed computation time is computed. Next, the acceleration of computations is determined in relation to the processing time in a single process. The obtained results are plotted in Figure 8.4.

Once again the approaches C and D yield very similar results. Differences are visible for 16 and 32 processes. Thus, it is difficult to determine which strategy performs better in general. The obtained acceleration characteristics is far from the ideal one. The maximum speedup achieved equals 30.8 for 128 instances and D approach (master slave + threads). A distinct deflection of the characteristics can be observed for C approach (master-slave) between 8 and 16 instances. It is most probably caused by the need to engage more than one processing node. The use of threads in D approach compensates for such effect, yet a similar bend is visible for D approach between 16 and 32 threads.

In an additional plot in Figure 8.5 the elapsed times of computations for the master-slave approach (C) are shown. Five measurements for each number of threads are marked and the median line are plotted. It needs to be underlined that the elapsed time depends on the configuration of the recognition engine. For example, a change in the sensitivity of the detectors influences the numbers of events detected and thus, the time needed to

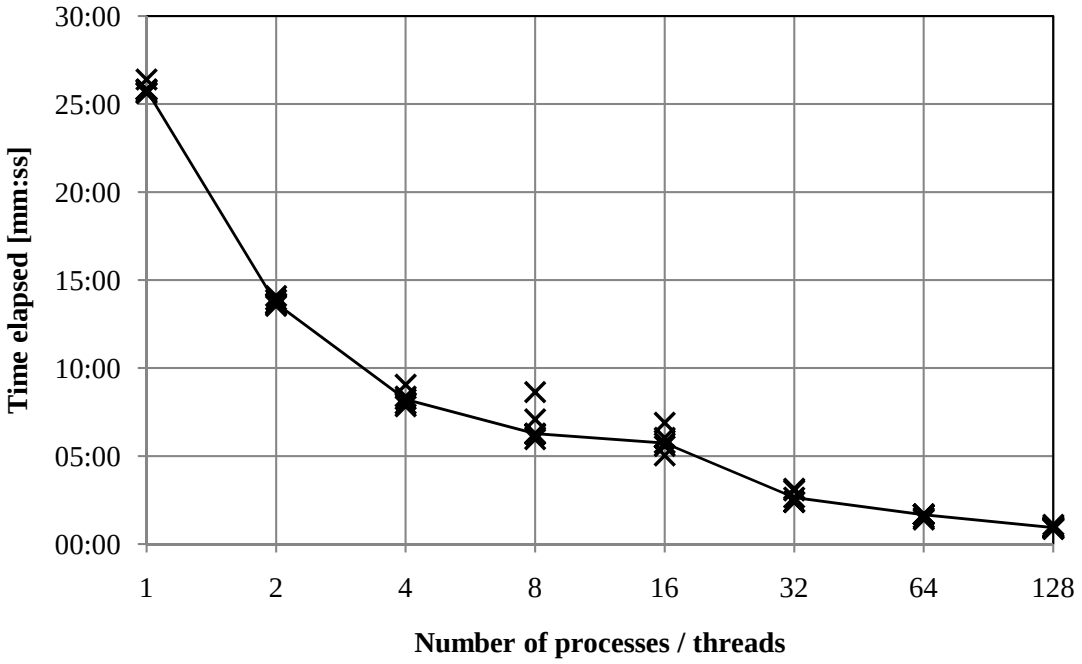


FIGURE 8.5: Elapsed time of computations in offline analysis for master-slave approach (C)

classify the events. In this experiments speech detector and impulse detector were used, both with a sensitivity equal to 0.7.

The time needed to analyze the 24-hour-long recording in a single threads equals ca. 25 minutes. Thanks to the employment of parallel processing on a supercomputing cluster, this time is reduced to less than 1 minute (50.2 sec for 128 threads).

8.2 Acceleration of decision making

This section deals with the time needed to make the decision about the type of detected event. The shorter the decision making time, the better the system's ability to work in practical conditions. For example, when a gunshot is fired in a public space, the algorithms for automatic detection and localization of the acoustic event [6, 7] can be used to instantly pinpoint the location of the shooter. However, every second of delay introduced by the algorithm gives the assailant more time to escape. A reasonable reference point for the decision making time is the human reaction time which, according to general knowledge, varies from 0.1 to 0.3 seconds.

An experiment is presented which evaluates the decision making time of the engineered sound recognition algorithm with different approaches to parallel processing on a supercomputing cluster. First, the definitions and metrics used to evaluate the decision time are presented. Next, the conditions of the experiment are introduced. Finally, the obtained results are outlined and discussed.

8.2.1 Decision process

In general, as shown in Chapter 4, three operations are required to recognize a detected acoustic event:

- buffering of the event's samples,
- feature extraction,
- classification.

As far as the decision making time is considered, the feature extraction is the most computationally expensive and influential operation. However, as it is shown in later sections, the approach to buffering, namely the size of the buffer and how often the events are divided into separate buffers, also has a strong impact on the decision time. The classification operation, performed after the features have been calculated, is considered less computationally demanding.

In the process of assessing the decision making time four time points are important:

- t_{ES} - event start time - which is defined as the moment at which the detection algorithm starts to recognize abnormalities in the audio data stream;
- t_{EE} - event end time - the time after which the event is ready for classification;
- t_{CS} - classification start time - the moment at which the buffered samples are passed to the classifier and feature extraction begins;
- t_{CE} - classification end time - the time at which the event has been classified and the decision is available.

8.2.2 Decision time metrics

Based on the above, we define the metrics employed for assessing the decision making time. Some of the measures can be expressed using absolute units (seconds $[s]$) and some in relative units (seconds per second $[s/s]$), in which case the respective time is divided by the duration of the event. The illustration of the decision process and the defined metrics is presented in Fig. 8.6.

The first metrics relates to the total time needed for detecting, buffering, parameterizing and classifying the event and is referred to as the processing time (PT).

$$PT[s] = t_{CE} - t_{ES} \tag{8.1}$$

$$PT[s/s] = \frac{t_{CE} - t_{ES}}{t_{EE} - t_{ES}} \tag{8.2}$$

The next measure concerns the time needed to make the decision, i.e. the time after the end of the event which is required to obtain the decision, and which it is referred to as decision time (DT). During this time all necessary operations, i.e. buffering, passing the event to the classifier, feature extraction and classification are completed. The DT measure expressed in relative units answers the question: "how long is it necessary to wait for the classification of one second of audio event?" If this value exceeds 1, it is inferred that the algorithm is so slow that the processing of the event lasts longer than the event itself.

$$DT[s] = t_{CE} - t_{EE} \tag{8.3}$$

$$DT[s/s] = \frac{t_{CE} - t_{EE}}{t_{EE} - t_{ES}} \tag{8.4}$$

Classification time (CT) is defined as the time needed to extract the features from the buffered signal and to perform its classification. It does not include the time needed to

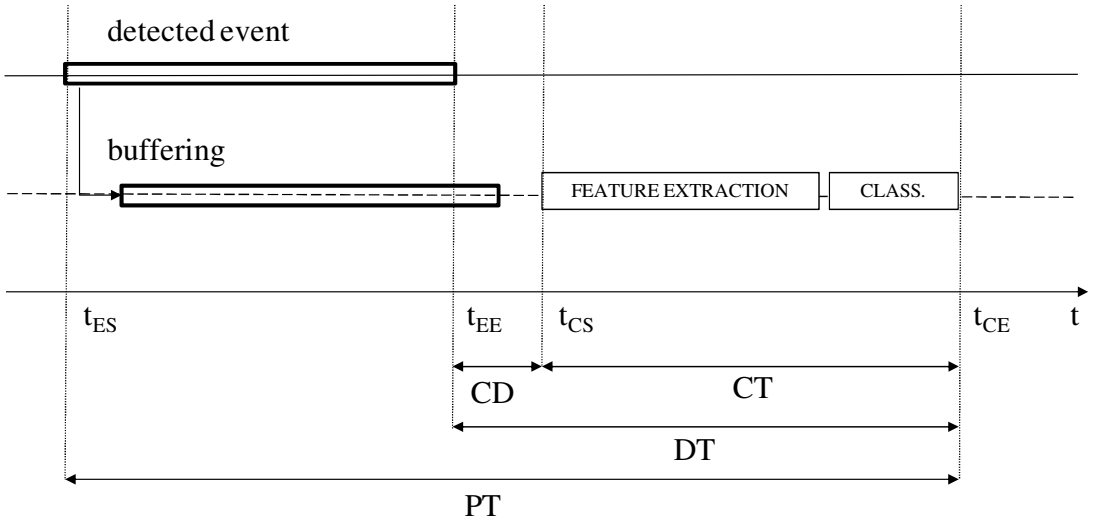


FIGURE 8.6: The illustration of the decision process and decision time metrics

wait until the classifier is free, nor to run the classification task. The relative value (in seconds per second) is important in practice, as feature extraction time depends strictly on the duration of the event.

$$CT[s] = t_{CE} - t_{CS} \tag{8.5}$$

$$CT[s/s] = \frac{t_{CE} - t_{CS}}{t_{EE} - t_{ES}} \tag{8.6}$$

In some cases, after the event has ended, the classifier is busy classifying the previous event and the new event has to wait in a queue before it can be classified. The classification delay metrics (CD) reflects this waiting period.

$$CD[s] = t_{CS} - t_{EE} \tag{8.7}$$

It is worth noting that $DT = CD + CT$.

8.2.3 Parallel processing strategies

In order to optimize the decision making time an effort is made to execute the implemented sound recognition methods in parallel. Four different approaches to parallel

processing are introduced. The four techniques (A,B, C and D) are presented in Fig. 8.7.

A Master-slave

In this approach a slave process is executed every time a new event is detected. The master service (stream algorithm) handles the real-time data processing with the acoustic event detection algorithm. Once detected, the buffered event is passed to a slave algorithm. The slave process then handles the feature extraction and classification over the received signal samples, as shown in Fig. 8.7a. The decision process is completed when the slave algorithm returns the result to the master service. The aim of the master-slave approach is to avoid backlogs in the event classification queue. The drawback is, however, the time required for the execution of the slave process, which makes this approach inefficient as far as the purpose of this work is concerned.

B Complex service

Since master-slave processing is not intended for real-time data stream processing, another approach is proposed, which utilizes a complex service. The approach is presented in Fig. 8.7b. The complex service comprises two simple services: a detection service which handles real-time input audio data processing and an acoustic event detection and classification service, which realizes the functions of feature extraction and classification. When a new acoustic event is detected by the detection service, the buffered samples of the event are passed to the second service in the chain - the classification service. The goal of this approach is to shorten the time needed to execute the classification task. The classifier object, allocated in the second service can handle the buffered event data processing at the same time that the detection object handles the input stream processing. However, if a new event is passed to the classification service while the classifier is busy, the buffered data has to wait until the classifier is free.

C Complex service with multithread classification

To overcome the problem mentioned in approach B, a modification is proposed. In this approach each event data is classified by a separate thread (Fig. 8.7c). Each thread has its own classifier object which has the resources required to parameterize and classify an acoustic event. Once a new event is detected, it is buffered and passed to the classification service. Subsequently, a classification thread is executed (TE) for the new event. The threads are synchronized in a thread pool. When a thread finishes and provides its result, the decision is made available.

D Complex service with sequential feature extraction

In the previous approaches the feature extraction (and the following classification) is only started after the event has finished. The last approach incorporates another strategy (shown in Fig. 8.7d). Whenever a frame of the detected event is present in the buffer, feature extraction from this frame is performed. The length of the frame equals the length of the classifier's frame - a_f. After all frames comprising the detected acoustic event have been gathered, classification is fired. The approach enables a significant shortening of the decision process.

8.2.4 Experimental methodoloogy

To assess the speed of decision making a test signal was prepared. To simulate a *stress* situation, in which there might be difficulties with *online* decision making, the following settings of acoustic events are employed:

- A series of a few impulsive events (gunshots, explosions), i.e. 3-5 impulsive events placed close in time (less than 1 second apart) - 4 sequences with 15 events were used;
- A medium-long event (scream or breaking glass) surrounded by impulsive events (gunshots, explosions) - 11 sequences with 36 events were used;

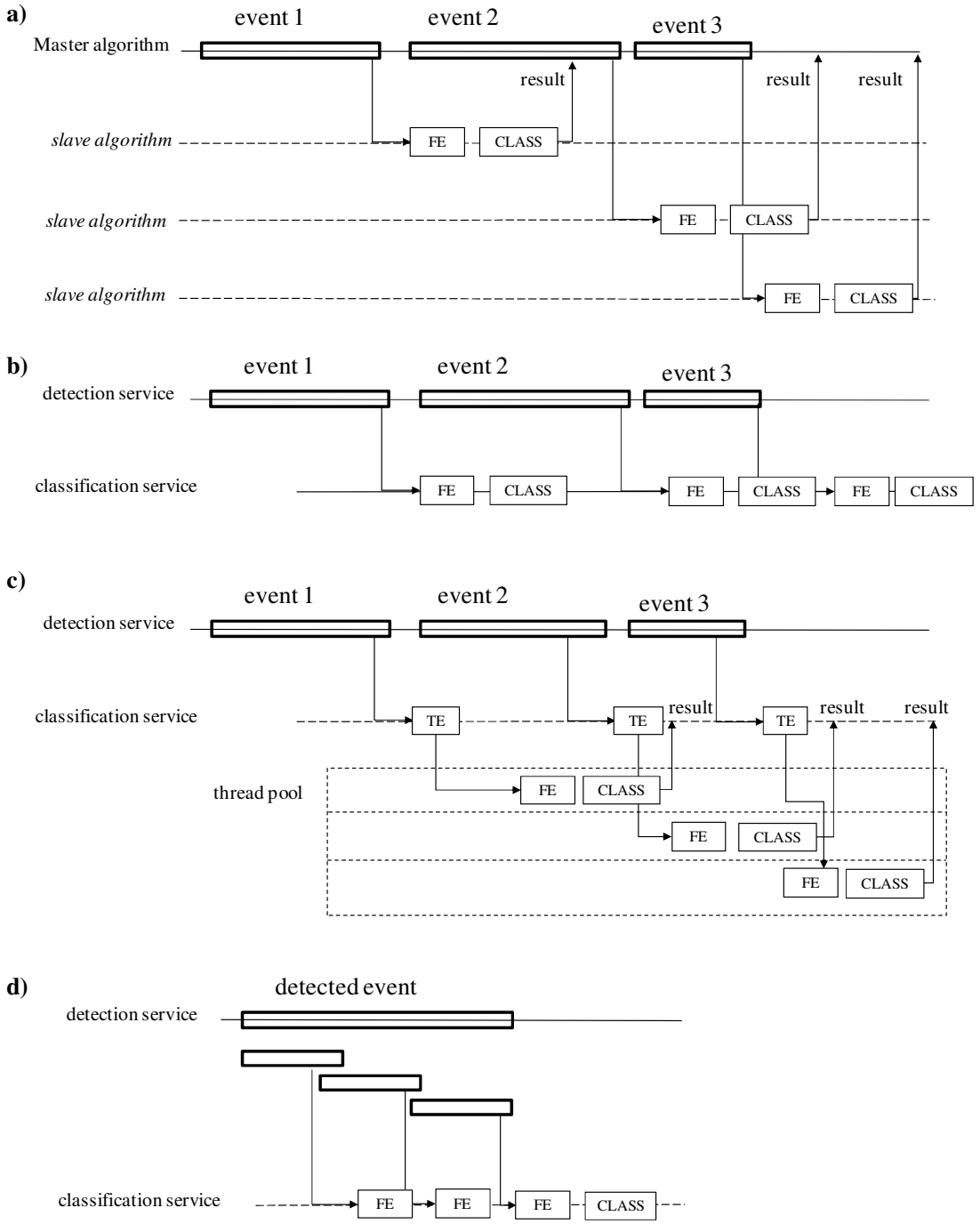


FIGURE 8.7: Processing flow in the employed parallel processing strategies: a) master-slave approach, b) complex service approach, c) complex service with multithread classification, d) complex service with sequential feature extraction

- Very long events (noisy events, non-threatening, 8-16 seconds long) followed by one or two threatening impulsive events (gunshot, explosion) - 10 sequences with 23 events;
- Very long events followed by another long or medium-long event - 3 sequences of 6 events;
- isolated events of all types, not pertaining to any of the mentioned sequences - 11 events.

A total number of 91 events are included in the experiment signal. The average duration of the acoustic event is 1.9 seconds, with a minimum duration of ca. 300 ms and a maximum duration of 16 seconds. The duration of the whole test signal equals 4 minutes 49 seconds. The break between events varies from 500 ms to 2.5 seconds.

The test signal is then analyzed by the sound recognition service in simulated online mode, i.e. the data from the experimental signal is fed into the service according to the sampling rate of 48000 samples per second. The detected events and their respective start and end times (defined in Section 8.2.2 and Fig. 8.6) are written to a log file. Next the timestamps are analyzed to calculate the decision time metrics: CT, CD, DT and PT. Three parameters of the sound recognition engine are changed during the experiment:

- analysis frame (a_f) - two values are used: 200 ms and 500 ms;
- frame overlap factor (OL) is equal to 25, 50 or 75%;
- buffer length (b_{len}) is assigned one of the three values: 2 s, 3 s and 5 s.

Each of these parameters has an impact on the decision time process, which is evaluated in the conducted experiment. We use representative values which yield a reasonable complexity and an adequate temporal resolution. The analysis frame has to be longer than the employed block size in Welch's method (equal to 85 ms). The buffer length corresponds to the approximate length of average short, medium-length and long acoustic events. The overlap factor determines the amount of computation which has to be

executed in order to parameterize the event. Larger overlap yields a deeper insight into the temporal qualities of the signal, which can improve the efficiency of classifying non stationary events. By changing the overlap, we intend *stress testing* of the algorithm and determine by how much the decision making time is extended, when a larger overlap is used. The combination of the values of these parameters yields 18 analyzes of the test signal for every parallel processing strategy. For every set of values and a particular processing method the maximum and mean values of the calculated decision time metrics are extracted. The results are aggregated and presented in the following subsection.

8.2.5 Decision making time results

In this section the results obtained during the experiment are presented. We focus on each of the defined decision time metrics and its dependence on the processing parameters (a_f, b_{len} and OL). The four employed strategies for parallel processing are compared to the *baseline* algorithm, by which we understand the algorithm operating according to the diagram in Figure 4.1, which introduces no parallel computations.

Decision time

Decision time is the most important metrics which tells us about the amount of time needed to wait for the decision after the event has ended. Minimization of this measure is the main purpose of the work presented in this section. The results of the average decision time per second of event duration (DT $[s/s]$) for every parameter combination and processing method are presented in Table 8.1. The desired values are under 1 second, which means that the decision time is no longer than the duration of the event. This requirement is met both by baseline algorithm, methods C (complex service with multithread classification) and D (sequential feature extraction) and, with some exceptions, by method B (complex service). Method A (master-slave approach) fails to yield reasonable decision times. The cause of such poor performance is the amount of time needed to execute the slave process in the KASKADA framework. Another observation that can be made is that the decision time is strongly correlated with the overlap factor OL. The greater the overlap, the greater the number of frames that need to be processed, which

extends the decision making time. It can also be seen that, as a rule, the decision time is longer for longer analysis frame a_f and buffer length b_{len}. The influence of the analysis frame length on DT can be explained by the fact that some features are calculated longer for longer frames (e.g. temporal features). The buffer length is important in cases in which two events are close in time to each other. If the buffer is longer, there is greater probability that the classifier will still be occupied when a new event is detected, since it takes more time to classify the last event. It is worth noting that this dependence is apparent in the baseline algorithm (e.g. for 250 ms frame and 75 % overlap we have 0.650 s/s with 2 s buffer vs. 0.808 s/s with 5 s buffer) and method B - complex service (0.746 s/s vs. 1.130 s/s), which are sensitive to mutual exclusions in classification loop (explained in Section 4.4) and not noticeable for method C - multi-thread classification (0.623 s/s vs. 0.6 s/s), which solves the problem of exclusions in the critical section of the classifier.

TABLE 8.1: Decision time per second (DT [s/s]) for different parallelization strategies

b_{len} [s]	a_f [ms]	OL	baseline	A	B	C	D
2	250	25%	0.272	9.607	0.304	0.279	0.031
2	250	50%	0.363	9.235	0.375	0.427	0.048
2	250	75%	0.650	7.582	0.746	0.623	0.058
2	500	25%	0.282	11.371	0.302	0.294	0.031
2	500	50%	0.363	9.715	0.369	0.366	0.048
2	500	75%	0.596	8.297	0.718	0.568	0.053
3	250	25%	0.268	6.748	1.225	0.301	0.026
3	250	50%	0.376	5.771	0.646	0.350	0.025
3	250	75%	0.863	5.915	0.406	0.590	0.027
3	500	25%	0.277	7.301	0.381	0.272	0.041
3	500	50%	0.380	10.185	0.552	0.345	0.044
3	500	75%	0.774	5.749	1.243	0.552	0.048
5	250	25%	0.325	2.986	0.286	0.264	0.024
5	250	50%	0.485	3.523	0.550	0.343	0.020
5	250	75%	0.808	3.784	1.130	0.600	0.017
5	500	25%	0.323	3.589	0.318	0.249	0.032
5	500	50%	0.496	5.489	0.460	0.349	0.039
5	500	75%	0.742	4.043	1.049	0.615	0.049

As far as these results are concerned it is observed that neither method A (master-slave), nor B (complex service) improves the decision making time. Some improvement is noticed after employing the complex service with multi-thread classification (C), although not in all cases. The last method (complex service with sequential feature extraction)

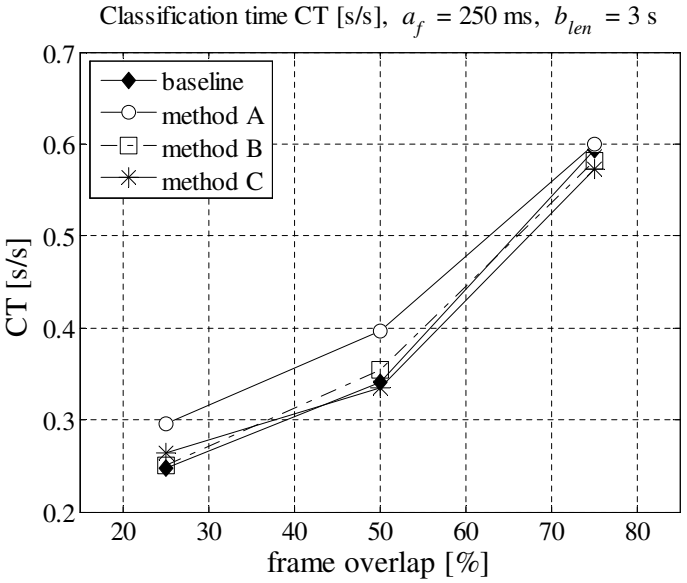


FIGURE 8.8: Classification time (CT [s/s]) for different parallel processing strategies

clearly outperforms all previous strategies by reducing the decision time below 100 ms, which equals the time needed to process the last short-time frame of the signal.

Classification time

Since the most significant operation in the decision making process is classification (here treated together with feature extraction), we take a closer look at the time needed to parameterize and classify the event. This time is reflected by the CT metrics defined in Equation 8.5 and Equation 8.6. This measure does not include the possible delay before the start of classification. The results of average classification time per second for a chosen frame length and buffer length are presented in Figure 8.8. It is noticeable how the classification time depends on the overlap factor. The dependence is similar to the one observed in Table 8.1. It is worth noting that this metrics does not pertain to the last processing method (D), since in this case feature extraction starts before the event is finished and the classification time calculated according to Equation 8.5 would yield negative values. The surprising finding is that for overlap factors equal to 50% and 75% the parallel processing does not shorten the classification time compared to the baseline algorithm. Some improvement is noticed for the 25% overlap for methods B and C. The gain, however, is very small.

Classification delay

As it was observed in the previous subsection, classification time *per se* does not significantly benefit from the employed time enhancement approaches. Therefore, it is noted that classification delay (CD) is the factor which can be improved by optimization of decision making. The measure was defined in Eq. (8.7). The results of CD evaluation for a chosen frame length and overlap factor are presented in Figure 8.9 Only the baseline algorithm and methods B and C are analyzed. Method A yields overly high values of CD and the metrics does not apply to the method. The plots in Figure 8.9 are drawn in relation to buffer length. It is noticeable that for the baseline method and method B the dependence on b_{len} is strong. Moreover, utilizing a complex service (B) does not shorten the average classification delay. In contrast, the CD values are higher for method B than for the baseline algorithm. The longer the event buffer, the higher the CD observed, because longer events occupy the classifier's resources for a longer time, thus leading to a delay in the recognition process. The employment of multi-thread classification (C), however, contributes to a significant decrease in classification delay. Moreover, the dependence on b_{len} is no longer apparent.

In Table 8.2 the average and maximum values of classification delay for all values of a_f, b_{len} and OL are presented. It is also demonstrated how they contribute to the average and maximum processing time (PT). It can be observed that method A introduces a huge delay which significantly extends the decision process. One can also observe that approach C does shorten the average and maximum classification delay, but it does not relate to the average values of processing time (it does, however, improve the decision making time in some cases, see Table 8.1 and Figure 8.10). Finally, it is worth noting that the classification delay metrics does not apply to approach D. As it can be seen in Figure 8.7d, the feature extraction in this method starts before the event has ended. The results prove that it leads to a significant improvement in the decision making time.

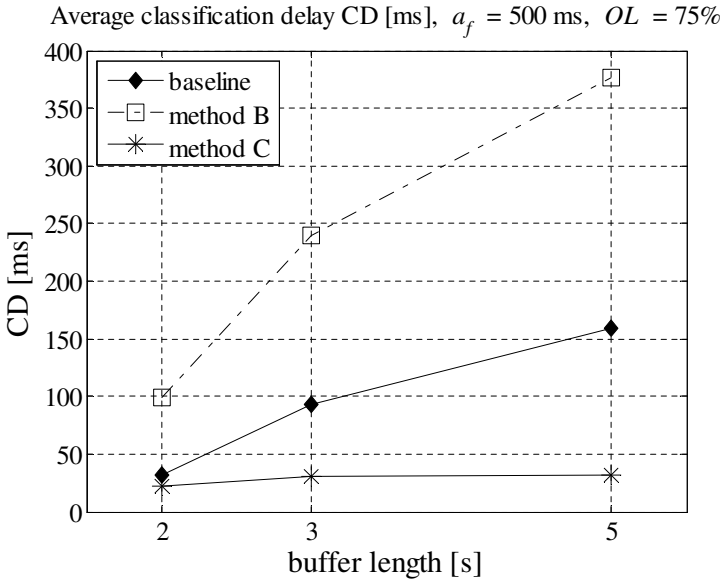


FIGURE 8.9: Classification delay (CD [ms]) for different parallel processing strategies

TABLE 8.2: Classification delay (CD) and processing time (PT) for different parallel processing strategies

	baseline	A	B	C	D
average CD [ms]	47.63	7322.79	116.84	26.29	N/A
maximum CD [ms]	1902.77	15423.21	3079.81	635.56	N/A
average PT [s]	3.310	10.572	3.532	3.489	3.111
maximum PT [s]	8.093	21.072	9.449	9.595	7.176

Processing time

Finally, we present the comparison of processing time per second of detected event. The results are shown in Figure 8.10. The 500 ms frame and 75% overlap are chosen for this analysis. It can be seen that the baseline method requires roughly 1.5-1.7 seconds of processing for every second of detected event. Method A achieves values which do not fall within the scale of the plot. Approach B performs slightly worse than the baseline method. Method C, employing multi-thread classification, yields some improvement, especially for a longer buffer. Method D again outperforms the other approaches, by reducing the processing time per second close to the minimum of 1 s/s.

8.2.6 Conclusions from decision time evaluation

The following conclusions are drawn from the conducted experiment for decision making time evaluation:

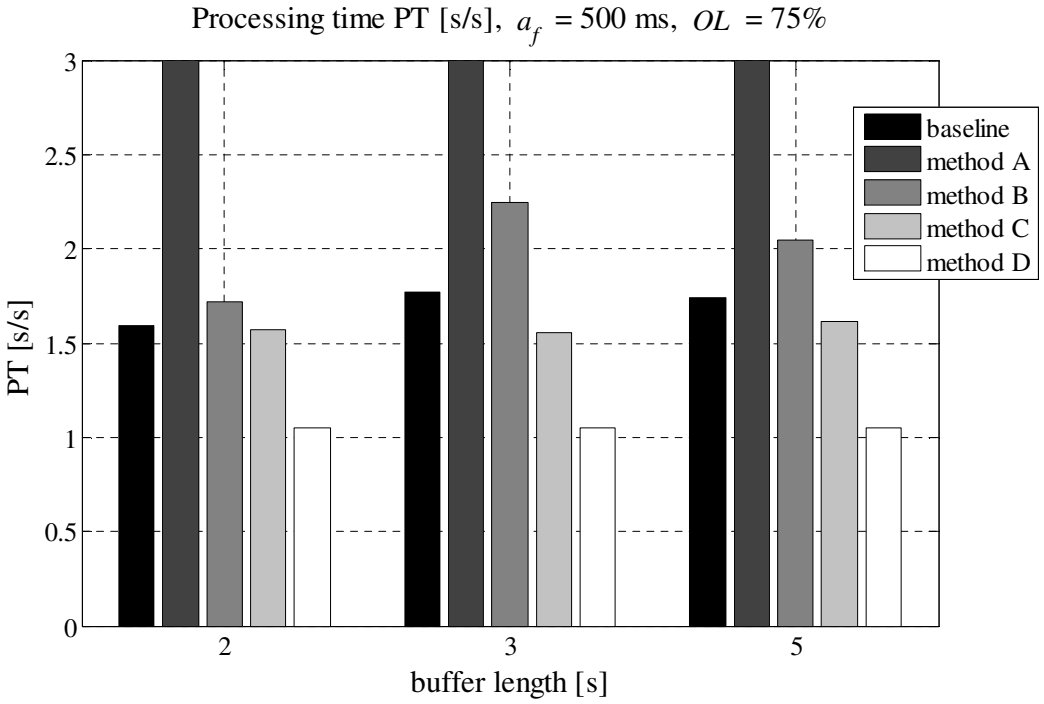


FIGURE 8.10: Processing time (PT [s/s]) for different parallel processing strategies

1. The master-slave approach is not appropriate for accelerating the decision making in online acoustic event recognition. It has been proved that executing the slave tasks consumes too much time and introduces a significant delay, which is unaffordable in time-critical decision making. Moreover, performing feature extraction in the slave task does not speed up the parameterization process.

2. The time needed to extract the features from the signal is not shortened if the feature extraction is performed by a slave task (method A) or another service (methods B,C). To increase the speed of feature extraction, other techniques should be used, e.g. the parallelization of the parameter calculation functions.

3. Employing the complex service scheme (approach B) does not lead to an improvement in the decision making time. It was shown that both the average classification time and the classification delay are extended in complex service approach compared to the baseline system. Therefore, in order to benefit from the complex service form additional mechanisms need to be implemented.

4. Parallel processing does, however, shorten the maximum classification delay, which is a considerable benefit for an automatic sound recognition system.

5. Employing multi-thread classification, in which a separate *Classifier* object is constructed for every event, leads to substantial improvements in classification delay and total processing time (approach C).

6. The last approach, i.e. complex service with sequential feature extraction (method D), outperforms the other presented methods by reducing the decision time to the desired level. Because of this approach the decision is made instantly at the end of an event.

The acceleration of decision making achieved with the employed parallel processing strategies is significant. The approach D, in particular, ensures that the decision is obtained almost instantly after the cessation of the event. It is also worth mentioning that the developed mechanisms for acceleration of decision making do not influence the recognition accuracy. The acoustic signal parameterization and classification operations are identical in all approaches. Therefore, we do not observe any deterioration of the recall, so that precision rates of the recognition engine compared to the baseline system. However, the length of the analysis frame (a_f) and overlap factor (OL) have an impact on the classifier's performance.

An important conclusion is that the performance could be improved even further provided that the feature extraction is executed in parallel. The research by Maka and Dziurzański [141] shows that a significant gain on computational time can be achieved when the features are extracted in parallel by utilizing standard OpenMP directives in ANSI C code. Such mechanism could also be successfully used in a Linux-based supercomputing cluster such as the one used in this thesis. It is a very promising direction for future research on the subject of accelerating the decision making in sound event recognition.

Chapter 9

Conclusions

In the dissertation, the methods for sound event recognition were examined in the environment of a supercomputing cluster. The employed algorithms were introduced and the experiments which validate the efficiency of recognition of hazardous events were featured. The works aiming at reducing the time required to process the acoustic data on a supercomputing cluster were presented. In the final chapter a discussion of the goals of this work is provided, as well as the possible applications and plans for further development.

9.1 Author's original work

All algorithms and methods featured in the work, apart from code realizing mathematical operations such as Fourier transform and SVM classification, are author's original work. The sound source localization methods, featured in Section 7.1 and in Section 7.2.4, are an exception, since they were developed by a coworker, Dr. Józef Kotus, and are evaluated together with sound event recognition in some experiments mentioned in this dissertation. The methods developed by the author are results of literature studies and original thoughts. Here, the most representative achievements are listed, which to the best knowledge of the author extend the current state of the art.

1. **Flexible adaptive methods for detection of acoustic events** - The sound event detection algorithms, presented in Section 4.1, are both computationally light (as opposed to detection-by-classification approach known from the literature) and advanced as far as adaptation is concerned. The sophisticated adaptation of detection thresholds, taking into account the complexity of the acoustic background, surpasses the simple adaptation methods known from the literature [64]. In particular, the detection thresholds are adapted not only to average sound level in the acoustic background, but also to its variance and temporal change rate (see Section 4.1.3). Moreover, the introduced detectors are flexible when it comes to choosing the detection parameter. Thus, the algorithms can be easily suited to detect new types of events in new types of environments. The introduced detection approach was tested in realistic conditions and achieved good results. It was shown in Section 7.2.1 that adaptation improves the performance of the detection algorithm.

2. **Implementation of a sound recognition engine on a supercomputing cluster** - according to the review presented in Chapter 3, the applications exploiting parallel processing of audio data are more and more frequent, yet some fields are still to be explored. This thesis presents pioneering work devoted to recognition of acoustic events on a supercomputing platform. It is based on a project, in which an innovatory framework for processing multimedia data streams in a supercomputing environment was created. The author's work was focused on developing supercomputing services related to sound recognition, executing and evaluating them on real-world data and conducting experiments to prove the efficiency of supercomputing. The results presented in Chapter 8 show that the employed methods for parallel processing of acoustical data improve the usability of the engineered methods and reduce the time needed to make the decision.

3. **Availability to operate online in different conditions** - to the knowledge of the author none of the published works featured a system operating online and tested in such diverse acoustic conditions. The state of the art systems are either tested on isolated events [43, 49, 53] or require fitting to specific events or

noise types [42, 80]. The engine proposed in this dissertation has the capacity of recognizing different types of events and work in various conditions, which was confirmed by experiments. It also was tested in online operations while processing acoustical data for multiple days.

4. **Precise evaluation of recognition performance in noise** - To properly investigate how the engineered methods react to different acoustic conditions, an experiment methodology ensuring precise control over noise type and intensity was proposed. The experiment featured in Section 7.1.3 is in itself innovatory and more thorough than the evaluation methods presented in the literature. Thanks to the employed approach precise calculation of SNR is possible, thus providing valuable insight into the recognition performance in realistic conditions. However, the environment simulated in an anechoic chamber is not identical with a real one. Hence, the evaluation was also performed in real-world conditions, using real-life events and noise. The achieved results were consistent with the ones achieved during simulations, which proves the correctness of the assumed approach.

9.2 Discussion of scientific theses

Three scientific theses were posed in this work. Below, the discussion in support of each of the theses is provided.

To reach the goal of establishing a system for efficient detection and recognition of hazardous events, partial goals were achieved. Firstly, the algorithms for detecting and classifying sound events were created. Secondly, the engineered methods were prepared for online operation. Finally, the developed sound recognition engine was implemented in the environment of a supercomputing cluster, in which it was evaluated.

Before the sound recognition engine was tested in realistic conditions, the evaluation of the sound event classifier on the training set was performed. A database of signals was utilized, comprising samples of such hazardous events as: explosion, breaking glass gunshot and scream, as well as other, non-threatening sounds. The optimum size and

composition of the feature vector was found and the optimum parameters of the classification model were obtained. It was shown that the sound classifier employed is capable of discerning between these classes of acoustic events with high accuracy. The results obtained on the training set are presented in Chapter 6. A number of 1278 of all 1301 events from the training set were recognized correctly in cross-validation testing. The achieved results confirm that the methods used for the analysis of acoustical signals were chosen and implemented correctly.

The accuracy achieved on the training set served as reference for evaluating the engine on real-world data. The practical conditions considered are indoor and outdoor environments. The developed methods were tested in such conditions as: outdoor urban space (with traffic and railway noise), bank operating hall, public event hall and indoor space with cocktail-party noise.

It needs to be underlined that achieving high efficiency of acoustic event recognition in practical conditions is a challenging task, due to noise and distortions which affect the signal, thus deteriorating the performance of the classifier. It was shown in Chapter 6 that in evaluation on the training set nearly 98% overall accuracy can be obtained, which yields very small loss. In Chapter 7 several experiments were presented whose aim was to determine the classification loss achieved in practical conditions. It was shown that the performance of the sound recognition engine depends strongly on the type of noise present and on the SNR. An experiment was conducted in a simulated environment in which both these qualities were under precise control. The results showed that the recall and precision rates are in some cases (i.e. for some values of SNR and some types of events) as high as those achieved on the training set.

The experiments conducted in real-life conditions have yielded similar results to those performed in the simulated environment. However, the SNR values in practical conditions could not be precisely determined. As far as the recognition rates are considered, high recall rate of gunshots was obtained in bank operating hall and in outdoor urban conditions. The recall and precision rates for scream and broken glass are also high (ca. 0.9) provided the SNR equals 10 dB or more (see Section 7.2.2). Moreover, it is observed that false positive classifications (i.e. classifying a non-threatening as hazardous)

are more frequent in realistic conditions that false negative classifications (i.e. classifying a dangerous sound as non-threatening). In the context of a security surveillance system such false positive errors are of lesser concern. Although lowering the overall classification accuracy, false alerts can be later discarded by the security personnel. The rate of false alerts can also be efficiently lowered by employing adaptation. Therefore, a substantially larger loss is generated by false negatives, which fortunately are not as numerous. Again, it needs to be underlined that the number of false positives can always be lowered at the cost of increasing false negatives and vice versa. From the practical usability point of view it is important that the examined decision system can be configured in such a way that the security of people and property is substantially increased while the loss generated in the decision process is acceptable.

In the light of the achieved results it can be assessed that the engineered techniques are applicable in real-world audio-based or audiovisual surveillance and are able to operate with a reasonably low loss. The achieved recognition rates are high enough to detect typical threatening events (such as explosion, breaking glass, gunshot and scream) in practical circumstances, thus improving the security level. This proves the scientific thesis of the dissertation:

1. **The developed methods for detection, parameterization and classification of selected hazardous acoustic events enable sufficiently low loss achieved in practical conditions to be used for security surveillance purposes.**

The main difficulty in acoustic event detection in real-world conditions is that the majority of the input audio data constitutes the acoustic background. It is very common that these noisy sounds are recognized as threatening, thus leading to a false alarm. This underlines the importance of sound detection algorithm, whose purpose is to separate the relevant acoustic events from the background sounds. In the dissertation threshold-based detection was employed with adaptation of the detector's threshold to the parameters of the acoustic background.

The mechanisms for adapting the detection threshold to the changing conditions were proposed in Section 4.1.3.. It was shown that the adaptation enables the detector to closely follow the time-varying characteristics of the environment. The adaptation feature was successfully used both in indoor and outdoor environment (bank operating hall and urban space). It was shown that thanks to adaptation of detection threshold the security level of the area under surveillance is increased, e.g. by automatically lowering the threshold at night.

In Section 7.2.1 efforts were presented to reduce the number of false positive results, while maintaining the rate of true positives. It was shown that employing adaptation leads to a decrease in the number of false positive detections. Moreover, three different adaptation strategies were evaluated. The most simple one, *simple adaptation*, was based on following the average sound level. In the *double adaptation* approach, the variance of the sound level was also considered. Finally, the method called *triple adaptation* was based on average sound level, its variance, and the rate at which the sound level changes with time. The three approaches were compared in an experiment aiming at detecting gunshots in outdoor urban noise based on sound level. The result is that the most advanced adaptation method (triple adaptation) yields the most favorable performance (lowest EER) of all the considered approaches.

The achieved performance of the adaptive sound event detector proves the scientific thesis:

2. **The proposed way of adaptation of the detection threshold to the variance and dynamics of the level of the acoustic background reduces the detector's equal error rate compared to the adaptation to average sound level.**

The most innovatory aspect of this thesis is the employment of a supercomputing cluster for recognition of events in audio data streams. The engineered methods were implemented in a specialized framework operating on a supercomputer. Supercomputing services were created to make the developed sound recognition engine available to the

end user, as described in Section 5.1.3. The implemented services were shown to operate reliably in online mode in real-life conditions.

Parallel processing techniques were introduced to speedup the processing of audio data. The shorter the time needed to recognize the audio data, the more time the security authorities have to react to a threatening situation. The parallel approach was employed in online sound event recognition in Section 8.2. Different parallel processing strategies were implemented and assessed to reduce the decision making time. It was shown that the decision making is efficiently accelerated compared to the baseline system and the time needed to recognize the detected event equals ca. 0.1-0.2 seconds, which is comparable to human reaction time.

It should be noted that the achieved time only corresponds to the latency introduced by the sound recognition algorithmic chain, disregarding the latency of the KASKADA platform (inside the supercomputing environment) and the delay introduced due to transmitting signals via RTSP (outside the supercomputing platform). These factors have not been extensively measured and are a topic for further investigation. Nevertheless, the delay introduced by transmission medium or the supercomputing platform are independent of the work carried out by the author of this thesis.

The achieved result is also comparable to state-of-the-art achievements in the field. Saraclar et al. published the research on a low-latency news transcription system [177]. The latency of their system consists of fixed latency (500 ms approx.) and variable latency which equals ca. 1 second on average. In a work by Salvi on *very low latency speech recognition* a constraint of maximum 100 ms delay is mentioned [178]. In a low-latency speech recognition system introduced by Seward the optimum performance was achieved for a latency equal to 150 ms [179]. It is also reasonable to compare the achieved delay with low-latency audio applications. The source of latency in PC audio is similar to the source of latency in the sound recognition engine examined in this work, namely it is buffering. In ASIO (Audio Stream Input/Output) drivers the minimum latency of 25 ms can be achieved, whereas with other types of drivers (e.g. MME - MultiMedia Extensions) a latency as high as 750 ms is encountered [180].

Considering the state-of-the-art results, the performance the engineered sound recognition algorithmic chain can be regarded as nearly real-time. Such low latency is achieved thanks to parallel processing of audio data employing the resources of the supercomputing cluster. The scalability of the KASKADA platform also ensures that the performance will be maintained if multiple sources are recognized simultaneously. The obtained results prove the scientific thesis:

3. **The implemented parallel processing schemes on a supercomputing cluster enable nearly real-time performance of the hazardous sound recognition algorithmic chain.**

9.3 Possible applications and further development

The developed sound recognition engine has a high potential for practical application. The mechanism of supercomputing services is highly flexible and facilitates the exploitation of the engineered methods. The KASKADA framework with the specialized software and hardware architecture only requires the source of audio data stream to be connected to the IP network. The fields in which the results of this work can be utilized are:

- Acoustic surveillance of urban space, e.g. detection of gunshots, traffic hazards, disturbance, but also noise monitoring. The signals from acoustic sensors mounted in the municipal infrastructure can be sent to the supercomputer center for analysis. The results can be directly passed to the authorities. The processing capabilities of the supercomputing platform enable efficient processing of numerous data stream and fast online decision making.
- Acoustic surveillance of public events such as concerts or sports events. The system is capable of processing numerous streams simultaneously, which make it particularly useful for large events and spaces. It was shown that the localization algorithms can be used to localize the sound source, in particular the person causing the hazardous events.

- *On demand* surveillance, e.g. ordered by private owners who have the need to detect threats to their property. In such case, a non-trained user only requires a microphone and a computer with Internet connection to send the audio data stream to the supercomputing platform and receive the event recognition results.

- Offline analysis of registered audio data. The law enforcement authorities can use the developed service for offline analysis of large amounts of registered data. Thanks to employment of parallel processing the days of registered audio can be analyzed in minutes and the relevant signals can be selected for further investigation.

Regarding the application in surveillance of outdoor urban space, the possible coverage should be discussed. According to the website of the leading commercial gunshot detection system [90], the gunshot detection solutions can be divided into two categories: point protection and wide area protection. The former are based on single sensors and enable 50-200 meters radius range. The latter are based on multiple sensors and cover large areas, e.g. cities. In the experiments featured in this dissertation, described in Section 7.2, gunshots were emitted from ca. 100 meters. The detection and recall rates for gunshots were still very high, event from such large distance. Thus, it can be stated that the recognition engine developed in this dissertation yields comparable performance to the commercial ones, as far as range is concerned. Provided that one sensor enables detection of hazardous events in 200 m radius (covering ca. 1/8 km^2), it would take about 50 sensors to cover a district of a city.

The successful implementation of the engineered sound recognition algorithms on a supercomputing platform opens the door for other distributed implementations. A particularly interesting case is the employment of cloud computing. A number of geographically spread sources could be connected via the Internet to the computing cloud, in which the methods for detecting threatening audio events would be installed. The advantage of such architecture is its high flexibility and scalability.

The results of the dissertation can also be useful in other fields than security surveillance. The parallel processing of audio data streams can be utilized i.a. for context-based search

of audio. By comparing the features extracted from the sample recording provided, similar recordings can be found, e.g. recorded in similar conditions or involving the same persons. Both the engineered feature extraction and classification techniques can be used in such case. The related fields such as music information retrieval or speech recognition can also benefit from parallel processing of audio data.

In further development it is essential to enhance the performance in noisy conditions. A closer investigation of signal features could lead to identifying those which are more affected by additive noise and deteriorate the classifier's performance. Further and more elaborated experiments devoted to evaluation of the recognition engine in real-life conditions could also lead to valuable conclusions. In the context of parallel recognition of audio streams it is an interesting notion to use multiple microphones to strengthen the decision. Loud events, such as gunshots, should be heard over a large area. Hence, the synchronization of recognition results from spaced sensors could be used to reduce the false alert rate. Finally, a substantial improvement could be achieved by joining the acoustic and visual modality. The topic was addressed in experiments featured in Section 7.2.4, where the localization data were used to steer the PTZ camera. As it was mentioned in Section 2.9 audio and video data can be used together to strengthen the decision in automatic surveillance and could also be applied in the KASKADA framework basing on algorithms presented in this thesis.

9.4 Privacy issues

In 1949 British writer George Orwell published his world-famous novel "Nineteen eighty-four" [181]. The book featured an all-knowing, all-seeing entity referred to as *Big Brother*, who soon became the symbol of a police state and excessive surveillance pushed to the point of invading the citizens' private lives. Orwell's concept has been often used to discredit the development of surveillance systems. The main fear causing the society's hostile attitude towards surveillance technology is the loss of privacy. Therefore, visual and acoustic monitoring of streets and public places is often subject to criticism. The acoustic surveillance in particular, causes people to fear that their private conversations

are being overheard or recorded, stored and analyzed. It needs to be underlined that the technology used by the author of this dissertation and described in the literature, namely acoustic event recognition, does not provide the means to analyze the *content* of speech. As it is shown in the dissertation, the signal is analyzed in the domain of *features* which provide the information about the spectral and temporal qualities of sound, not the meaning of utterances. As far as human voice is concerned, the author of the dissertation focuses only on determination if the sound is regular spoken voice or scream.

However, every surveillance system has to register data in order to operate. Such data are often stored for a specified amount of time. Efforts are made to limit the need for storing data which are not related to any threatening or illegal actions. The project ADDPRIV (Automatic Data relevancy Discrimination for a PRIVacy-sensitive video surveillance) was conducted in the years 2011-2014 [182]. Its aim was to erase the multimedia data which are not relevant from the point of view of detection of abnormal events in order to protect the privacy of the people involved in the recorded scenes. Another method to improve the privacy level of automatic surveillance is to employ video stream anonymization, which allows for hiding the faces or license plates present in the recordings [183]. Reversible anonymization is a technique which allows the authorized people (e.g. the police) to extract the original image from the anonymized recording with the use of cryptography [184].

As far as the registered audio material used in this thesis is concerned, the author states that the recorded data do not contain sensitive material such as private conversations of unwitting persons. The recordings were performed either in public spaces where people were located far from the microphone (with a pertinent disclaimer displayed in the area under surveillance) or with actors who agreed to use the recordings of their voice for experiments.

Appendix A

List of selected features

The following features are used in the decision process. See Section 4.3 for feature definitions and formulae.

Audio Spectrum Envelope (ASE) 5,6,8,15,16,17,21,24,25,28,30,32,34
Audio Spectrum Centroid (ASC)
Cepstral Crest Factor (CCF)
Crest Factor (CF)
kurtosis
Log Attack Time (LAT)
MFCC 1,2,4,5,6,8
Peak-Valley Difference (PVD)
Periodicity
Spectral Energy (SE) 1,2,3,4,5,7,8
SFM
SFMa 10,15,16,18,19,23,24
SFMb 1,2,3,4,7
Spectral Roll-Off
Spectral Slope
Speech Energy
Zero Crossing Rate (ZCR)

List of Figures

List of Tables

Bibliography

[1] M. Bullock, "The evolution of surveillance technology beyond the panopticon," Master's thesis, University of California Santa Cruz, 2009.

[2] A. Czyżewski, G. Szwoch, P. Dalka, P. Szczuko, D. Ellwart, T. Merta, K. Łopatka, L. Kulasek, and J. Wolski, *Video Surveillance*, ch. Multi-stage video analysis framework, pp. 145–171. Intech, 2011.

[3] "Mayday Euro 2012 project website." http://mayday2012.gda.pl (visited 2013-07-10).

[4] J. Kotus, K. Łopatka, A. Czyżewski, and G. Bogdanis, "Processing of acoustical data in a multimodal bank operating room surveillance system," *Multimedia Tools and Applications*, published online http://dx.doi.org/10.1007/s11042-014-2264-z, 17.10.2014.

[5] K. Łopatka and A. Czyżewski, "Acceleration of decision making in sound event recognition employing supercomputing cluster," *Information Sciences*, vol. 285, no. 1, pp. 223–236, 2014.

[6] J. Kotus, K. Łopatka, and A. Czyżewski, "Detection and localization of selected acoustic events in acoustic field for smart surveillance applications," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 5–21, 2014.

[7] K. Łopatka, J. Kotus, and A. Czyżewski, "Application of vector sensors to acoustic surveillance of a public interior space," *Archives of Acoustics*, vol. 36, no. 4, pp. 851–860, 2011.

[8] K. Łopatka and A. Czyżewski, "Recognition of hazardous acoustic events employing parallel processing on a supercomputing cluster," in *138th Convention of the AES, in print*, (Warsaw), 2015.

[9] K. Łopatka, J. Kotus, and A. Czyżewski, "Evaluation of sound event detection, classification and localization in the presence of background noise for acoustic surveillance of hazardous situations," in *Multimedia Communications, Services and Security*, vol. 429 of *Communications in Computer and Information Science*, pp. 96–110, Springer International Publishing, 2014.

[10] J. Kotus, K. Łopatka, A. Czyżewski, and G. Bogdanis, "Audio-visual surveillance system for application in bank operating room," in *6th Int. Conf. on Multimedia, Communications, Services and Security*, pp. 107–120, 2013.

[11] K. Łopatka and A. Czyżewski, "Automatic regular voice, scream and raised voice recognition employing fuzzy logic," in *132nd Convention of the AES, preprint no. 8636*, (Budapest), 2012.

[12] J. Kotus, K. Łopatka, A. Czyżewski, and H. Krawczyk, "Multimedia system assisting lecturers and public speakers," in *INFOBAZY 2011*, pp. 80–86, 2011.

[13] K. Łopatka, J. Kotus, M. Szczodrak, P. Marcinkowski, A. Korzeniewski, and A. Czyżewski, "Multimodal audio-visual recognition of traffic events," in *Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on*, pp. 376–380, 2011.

[14] K. Łopatka, J. Kotus, and A. Czyżewski, "Monitoring of public events audience employing acoustic vector sensors," in *14th International Symposium on Sound Engineering and Tonmeistering*, 2011.

[15] K. Łopatka, A. Czyżewski, and H. Krawczyk, "Automatic recognition of events in audio data using supercomputer cluster," in *130th Convention of the AES, preprint no. 8337*, (London), 2011.

[16] K. Łopatka, J. Kotus, and A. Czyżewski, "Improving automatic surveillance by sound analysis," in *5th Future Security Conference*, pp. 51–51, 2010.

[17] J. Kotus, K. Łopatka, K. Kopaczewski, and A. Czyżewski, "Automatic audio-visual threat detection," in *IEEE Int. Conf. on Multimedia, Communications, Services and Security*, pp. 140–144, 2010.

[18] K. Łopatka, P. Żwan, and A. Czyżewski, "Dangerous sound event recognition using support vector machine classifiers," *Advances in Multimedia and Network Information System Technologies*, vol. 80, pp. 49–57, 2010.

[19] K. Łopatka, P. Żwan, and A. Czyżewski, "Parameterization of sounds for recognizing hazardous events," *Zeszyty Naukowe Wydzialu Elektroniki, Telekomunikacji i Informatyki Politechniki Gdanskiej: Technologie Informacyjne*, vol. 19, pp. 225–230, 2010.

[20] A. Ciarkowski, J. Cichowski, D. Ellwart, P. Guzik, K. Kopaczewski, J. Kotus, K. Lisowski, K. Łopatka, A. Matiolański, M. Papaj, M. Szczodrak, and G. Szwoch, *KASKADA platform and multimedia applications*, vol. 2, ch. Applications for recognition of persons and events. Gdansk University of Technology, 2013.

[21] R. Lyon, "Machine hearing: An emerging field [exploratory dsp]," *Signal Processing Magazine, IEEE*, vol. 27, pp. 131–139, Sept 2010.

[22] A. Temko, *Acoustic Event Detection and Classification*. PhD thesis, Universitat Politecnica de Catalunya, 2008.

[23] A. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 4–37, Jan.

[24] A. Fazel and S. Chakrabartty, "An overview of statistical pattern recognition techniques for speaker verification," *Circuits and Systems Magazine, IEEE*, vol. 11, no. 2, pp. 62–81, Secondquarter.

[25] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2895 – 2907, 2003.

[26] X. Huang and L. Deng, "An overview of modern speech recognition," in *Handbook of Natural Language Processing, Second Edition* (N. Indurkhya and F. J. Damerau, eds.), Boca Raton, FL: CRC Press, Taylor and Francis Group, 2010. ISBN 978-1420085921.

[27] J. W. Dennis, *Sound event recognition in unstructured environments using spectrogram image processing*. PhD thesis, Nanyang Technological University, 2014.

[28] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shaughnessy, "Developments and directions in speech recognition and understanding, part 1 [dsp education]," *Signal Processing Magazine, IEEE*, vol. 26, pp. 75–80, May 2009.

[29] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *Multimedia, IEEE Transactions on*, vol. 13, pp. 303–319, April 2011.

[30] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, pp. 668–696, April 2008.

[31] J. Downie and J. Futrel, "Terascale music mining," in *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pp. 71–71, 2005.

[32] J. s. Roger Jang, J.-C. Chen, and M. yang Kao, "Miracle: A music information retrieval system with clustered computing engines," in *In Proceedings of International Symposium on Music Information Retrieval*, pp. 11–12, 2001.

[33] C. H. Chen, "Pattern recognition applications in underwater acoustics," *The Journal of the Acoustical Society of America*, vol. 75, no. S1, pp. S75–S75, 1984.

[34] J. C. Wang, H. P. Lee, J. F. Wang, and C. B. Lin, "Robust environmental sound recognition for home automation," *Automation Science and Engineering IEEE Transactions on see also Robotics and Automation IEEE Transactions on*, vol. 5, no. 1, pp. 25–31, 2008.

[35] A. Temko and C. Nadeu, "Acoustic event detection in meeting-room environments," *Pattern Recognition Letters*, vol. 30, no. 14, pp. 1281–1288, 2009.

[36] A. Baijal, J. Kim, J.-h. Jeong, I. Hwang, J. Park, and B.-S. Ko, "Real-time infant cry detection in diverse environments: A novel approach," in *Audio Engineering Society Convention 137*, Oct 2014.

[37] I.-C. Yoo and D. Yook, "Automatic sound recognition for the hearing impaired," *IEEE Trans. on Consum. Electron.*, vol. 54, pp. 2029–2036, Nov. 2008.

[38] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," *Multimedia and Expo 2005 ICME 2005 IEEE International Conference on*, pp. 1306–1309, 2005.

[39] M. Cristani, M. Bicego, and V. Murino, "On-line adaptive background modelling for audio surveillance," *Proceedings of the 17th International Conference on Pattern Recognition 2004 ICPR 2004*, pp. 399–402 Vol.2, 2004.

[40] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "An adaptive framework for acoustic monitoring of potential hazards," *EURASIP Journal on Audio Speech and Music Processing*, no. c, pp. 1–15, 2009.

[41] R. C. Maher, "Acoustical characterization of gunshots," *Signal Processing Applications for Public Security and Forensics*, no. April, pp. 109–113, 2007.

[42] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 21–26, 2007.

[43] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 813–816, 2006.

[44] R. C. Maher, "Modeling and signal processing of acoustic gunshot recordings," 2006.

[45] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[46] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," *CUIDADO IST Project Report*, vol. 54, no. version 1.0, pp. 1–25, 2004.

[47] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, pp. 504 – 516, oct 2002.

[48] H.-G. Kim, N. Moreau, and T. Sikora, *MPEG-7 audio and beyond*. J. Wiley, 2005.

[49] J. D. Krijnders, M. E. Niessen, and T. C. Andringa, "Sound event recognition through expectancy-based evaluation ofsignal-driven hypotheses," *Pattern Recogn. Lett.*, vol. 31, pp. 1552–1559, Sept. 2010.

[50] B. Ghoraani and S. Krishnan, "Frequency matrix feature extraction and classification of environmental audio signals," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 2197–2209, Sept 2011.

[51] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing mel-frequency cepstral coefficients on the power spectrum," *Computer*, vol. 1, pp. 73–76, 2001.

[52] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," *18th European Signal Processing Conference*, pp. 1267–1271, 2010.

[53] J.-L. Rouas, J. Louradour, and S. Ambellouis, "Audio events detection in public transport vehicle," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 733–738, Sept.

[54] M. Vacher, D. Istrate, L. Besacier, J. F. Serignat, and E. Castelli, "Sound Detection and Classification for Medical Telesurvey," in *Proc. 2nd Conference on Biomedical Engineering* (C. ACTA Press, ed.), (Innsbruck, Austria), pp. 395–398, Feb. 2004.

[55] E. Kiktova, M. Lojka, M. Pleva, J. Juhar, and A. Cizmar, "Comparison of different feature types for acoustic event detection system," in *Multimedia Communications, Services and Security* (A. Dziech and A. Czyżewski, eds.), vol. 368 of *Communications in Computer and Information Science*, pp. 288–297, Springer Berlin Heidelberg, 2013.

[56] A. Lindsay and S. Quackenbush, "Overview of mpeg-7 audio," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 725–729, 2001.

[57] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[58] S. Li, R. Xia, C. Zong, and C.-R. Huang, "A framework of feature selection methods for text categorization," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, (Stroudsburg, PA, USA), pp. 692–700, Association for Computational Linguistics, 2009.

[59] L. Ladha and T. Deepa, "Feature selection methods and algorithms," *International Journal on Computer Science and Engineering*, vol. 3, pp. 1787–1790, 2011.

[60] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 9, pp. 1106–1119, July 2012.

[61] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *ARTIFICIAL INTELLIGENCE*, vol. 97, no. 1, pp. 273–324, 1997.

[62] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[63] L. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review." Tilburg University Technical Report, 2008.

[64] K. Sakhnov, E. Verteletskaya, and B. Simak, "Approach for energy-based voice detector with adaptive scaling factor," *International Journal of Computer Science*, vol. 36, no. 4, pp. 1–5, 2009.

[65] A. Dufaux, *Detection and Recognition of Impulsive Sound Signals.* PhD thesis, University of Neuchatel, 2001.

[66] S. Pfeiffer, "Pause concepts for audio segmentation at different semantic levels," in *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, (New York, NY, USA), pp. 187–193, ACM, 2001.

[67] F. Beaufays, D. Boies, M. Weintraub, and Q. Zhu, "Using speech/non-speech detection to bias recognition search on noisy data," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 1, pp. I–424 – I–427 vol.1, april 2003.

[68] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.

[69] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[70] J. A. Arias, J. Pinquier, and R. Andre-Obrecht, "Evaluation of Classification Techniques for Audio Indexing ," in *13th European Conf. on Signal Processing (EUSIPCO'2005) , Antalya, Turkey*, 9 2005.

[71] A. Ito, A. Aiba, A. Ito, and S. Makino, "Detection of abnormal sound using multi-stage gmm for surveillance microphone," in *Information Assurance and Security, 2009. IAS '09. Fifth International Conference on*, vol. 1, pp. 733–736, 2009.

[72] C.-F. Chan and E. W. Yu, "An abnormal sound detection and classification system for surveillance applications," in *18th european Signal Processing Conference*, pp. 1851–1855, August 23-27 2010.

[73] G. Guo and S. Li, "Content-based audio classification and retrieval by support vector machines," *Neural Networks, IEEE Transactions on*, vol. 14, pp. 209 – 215, jan 2003.

[74] A. Temko and C. Nadeu, "Classification of acoustic events using svm-based clustering schemes," *Pattern Recogn.*, vol. 39, pp. 682–694, Apr. 2006.

[75] A. Rabaoui, H. Kadri, Z. Lachiri, and N. Ellouze, "Using robust features with multi-class svms to classify noisy sounds," in *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*, pp. 594–599, 2008.

[76] S. Nirjon, R. F. Dickerson, P. Asare, Q. Li, D. Hong, J. A. Stankovic, P. Hu, G. Shen, and X. Jiang, "Auditeur: a mobile-cloud service platform for acoustic event detection on smartphones," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, MobiSys '13, (New York, NY, USA), pp. 403–416, ACM, 2013.

[77] T. Theodorou, I. Mporas, and N. Fakotakis, "Automatic sound classification of radio broadcast news," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, 2012 2012.

[78] I.-J. Ding, "Fuzzy rule-based system for decision making support of hybrid SVM-GMM acoustic event detections," *International Journal of Fuzzy Systems*, vol. 14, no. 1, pp. 118–130, 2012.

[79] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[80] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic novelty detection for acoustic surveillance under real-world conditions," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, 2011.

[81] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, and D. Povey, "The HTK book (for htk version 3.4)," *Cambridge University*, vol. 2, no. 2, 2006.

[82] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[83] V. N. Vapnik, "An overview of statistical learning theory.," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[84] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[85] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[86] K. bo Duan and S. S. Keerthi, "Which is the best multiclass SVM method? an empirical study," in *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 278–285, 2005.

[87] H. Tran and H. Li, "Sound event recognition with probabilistic distance svms," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 1556 –1568, aug. 2011.

[88] Z. Saric, D. Kukolj, and N. Teslic, "Acoustic source localization in wireless sensor network," *Circuits, Systems and Signal Processing*, vol. 29, no. 5, pp. 837–856, 2010.

[89] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 53, pp. 44–53, Jan 2005.

[90] "Shotspotter brochure." http://www.shotspotter.com/sites/default/files/SST_ShotSpotter_Flex_Brochure_FPV.pdf (visited 2013-07-10).

[91] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide To Theory, Algorithm and System Development*. Prentice Hall, 2001.

[92] J. Stachurski, L. Netsch, and R. Cole, "Sound source localization for video surveillance camera," *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, vol. 0, pp. 93–98, 2013.

[93] A. Pourmohammad and S. M. Ahadi, "N-dimensional n-microphone sound source localization," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 27, 2013.

[94] F. J. Fahy, ed., *Sound Intensity*. London: E & FN Spon, 1989.

[95] F. Jacobsen, "Sound intensity," in *Springer Handbook of Acoustics*, pp. 1053–1075, Springer New York, 2007.

[96] Microflown, "Microflown - acoustic vector sensors." http://www.microflown-avisa.com/acoustic-vector-sensors/ (visited 2014-05-21).

[97] D. H. de Bree and prof.dr.ir. W.F. Druyvesteyn, "A particle velocity sensor to measure the sound from a structure in the presence of background noise," in *Proceedings of the International Conference FORUM ACUSTICUM*, 2005.

[98] D. F. Comesana, E. Tijs, P. Cats, and D. Cook, "Visualization of acoustic intensity vector fields using scanning measurement techniques." September 2013.

[99] H.-E. de Bree and J. W. Wind, "The acoustic vector sensor: a versatile battlefield acoustics sensor," in *Proc. SPIE*, vol. 8047, pp. 80470C–80470C–8, 2011.

[100] J. Kotus and A. Czyżewski, "Acoustic radar employing particle velocity sensors," in *Advances in Multimedia and Network Information System Technologies*, vol. 80, pp. 93–103, Springer Berlin Heidelberg, 2010.

[101] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," *Fifth European Conference on Speech Communication and Technology*, vol. 97, no. 4, pp. 1895–1898, 1997.

[102] T. Fawcett, "An introduction to roc analysis," *Pattern Recogn. Lett.*, vol. 27, pp. 861–874, June 2006.

[103] H. Masnadi-Shirazi and N. Vasconcelos, "On the design of loss functions for classification: theory, robustness to outliers, and savageboost," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1049–1056, Curran Associates, Inc., 2009.

[104] G. Parmigiani and L. Inoue, *Decision Theory: Principles and Approaches*. Wiley & Sons, 2009.

[105] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," *Comput. Linguist.*, vol. 22, pp. 249–254, June 1996.

[106] T. F. W. Embleton, "Tutorial on sound propagation outdoors," *The Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 31–48, 1996.

[107] K. Attenborough, "Sound propagation in the atmosphere," in *Springer Handbook of Acoustics*, pp. 113–147, Springer New York, 2007.

[108] M. C. Berengier, B. Gauvreau, Blanc-Benon, and D. Juve, "Outdoor Sound Propagation: A Short Review on Analytical and Numerical Approaches," *Acta Acustica united with Acustica*, pp. 980–991, 2003.

[109] A. Czyżewski, J. Kotus, and M. Szczodrak, "Creating acoustic maps employing supercomputing cluster," *Archives of Acoustics*, vol. 36, no. 2, pp. 395–418, 2011.

[110] M. Szczodrak, J. Kotus, B. Kostek, and A. Czyzewski, "Creating Dynamic Maps of Noise Threat Using PL-Grid Infrastructure," *Archives of Acoustics*, vol. 38, no. 2, pp. 235–242, 2013.

[111] J. Mateus, F. Malheiro, S. Cavaco, N. Correia, and R. Jesus, "Video annotation of tv content using audiovisual information," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, pp. 113–118, May 2012.

[112] L. De Silva and P. C. Ng, "Bimodal emotion recognition," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 332–335, 2000.

[113] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences.," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 257–267, 2007.

[114] I.-H. Jhuo, G. Ye, S. Gao, D. Liu, Y.-G. Jiang, D. Lee, and S.-F. Chang, "Discovering joint audio–visual codewords for video event detection," *Machine Vision and Applications*, vol. 25, no. 1, pp. 33–47, 2014.

[115] C. Canton-Ferrer, T. Butko, C. Segura, X. Giro, C. Nadeu, J. Hernando, and J. Casas, "Audiovisual event detection towards scene understanding," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 81–88, June 2009.

[116] M. Maroti, G. Simon, A. Ledeczi, and J. Sztipanovits, "Shooter localization in urban terrain," *Computer*, vol. 37, pp. 60–61, Aug. 2004.

[117] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, (New York, NY, USA), pp. 1–12, ACM, 2004.

[118] J. Travis, "Using gunshot detection technology in high-crime areas," 1998.

[119] NetLogix, "viGDS brochure." http://www.netlogix.com/pdf/NetLogix-video-integrated-gunshot-detection-system.pdf (visited 2013-07-10).

[120] BBN Technologies, "Boomerang shooter detection technology." http://bbn.com/resources/pdf/datasheet_BoomerangFixedSite.pdf (visited 2013-07-10).

[121] J. Proficz, *Computational resources management in cluster environment for multimedia streams processing*. PhD thesis, Gdańsk University of Technology, 2012.

[122] M. Flynn, "Some computer organizations and their effectiveness," *Computers, IEEE Transactions on*, vol. C-21, pp. 948–960, Sept 1972.

[123] M. D. Hill, "What is scalability?," *SIGARCH Comput. Archit. News*, vol. 18, pp. 18–21, Dec. 1990.

[124] D. Eager, J. Zahorjan, and E. Lazowska, "Speedup versus efficiency in parallel systems," *Computers, IEEE Transactions on*, vol. 38, pp. 408–423, Mar 1989.

[125] P. Jogalekar, M. Woodside, and S. Member, "Evaluating the scalability of distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, pp. 589–603, 2000.

[126] "Intel architecture instruction set extensions programming reference." http://download-software.intel.com/sites/default/files/319433-014.pdf.

[127] "CUDA toolkit documentation." http://docs.nvidia.com/cuda (visited 2013-07-10).

[128] N. Tsingos, W. Jiang, and I. Williams, "Using programmable graphics hardware for acoustics and audio rendering," *J. Audio Eng. Soc*, vol. 59, no. 9, pp. 628–646, 2011.

[129] N. Jillings and Y. Wang, "Cuda accelerated audio digital signal processing for real-time algorithms," in *Audio Engineering Society Convention 137*, Oct 2014.

[130] L. Savioja, V. Välimäki, and J. O. Smith, "Real-time additive synthesis with one million sinusoids using a GPU," in *AES 128th Convention*, (London, UK), 2010.

[131] L. Savioja, "Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics," in *Proc. Int. Conf. Digital Audio Effects*, (Graz, Austria), 2010.

[132] A. Angus, Jamie A. S.; Caunce, "A GPGPU approach to improved acoustic finite difference time domain calculations," in *Audio Engineering Society Convention 128*, 5 2010.

[133] N. Röber, U. Kaminski, and M. Masuch, "Ray acoustics using computer graphics technology," in *10th Int. Conf. on Digital Audio Effects*, (Bordeaux, France), pp. 117–124, 2007.

[134] T.-Y. Liang, T.-H. Wang, M.-T. Chou, and S.-W. Chen, "A cloud computing service for fast audio source signal separation," in *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pp. 1–6, 2011.

[135] I. Schmadecke, J. Morschbach, and H. Blume, "Gpu-based acoustic feature extraction for electronic media processing," in *Electronic Media Technology (CEMT), 2011 14th ITG Conference on*, pp. 1–6, March 2011.

[136] P. R. Dixon, T. Oonishi, and S. Furui, "Harnessing graphics processors for the fast computation of acoustic likelihoods in speech recognition," *Computer Speech & Language*, vol. 23, no. 4, pp. 510 – 526, 2009.

[137] Y. Chen, W. Wei, and Y. Zhang, "Parallel audio quick search on shared-memory multiprocessor systems," in *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pp. 1–6, 2007.

[138] J. Schimmel, "Using parallel signal processing in real-time audio matrix systems," *W. Trans. on Comp.*, vol. 9, pp. 174–183, Feb. 2010.

[139] P. Dziurzański and T. Maka, "Features extraction system for automatic speech recognition core mapping into an irregular network on chip," *Elektronika: konstrukcje, technologie, zastosowania*, vol. 53, no. 9, pp. 154–156, 2012.

[140] P. Dziurzański and T. Maka, "Core mapping into an irregular network on chip - features extraction system for automatic speech recognition case study," in *Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on*, pp. 494–498, Feb 2013.

[141] T. Maka and P. Dziurzański, "Parallel audio features extraction for sound indexing and retrieval systems," in *ELMAR, 2013 55th International Symposium*, pp. 185–189, Sept 2013.

[142] E. M. Schmidt, K. West, and Y. E. Kim, "Efficient acoustic feature extraction for music information retrieval using programmable gate arrays," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, (Kobe, Japan), pp. 273–278, October 26-30 2009. http://ismir2009.ismir.net/proceedings/PS2-14.pdf.

[143] T. Yiyu, Y. Inoguchi, Y. Sato, M. Otani, Y. Iwaya, and T. Tsuchiya, "Design and implementation of a two-dimensional sound field solver based on the digital huygens' model," *Microprocessors and Microsystems*, vol. 38, no. 3, pp. 216 – 225, 2014.

[144] T. Blechmann, "Supernova: A multiprocessor aware real-time audio synthesis engine for supercollider," Master's thesis, Vienna University of Technology, 2011.

[145] "Waves soundgrid: Audio-over-ethernet networking & processing platform." http://www.waveslive.com/pdf/soundgrid-white-paper.pdf, 2013.

[146] K. You, J. Chong, Y. Yi, E. Gonina, C. Hughes, Y.-K. Chen, W. Sung, and K. Keutzer, "Parallel scalability in speech recognition," *Signal Processing Magazine, IEEE*, vol. 26, no. 6, pp. 124–135, 2009.

[147] J. Chong, G. Friedland, A. Janin, N. Morgan, and C. Oei, "Opportunities and challenges of parallelizing speech recognition," in *Proceedings of the 2nd USENIX conference on Hot topics in parallelism*, HotPar'10, (Berkeley, CA, USA), pp. 2–2, USENIX Association, 2010.

[148] D. Lee, M. Schultz, and F. Saied, "Supercomputers in computational ocean acoustics," in *Supercomputing, 1989. Supercomputing '89. Proceedings of the 1989 ACM/IEEE Conference on*, pp. 133–140, 1989.

[149] X. Cai and A. Odegard, "Parallel simulation of 3d nonlinear acoustic fields on a linux-cluster.," in *CLUSTER*, pp. 185–192, IEEE Computer Society, 2000.

[150] L. Giraud, *Large Scale Acoustic Simulations on Clusters of SMPs*, pp. 61–66. Springer, 2004.

[151] "Benefits of supercomputing." http://www.deisa.eu/news/press/Media/DEISA-Digest.pdf, 2008.

[152] J. Xiaojing, "Google cloud computing platform technology architecture and the impact of its cost," in *Software Engineering (WCSE), 2010 Second World Congress on*, vol. 2, pp. 17–20, Dec 2010.

[153] M. Gusev and S. Ristov, "Superlinear speedup in windows azure cloud," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pp. 173–175, Nov 2012.

[154] J. Wenyu, Z. Yongwei, B. Xiaoming, and Y. Rongshan, "Cloud-based audio fingerprinting service," in *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pp. 1–6, Dec 2012.

[155] H. Krawczyk, *Platform and applications for data stream processing*, vol. 1, ch. Platform and applications for data stream processing. Gdansk University of Technology, 2013.

[156] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, pp. 216–231, Feb 2005.

[157] S. Winitzki, "A handy approximation for the error function and its inverse," *private communication*, vol. 2667, pp. 6–8, 2008.

[158] P. Żwan and A. Czyżewski, "Verification of the parameterization methods in the context of automatic recognition of sounds related to danger," *Journal of Digital Forensic Practice*, vol. 3, no. 1, pp. 33–45, 2010.

[159] J. Antoni, "The spectral kurtosis: a useful tool for characterising non-stationary signals," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 282 – 307, 2006.

[160] M. Kos, Z. Kacic, and D. Vlaj, "Acoustic classification and segmentation using modified spectral roll-off and variance-based features," *Digital Signal Processing*, vol. 23, no. 2, pp. 659 – 674, 2013.

[161] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *Audio and Electroacoustics, IEEE Transactions on*, vol. 15, pp. 70 – 73, jun 1967.

[162] H. Krawczyk and J. Proficz, "KASKADA - multimedia processing platform architecture," in *Signal Processing and Multimedia Applications (SIGMAP), Proceedings of the 2010 International Conference on*, pp. 26–31, July 2010.

[163] H. Krawczyk and J. Proficz, *Real-Time Multimedia Stream Data Processing in a Supercomputer Environment*, pp. 289–312. InTech, 2012.

[164] H. Krawczyk and J. Proficz, *Interactive Multimedia*, ch. Real-time multimedia stream data processing in a supercomputer environment. InTech, 2012.

[165] H. Krawczyk, R. Knopa, and J. Proficz, "Basic management strategies on kaskada platform," in *International Conference on Computer as a Tool & 8th Conference on Telecommunications*, 2011.

[166] P. Szczuko, "Hierarchical estimation of human upper body based on 2d observation utilizing evolutionary programming and "genetic memory"," in *Multimedia Communications, Services and Security* (A. Dziech and A. Czyżewski, eds.), vol. 149 of *Communications in Computer and Information Science*, pp. 82–90, Springer Berlin Heidelberg, 2011.

[167] P. Marcinkowski, A. Korzeniewski, and A. Czyzewski, "Human tracking in multi-camera visual surveillance system," in *Multimedia Communications, Services and Security* (A. Dziech and A. Czyżewski, eds.), vol. 149 of *Communications in Computer and Information Science*, pp. 277–285, Springer Berlin Heidelberg, 2011.

[168] M. Szczodrak, J. Kotus, K. Kopaczewski, K. Łopatka, A. Czyżewski, and H. Krawczyk, "Behavior analysis and dynamic crowd management in video surveillance system," in *Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on*, pp. 371–375, Aug 2011.

[169] K. Kopaczewski, M. Szczodrak, A. Czyżewski, and H. Krawczyk, "Application of virtual gate for counting people participating in large public events," in *Multimedia Communications, Services and Security* (A. Dziech and A. Czyżewski, eds.), vol. 287 of *Communications in Computer and Information Science*, pp. 316–327, Springer Berlin Heidelberg, 2012.

[170] J. Proficz, A. Brzeski, P. Czarnul, B. Daca, R. Knopa, and M. Westa, *KASKADA platform and multimedia applications*, vol. 1, ch. Functionality of the KASKADA platform. Gdansk University of Technology, 2013.

[171] P. Czarnul, "Kaskada platform," *High performance computing systems: laboratory manual*, Gdańsk University of Technology, 2010.

[172] B. Shannon and K. Paliwal, "A comparative study of filter bank spacing for speech recognition," in *Proceedings of Microelectronic Engineering Research Conference*, 2003.

[173] J. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401–409, 1969.

[174] S. Glantz and B. Slinker, *Primer of Applied Regression & Analysis of Variance*. McGraw-Hill Education, 2000.

[175] K. Kopaczewski, M. Szczodrak, A. Czyzewski, and H. Krawczyk, "A method for counting people attending large public events," *Multimedia Tools and Applications*, pp. 1–13, 2013.

[176] "Free sound database." http://freesound.org (visited 2014-12-01).

[177] M. Saraclar, M. Riley, E. Bocchieri, and V. Goffin, "Towards automatic closed captioning : low latency real time broadcast news transcription.," in *INTERSPEECH* (J. H. L. Hansen and B. L. Pellom, eds.), ISCA, 2002.

[178] G. Salvi, "Truncation error and dynamics in very low latency phonetic recognition," in *In Non-linear Speech Signal Processing, NOLISP2003 Sjölander, K*, 2003.

[179] A. Seward, "Low-latency incremental speech transcription in the synface project.," in *INTERSPEECH*, ISCA, 2003.

[180] M. Walker, "Dealing with computer audio latency." http://www.soundonsound.com/sos/apr99/articles/letency.htm, 1999.

[181] G. Orwell, *Nineteen eighty-four*. London: Secker and Warburg, 1949.

[182] "ADDPRIV project website." www.addpriv.eu (visited 2013-07-10).

[183] A. Chattopadhyay and T. Boult, "Privacycam: a privacy preserving camera using uclinux on the blackfin dsp," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, 2007.

[184] J. Cichowski and A. Czyżewski, "Reversible video stream anonymization for video surveillance systems based on pixels relocation and watermarking," in *Int. Conf. on Computer Vision*, (Barcelona), pp. 1971–1977, 2011.